X-терминал или вторая жизнь Вашего ПК

Уважаемые читатели!

Представляю Вам мою книгу, которая
появилась в результате экспериментов по использовании морально устаревших компьютеров в реальных условиях современного офиса.

Надеюсь, что кому-то из Вас она поможет настроить и
использовать старые ПК такие как Intel Pentium или i486.

Если у Вас возникнут вопросы, пожелания или предложения,
вы можете отправить их мне на e-mail: senyk_cw@mail.ru

Книгу в её первоначальном варианте вы можете посмотреть http://www.senyk.poltava.ua/projs/books/XTerms/index.html#contents

С уважением, Сеник Николай

Содержание:

Часть I. Терминалы

Глава 1. Общие сведения о бездисковых компьютерах

История возникновения персонального компьютера без “жесткого” диска

Общая архитектура и область применения

Общая архитектура и область применения (продолжение)

Обзор существующих решений для сетевых компьютеров

Citrix MetaFrame

Windows Terminal Services

VNC

NCD ThinPATH

NoMachine

Thinstation

LTSP

X terminal kit

X-Windows и XDMCP

Достоинства и недостатки терминальных решений

Минусы

Плюсы

Общие сведения о используемых протоколах (TFTP, DHCP, NFS, UDP, TCP/IP, SSH)

IP

ICMP

ARP

RARP

TCP

UDP

NFS

TFTP

DHCP
SSH
Администрирование и сопровождение терминальной инфраструктуры
Часть II. X-терминалы
Глава 2. Общие сведения о X-терминалах
Как это работает (краткое описание процесса работы и загрузки)
Удаленный запуск консольных программ
X-терминалы – бездисковые клиенты X-Windows
Удаленный запуск GUI-приложений (XForwarding)
Глава 3. X-Windows
Описание X-протокола
Архитектура X-Windows
X-сервер и X-клиент
X-сервер
X-клиент
Запросы
Ответы
События
Сообщения об ошибках
Настройка X-сервера и XDM/XDMCP (часть 1)
Настройка X-сервера и XDM/XDMCP (часть 2)
Настройка X-сервера и XDM/XDMCP (часть 3)
Настройка X-сервера и XDM/XDMCP (часть 4)
Менеджер дисплея и XDMCP
Глава 4. Аппаратная база X-терминальной сети
Системные требования к серверу X-терминалов
Требования к аппаратному обеспечению клиентов
Делаем терминал из персональных компьютеров серии 386/486/Pentium
Список видеоадаптеров, использующих сервер XFree86 версии 4.x
Список видеоадаптеров, использующих сервер XFree86 версии 3.3.6
Минимальные требования к видеокарте X-терминала

Собираем X-терминал сами
Готовые терминалы: что можно купить за деньги?
Требования к локальной сети (проектирование сети на оптимальную загрузку)
Глава 5. Выбор операционной системы для сервера
Ключевые моменты выбора операционной системы
ASPLinux
RedHat Linux
Fedora Core
Debian GNU/Linux
SuSE Linux
Mandrakelinux
AltLinux
Gentoo
Slackware
Linux XP
Linspire
myLinux
Планирование эксплуатационного обслуживания сервера
Планирование эксплуатационного обслуживания сервера (продолжение)
Перенос операционной системы на новый сервер
Часть III. Настройка системы (на примере использования LTSP)
Глава 6. Настройка сервера
Установка программного обеспечения (ltsp)
Установка программного обеспечения (nfs, dhcpd, tftp-server, xinetd)
Установка программного обеспечения (инициализация ltsp)
Проверка DHCPD
Проверка NFS и portmap
Проверка Syslog
Проверка xfs
Проверка xinetd и tftpd
Общая настройка системы (inittab, syslog)

Настройка inittab
Настройка syslog
Конфигурирование сетевых служб (dhcp, tftp, nfs, tcpd, dns)
Настройка DHCP
Настройка xinetd
Настройка TFTP
Настройка NFS
Настройка переключателей TCP
Настройка DNS
Настройка X-Windows (Xaccess, Xsetup_workstation, xdm, kdm, gdm, xfs)
Настройка удаленного доступа к X-серверу (Xaccess)
Настройка экрана ввода имени пользователя и пароля (Xsetup_workstation)
Настройка диспетчера экрана XDM
Настройка менеджера дисплея KDM
Настройка менеджера дисплея GDM
Выбор менеджера дисплея по умолчанию
Настройка сервера шрифтов (xfs)
Главный файл конфигурации сервера LTSP
Реконфигурация сервера при добавлении новых терминалов
Глава 7. Настройка терминалов
Выбор метода сетевой загрузки
Создание дискеты с загрузочным образом
Создание дискеты с загрузочным образом (продолжение)
Определения MAC-адреса сетевой карты
Загрузка X-терминала с других носителей информации(lilo, cd-rom, msdos)
Загрузка при помощи(LInux LOader)
Создание загрузочного компакт диска
Загрузка в режиме MS DOS
Использование сетевых карт с BootROM или PXE
BootROM
Использование сетевых адаптеров PXE

Настройка сетевой карты ISA

Создание пользовательского образа для загрузки

Загрузка в текстовом режиме

Загрузка в текстовом режиме (продолжение)

Конфигурирование X-клиента

Автоматическое построение файла XF86Config

Оптимизация конфигурации клиента

Оптимизация конфигурации клиента (продолжение)

Типовые конфигурации клиентов

Клавиатура

“Мышка” подключенная к порту PS/2 трех кнопочная без скроллера

“Мышка” подключенная к порту PS/2 двух кнопочная без скроллера

“Мышка” подключенная к порту PS/2 двух кнопочная со скроллером

“Мышка” подключенная к COM-порту (COM1) без скроллера (3 кнопки)

“Мышка” подключенная к COM-порту (COM2) без скроллера (2 кнопки с эмуляцией нажатия третьей)

“Мышка” подключенная к COM-порту (COM1) со скроллером (2 кнопки)

“Мышка” подключенная к USB-порту со скроллером (2 кнопки)

Разрешение для ЭЛТ-монитора с диагональю 15” (800x600)

Разрешение для ЭЛТ-монитора с диагональю 17” (1024x768)

Разрешение для ЭЛТ-монитора с диагональю 19” (1280x1024)

Разрешение для LCD-монитора с диагональю 15” (1024x768)

Разрешение для ЭЛТ-монитора с диагональю 17” (1280x1024)

Глубина цвета HiColor (16 бит)

Глубина цвета TrueColor (24 бита)

Глубина цвета TrueColor (32 бита)

Горизонтальная и вертикальная частоты разверстки для ЭЛТ-монитора с диагональю 17”

Запрет аппаратной акселерации (иногда помогает избавиться от артефактов)

Работа в режиме VESA

X-сервер для большинства видеоадаптером на чипсете s3

X-сервер SVGA версии 3.3.6

Использование принтера подключенного к параллельному порту X-терминала (порт tcp/ip 9100)

Использование принтера подключенного к параллельному порту X-терминала с нестандартным портом tcp/ip, например, 9150

Использование принтера подключенного к USB-порту X-терминала (порт tcp/ip 9100)

Автоматическая настройка звуковой карты на X-терминале (демон звука EsounD)

Ручная настройка звуковой карты на X-терминале (демон звука EsounD, звуковая карта на чипсете OPL3sa2)

Влияние видеосистемы на работу X-терминала

Проблемы, требующие собственного файла XF86Config

Выбор оптимального программного обеспечения для работы на X-терминале

Выбор оптимального программного обеспечения для работы на X-терминале (продолжение, IceWM vs XFce)

Советы по работе и настройке типового софта – OpenOffice, Gimp, Evolution и т.д.

Файловый менеджер

Веб-браузер

Почтовый клиент

Клиент ICQ

Пакет офисных приложений

CrossOver Office

StarOffice

KOffice

Gnome Office

OpenOffice

OpenOffice (продолжение)

Графический редактор Gimp

Архиватор

Дополнительные параметры файла lts.conf
Глава 8. Администрирование сети X-терминалов
Если что-то не работает – порядок выполнения диагностики
Проблемы загрузочного образа
Неполадки DHCP
Неполадки DHCP (продолжение)
Проблемы с TFTP
Устранение проблем с NFS
Проблемы X-сервера
Проблемы X-сервера (продолжение)
Проблемы менеджера дисплея
XDM
KDM
GDM
Ошибки при работе в графическом режиме
Подготовка пользователей для работы на X-терминале
Порядок включения и выключения X-терминала
Вход в систему и завершение сеанса работы
Проверка наличие сетевого кабеля
Практическая бесполезность перезагрузки
Общая рабочая среда
Универсальность рабочих мест
Особенности работы с устройствами хранения данных и передача файлов
Новая операционная система Linux и ее прикладные программы
Инструкция для пользователя
Аудит сети X-терминалов
"Уборка мусора"
"Уборка мусора" (продолжение)
Оптимальный выбор программного обеспечения
Анализ использования локальной сети
Анализ использования локальной сети (продолжение)

Распределение процессорного времени
Контроль за использованием дискового пространства
Глава 9. Работа с периферией на терминале
Подключение к терминалу принтера CUPS (LPT, USB)
Подключение к терминалу принтера CUPS (продолжение)
Подключение принтера к параллельному порту (LPT)
Подключение принтера к порту USB
Музыка на терминале
Музыка на терминале (продолжение)
Музыка на терминале (продолжение)
Смотрим видеофильмы на терминале
Использование сканера
Доступ пользователей к флоппи дисководам
Локальный запуск приложений
Локальный запуск приложений (продолжение)
Приложение А. Реальный пример внедрения X-терминалов
Linux в реальном офисе (часть 1)
Linux в реальном офисе (часть 2)
Linux в реальном офисе (часть 3)
Linux в реальном офисе (часть 4)
Linux в реальном офисе (часть 5)
Linux в реальном офисе (часть 6)
Linux в реальном офисе (часть 7)
Linux в реальном офисе (часть 8)
Linux в реальном офисе (часть 9)
Linux в реальном офисе (часть 10)
Linux в реальном офисе (часть 11)
Приложение В. Отличия в проекте LTSP 4.x
Установка
Конфигурирование и настройка
Сценарии для виртуальных экранов

Новое в синтаксисе файла /opt/ltsp/i386/etc/lts.conf

Новый демон ltspinfod

Локальный запуск приложений

Исходный код проекта LTSP

Среда компиляции программ для LTSP

Доступ к локальным устройствам

Часть I. Терминалы

Глава 1. Общие сведения о бездисковых компьютерах

История возникновения персонального компьютера без “жесткого” диска

Если говорить по существу, то идея использования бездискового компьютера поначалу была воспринято мною без особого энтузиазма. Опыт работы на персональных компьютерах просто не позволял допустить, что компьютер без “винчестера” может быть полноценным рабочим инструментом. Как можно отказаться от такой полезной штуки как жесткий диск, и главное зачем? А на практике все было достаточно просто. Как системный администратор, я стал видеть, что парк компьютерной техники морально устаревает значительно быстрее, чем происходит физический износ. Имея в своем распоряжении десяток “Пентиумов”, приобретенных в 1997-98 годах и находящихся полностью в работоспособном состоянии, мне стало все сложнее и сложнее их обслуживать, так как от пользователей постоянно слышались жалобы на медленную работу их компьютеров. Конечно, при достаточном финансировании ничто не мешает приобретать новые компьютеры раз в 3 года,

но в условиях нашей страны это может позволить себе далеко не каждая организация. В общем, денег на новые компьютеры не дали, а меня как системного администратора обязали найти выход из сложившейся ситуации. Аргументы начальства были "железные": зачем нам новые компьютеры, если пользователю нужно только набрать текстовый документ, посчитать что-то в электронной таблице, напечатать это на принтере ну и поработать в Интернете. И выход был вскоре найден в использовании бездисковых сетевых компьютеров, которые были переделаны из старых “морально изношенных” персоналок. Но обо всем по-порядку.

Жесткий диск или “винчестер” - это не самый главный компонент персонального компьютера. На этапе зарождения эры персональных компьютеров – они вообще были лишены дисковых накопителей. Единственным средством хранения информации были гибкие диски (дискеты) и флоппи-дисководы для работы с ними. Хотя первый винчестер был сконструирован в 1956 году (на самом деле, это был прототип винчестера - жесткий диск КАМАС 305, созданный IBM), их массовое применение началось только с 1983 года, когда фирма Seagate Technologyс выпустила их для пользователей персональных компьютеров IBM PC/XT. Использование жесткого диска как средства хранения и многократного использования информации поставило персональные компьютеры на качественно новый уровень. Посудите сами: скорость и емкость современных винчестеров просто поражают. Теперь на них можно хранить музыку, фотографии, фильмы, программы, документы и т.д. В общем, как ни крути, хорошая штука этот “винчетсер” и совсем не понятно, зачем нужен компьютер без него. Но если более детально подойти к вопросу о необходимости “винчестера” в компьютере, то оказывается, что эта деталь имеет и ряд существенных изъянов. Главный минус “винчестера” - это его механическая природа, вращающиеся внутри компоненты подвержены износу, а, стало быть, могут послужить причиной выхода из строя. Вот вам и парадокс: самое ценное, что есть в компьютере - информацию, мы доверяем одному из самых ненадежных его компонентов - “винчестеру”. Конечно, скептики скажут, что не так уж часто

ломаются жесткие диски, и существуют модели “винчестеров”, которые исправно работают на протяжении нескольких лет, но факт остается фактом: “винчестер” ненадежная механическая деталь персонального компьютера. Кроме своей “ненадежности”, жесткому диску присуща еще одна отрицательная черта – стоит он достаточно дорого. Рассмотрим простой пример: локальную сеть банка. Все служащие банка работают с информацией, которая составляет информационную банковскую систему. Понятно, что ей присущи меры обеспечения безопасности, разграничения полномочий и т.д., но по сути это одна большая база данных, которая хранится на серверах, физический доступ к которым не имеют рядовые сотрудники. Вопрос состоит в том, зачем локально хранить какую-либо информацию на винчестере, если все равно важных данных там нет? А если речь идет об отделе с сотней компьютеров, вы только представьте себе о какой колоссальной экономии идет речь (100 х средняя стоимость “винчестера”)!

Из приведенных выше замечаний не следует делать вывод, что я предлагаю вам выбросить ваш “винчестер” и с головой окунуться в мир бездисковых сетевых компьютеров. Просто давайте посмотрим на персональный компьютер под другим углом, представим его частью локальной сети и задумаемся: “А такая ли необходимая и незаменимая вещь жесткий диск?”. С радостью сообщаю, что в наших мыслях мы будем далеко не одиноки. Идея сетевых компьютеров будоражит информационные технологии уже не первый год. Как сейчас помню, что в конце 90-х годов XX века, чуть ли не в каждом номере компьютерного журнала или газеты можно было прочитать о новых разработках ПК без жесткого диска, о так называемых “тонких” клиентах или сетевых компьютерах. В последнее время интерес к ним вроде снизился, но это никак не связано с бесперспективностью данного технологического направления. Дело просто в том, что на рынке появились стабильные реализации терминальных компьютерных сетей и интерес к бездисковым ПК перешел из разряда новинок в разряд изученных и применяемых методик, бездисковые компьютеры перестали быть чем-то принципиально новым.

Честно признаться, когда я в первые услышал о ПК без жесткого диска, то эта новость вызвала у меня лишь слабую улыбку, так как более бесполезное изобретение сложно было себе даже представить. Пожалуй конкуренцию ему могут составить разве что: “ПК без центрального процессора” и “ПК без оперативной памяти”. Мое сознание рисовало мне картины подключения этого чуда буржуазной мысли к какой-то невиданной компьютерной сети, через которую этот, по моему мнению, “безмозглый” ПК, загружал операционную систему и получал доступ к удаленным файлам. Но здравый смысл мне подсказывал, что в любом случае информацию нужно где-то локально хранить, пусть даже временно, а следовательно этот “тонкий” клиент должен иметь фантастический объем оперативной памяти. Тогда где экономия от отсутствия “винчестера”? В общем, вопросов было больше, чем ответов и тогда я вовсе не проникся идеей бездискового компьютера. Со временем я все больше и больше погружался в рутину администрирования, а это - установка операционной системы, установка программ, обновление программных продуктов, управление пользователями, сетевыми устройствами, правами доступа и т.п., и поневоле стал замечать, что обслуживание одного отдельно стоящего ПК на порядок сложнее, чем работа с компьютерами, которые объединены в локальную сеть. Более того, я реально стал видеть, что в офисе практически все пользователи компьютеров работают с однотипным набором программных продуктов, и за редким исключением их интересы выходят за пределы офисных приложений, электронной почты и Интернет браузера. К тому же год за годом, парк компьютерной техники на глазах подвергался моральному износу, то есть все компьютеры работали и даже на “винчестерах”, которым было по 5 лет, не было BAD-секторов, но установить, а главное комфортно работать с современным программным обеспечением на них не было никакой возможности. Современные программные продукты действительно нерационально используют аппаратные ресурсы компьютера. Сами посудите, ну зачем для работы с текстовым документом вам нужен компьютер с центральным процессором в 3 ГГц! К тому же администрирование “разношерстной” локальной сети (имеется в виду компьютерная локальная сеть

состоящая из ПК с различными типами и версиями операционных систем) порой оказывается достаточно сложной задачей и идея привести все рабочие станции к одному знаменателю выглядит как никогда заманчивой. Только представьте, что в новой локальной сети есть сервер, на котором храниться вся информация о настройках удаленных компьютеров, правах доступа к файлам, сетевый устройствам и т.д. Пользовательский компьютер подключается к такому серверу и получает от последнего все, о чем только пожелает. В этой локальной сети системный администратор фактически администрирует только один компьютер (сервер) и именно работе с ним посвящает практически все свое рабочее время. Рабочие станции пользователей по сути утрачивают свою индивидуальность, так как человеку становится одинаково на каком компьютере работать. Сев за любую машину (прошу прощения, персональный компьютер) вы получаете ваш привычный рабочий стол, общие и личные документы, прикладные программы и знакомую операционную систему. Кажется слишком радужная получилась картина, но с небольшой натяжкой именно так и выглядит работа сети бездисковых компьютеров и вообще терминальные вычислительные решения.

Хранение совместно используемых данных на "винчестере" обычного персонального компьютера сопрящено еще с одной трудностью. Получить к ним доступ становиться возможным только после того, как этот компьютер будет включен и загрузиться его операционная система, а следовательно это не очень удобно, особенно если пользователи компьютеров находятся в разных кабинетах или даже зданиях.

Из вышесказанного напрашивается вывод, что “винчестер” не является краеугольным камнем в борьбе между традиционными персональными компьютерами и “тонкими” сетевыми клиентами. Проблема “винчестера” заключается в том, что при коллективной работе и обмене информацией, локальное хранение данных препятствует их совместному использованию, и это утверждение особенно справедливо для тех случаев, когда количество участников совместной деятельности возрастает.

Общая архитектура и область применения

Идея терминальной вычислительной среды не нова. Еще на заре эпохи компьютеризации, когда физические размеры компьютеров просто поражали, стоили они очень дорого, а в Советском Союзе из называли ЭВМ (электронно-вычислительные машины), кем-то было принято решение придумать способ параллельной работы с этими агрегатами сразу нескольких пользователей. Фактически к одному большому компьютеру подсоединялись терминалы, оборудованные мониторами (раньше их называли дисплеями) и клавиатурами. В результате пользователи имели значительное снижение стоимости эксплуатации информационного центра в пересчете на количество активных пользователей. Дальнейшую историю развития информационной технологии вы знаете, появились относительно дешевые персональные компьютеры, которые и вытеснили терминальные решения как своего рода анахронизм. Когда персональных компьютеров стало слишком много и повсеместно стали внедряться локальные сети, то оказалось, что персональному компьютеру в сети не хватает сервера для хранения общих документов, администрирования и зашиты от внешней глобальной сети Интернет. Дальше все еще интереснее, с ростом локальных компьютерных сетей возникла потребность в их удаленном администрировании и, в идеальном варианте, мы опять вернулись к идеи терминального компьютера. По сути, компьютерная отрасль как бы раскачивается между двумя полюсами: идеей частного самодостаточного компьютера и глобальной вычислительной средой. В зависимости от уровня развития технологии и потребностей конечных пользователей, на верх всплывает та или иная идея. Такая колебательная особенность развития информационной технологии подтверждает ее системный характер.

Давайте схематично рассмотрим отличия работы терминала и традиционного персонального компьютера. В терминальном компьютере присутствует некая программа, которая позволяет ему работать с локальным оборудованием и выполняет одну главную функцию: обеспечивает процесс передачи информации между клиентом и сервером. Все основные вычисления выполняются на сервере, а поэтому к терминалам предъявляются такие низкие аппаратные требования. То есть реально присутствует распределенная клиент-

серверная архитектура, причем зависимость клиента от сервера очень значительна.

Для более наглядной иллюстрации этой идеи посмотрите на схематичное изображение зависимости программы в памяти и ее прикладного пользовательского интерфейса для традиционного персонального компьютера (см. рис. 1.1):

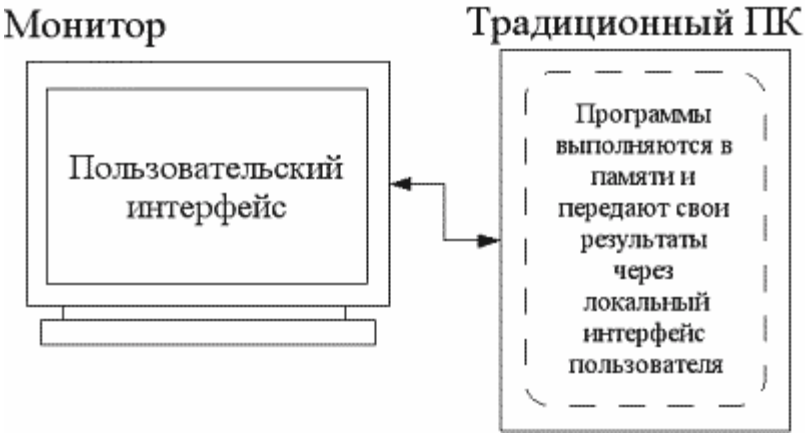

Рис. 1.1. Схема связи пользовательского интерфейса для традиционного персонального компьютера

Как видно из схемы, программа, которая выполняется в памяти персонального компьютера, непосредственно выводит информацию на локальный монитор. То есть никакие дополнительные службы в процессе обмена информацией между компьютером и пользователем не участвуют. Совсем другая картина имеет место у бездисковых терминальных клиентов (см. рис. 1.2):

Рис. 1.2. Схема взаимодействия пользовательского интерфейса бездискового компьютера

Как видно на рисунке 1.2, сервер выполняет все пользовательские приложения, а результаты отсылает клиенту в унифицированной форме, что позволяет последнему правильно их обрабатывать и выводит на экран монитора. В свою очередь, если пользователь бездискового компьютера взаимодействует со своим интерфейсом посредством нажатия клавиш на клавиатуре или при помощи “мыши”, то эти данные собираются клиентским приложением бездискового компьютера и отправляются также в унифицированной форме приложению, что работает на сервере для последующей обработки.

Данная схема конечно очень сильно упрощена, и в реальных терминальных структурах может присутствовать более изощренная методика обработки запросов удаленных пользователей. Так, например, при достаточном аппаратном обеспечении бездисковых компьютеров, они могут выполнять не одну, а сразу несколько клиентских программ, чем значительно увеличивают свои возможности (например, вполне логично запустить mp3-плеер

непосредственно локально на бездисковом компьютере, чем использовать для передачи музыки локальную сеть).

Область применения бездисковых терминальных решений очень широка. В первую очередь ею должны заинтересоваться учреждения офисного типа: банки, крупные и средние предприятия. Главное условие, которое должно выполняться – это наличие достаточного количества пользователей (от 5-ти и выше). Чем больше человек работает в организации, тем очевиднее становятся преимущества от внедрения терминальных компьютеров. Любая организации в своей работе сталкивается с затратами на компьютерную технику. Будь-то покупка аппаратного обеспечения, его ремонт или затраты на обслуживание и администрирование. Не следует также забывать и о стоимости лицензионного программного обеспечения. Другими словами, с увеличением количества компьютеров в организации резко возрастают затраты на поддержание их в работоспособном состоянии. Поэтому использование терминальных решений помогает таким структурам не только снизить стоимость эксплуатации всей своей информационной системы, но и значительно улучшить качество ее администрирования, а в результате получить более надежную и защищенную корпоративную информационную среду.

Общая архитектура и область применения (продолжение)

Также очень перспективно применение терминальных вычислений в образовательных заведениях, таких как школы, институты, университеты и техникумы. Посетите любой компьютерный класс в ближайшей школе или университете. Скорее всего вы там найдете десяток морально устаревших компьютеров, возраст которых 3-4 года (“преклонный” по меркам информационной компьютерной индустрии). Скажите, о какой современной науке или образовании может идти речь, когда учащиеся и студенты вынуждены проводить образовательный процесс на морально устаревшем оборудовании. Конечно, можно каждые два года закупать новые компьютеры и тем самым поддерживать аппаратную базу на должном уровне, но прошу не забывать в какой стране мы живем, скорее всего подавляющему количеству

учебных заведений это просто не по карману. Сфера образования может найти в терминальных бездисковых компьютерах новое применение своему устаревшему парку компьютерной техники. Тем самым при минимальных финансовых затратах в учебных заведениях можно работать с самым новым программным обеспечением и при этом срок службы терминальных компьютеров резко увеличивается по сравнению с традиционными ПК. Кроме этого, использование в образовательном процессе клиент-серверной архитектуры терминальных бездисковых компьютеров позволит обучающимся ближе познакомиться с современными технологическими решениями и даже применять их на практике. За использование в учебных заведениях бездисковых компьютеров говорит и тот факт, что программное обеспечение, которое нужно среднестатистическому студенту, отлично работает на терминалах (офисные приложение, веб-браузер, электронная почта, файловый менеджер и т.д.)

Общественные библиотеки также могут быть с успехом оснащены терминальными компьютерами. Главные козыри – дешевизна терминалов и стандартный набор прикладных программ, а также очень простой способ увеличения количества клиентских компьютеров, то есть в итоге хорошая масштабируемость терминальных решений.

Интернет-кафе и подобные ему заведения безусловно могут использовать преимущества бездисковых компьютеров. Единственный здесь “минус” - это невозможность использования таких компьютеров для современных трехмерных игр. Хотя для игровых Интернет-салонов можно применить смешанный подход, то есть часть компьютеров будут традиционными персональными компьютерами и будут предназначаться для игр и работы в Интернете, а часть бездисковыми терминалами, которые предназначены только для работы с офисными пакетами и программами для работы в глобальной сети Интернет (веб-браузеры, клиенты электронной почты, ICQ, FTP-клиенты и т.п.).

Для государственных структур бездисковые компьютеры также являются прекрасным альтернативным решением, так как одной из положительных

сторон использования бездисковых компьютеров является повышенная надежность хранения информации, что для организаций подобного типа является очень важным условием. Не маловажным фактором является здесь и экономия средств налогоплательщиков, которые идут на обеспечение государственных заведений компьютерной техникой.

Коммунальные службы с их многочисленными абонентскими отделами также с успехом могут применять терминальные решения. Среда их работы итак максимально приближена к унифицированным рабочим местам, так как практически вся работа выполняется в одной программе. Переход на работу с бездисковыми компьютерами поможет сэкономить на аппаратном обеспечении, а также упростит процедуру администрирования клиентов локальной сети коммунального предприятия.

Поначалу я полагал, что для домашних пользователей идея бездисковых компьютеров не так уж привлекательна. Редкий человек в нашей стране имеет дома больше одного компьютера. А для бездисковых терминалов их нужно как минимум два (клиент и сервер). Но с другой стороны, кто живет один дома? Практически у всех есть семья, и я из своего опыта знаю, что бывают случаи, когда на один домашний компьютер претендует сразу два члена семьи. Покупка второго компьютера, как правило, не рассматривается именно из-за его приличной стоимости. Вот вам и ответ на использование терминалов дома – при ограниченных средствах, бездисковый "тонкий" клиент - это идеальный вариант второго домашнего персонального компьютера. Стоит он копейки (не считая монитора, конечно), ремонтировать и модернизировать его не нужно: протянул сетевой кабель и работай на здоровье. Конечно, чем больше количество домашних пользователей, тем эффективнее применение этой технологии удаленного доступа.

Кроме использования в определенных отраслях, у бездисковых компьютеров есть и преимущества для работы с определенными типами программного обеспечения. Если не углубляться в особенности реализации такого программного обеспечения, то ему свойственна относительно низкая доля операций с пользовательским интерфейсом. Другими словами,

взаимодействие с пользователем осуществляется через стандартизированный интерфейс (стандартные окна, элементы управления и т.д.), а уникальных интерфейсных элементов не очень много. Типичный пример подобного рода программы - текстовый редактор. Работая с ним, пользователь одновременно изменяет только малую часть рабочей области. Неподвижными остаются предыдущие и последующие абзацы текста, панели инструментов также меняются не значительно. Совсем другое дело графический редактор, в котором даже простая пользовательская команда может очень сильно изменить изображение на экране (например, масштабирование изображения или применение графического фильтра).

Следует также отметить, что некоторые программы выполняются эффективнее именно в терминальной среде, и эта эффективность не зависит от используемого пользовательского интерфейса. Простой пример: прикладные программы, которые используют базу данных с архитектурой файл-сервер. Так, в свое время, на территории нашей страны очень большой популярностью пользовались программные продукты типа FoxPro, DBase и Clipper. Так вот, для доступа к базе данных они используют файл-серверную архитектуру, а следовательно производительность удаленных клиентов всегда хуже чем у тех, кто использует базы данных локально. В терминальной среде все клиенты практически работают на одном компьютере (сервере), а следовательно производительность подобного рода прикладных программ у всех клиентов локальных и удаленных практически одинаковая.

Хотя в терминальной локальной сети нет ничего принципиально нового, ее стоит принимать в расчет при проектировании локальных компьютерных сетей и организации работы информационной структуры любого предприятия. Даже если терминальным решениям не будет отдано предпочтение, целесообразно рассматривать их как альтернативный, запасной вариант развития практически любой информационной системы.

Обзор существующих решений для сетевых компьютеров

Как вы уже наверное поняли, идея терминальный вычислений не нова, и в информационной отрасли есть целый ряд проверенных и надежных решений. Каждое из которых имеет как свои сильные стороны (надежность, функциональность, многоплатформенность) так и слабые (как правило, стоимость лицензии). Свое основное применение они находят там, где, с одной стороны, имеется информационная инфраструктура, а с другой стороны – желание уменьшить расходы связанные с поддержанием ее в работоспособном состоянии и повысить качество управляемости. В общем из представленных на рынке программных продуктов можно выделить такие:

Citrix MetaFrame

Citrix MetaFrame – наверное самый известный программный продукт данной серии (сайт http://www.citrix.com). В его основе лежит наиболее распространенная в информационной индустрии технология Citrix Independent Computing Architecture (ICA®), которая позволяет строить распределенные вычислительные среды, базируясь на различном аппаратном и программном обеспечении, а также используя разные каналы связи. В современном виде, данная технология позволяет запускать программы для Windows, UNIX и Java практически на любом устройстве и через любое подключение. В арсенале данного программного продукта удобные средства администрирования и централизованного управления распределенной сетевой средой, средства мониторинга и слежения за аппаратным обеспечением, защищенная и безопасная сетевая среда, а также возможность мобильного удаленного доступа к документам пользователей через Интернет. Фактически использование Citrix MetaFrame – это качественно новый уровень управления информационной средой, и это особенно заметно, если имеет место использование различных операционных систем в одной локальной сети. Другими словами, про преимущества использования Citrix MetaFrame можно рассказывать очень долго, но есть у данного проекта и существенные недостатки. На первом месте в списке негативных факторов идет конечно же цена лицензии на использование данного программного продукта, а она, поверьте, не маленькая. Это особенно существенно для нашей страны и организаций с ограниченным

бюджетом. Также не следует забывать и о том, что как любая сложная система, продукт от Citrix для успешного функционирования требует от администратора соответствующих знаний, опыта и специальной подготовки. Иначе говоря, Citrix MetaFrame – это хорошее надежное решение, но к сожалению оно далеко не всем по-карману.

Windows Terminal Services

Windows Terminal Services– программный продукт от софтверного гиганта Microsoft Corp. (сайт http://www.microsoft.com). Данное программное обеспечение предназначено для организации удаленного доступа к приложениям сервера от удаленных клиентов. В своей работе использует протокол RDP. Сервер должен работать под управлением операционных систем Windows NT Server 4.0 Terminal Server Edition, Windows 2000 Server или Windows 2003 Server. Клиентская часть может работать под управлением Windows 3.11 (16–разрядный клиент) и Windows 95, Windows 98, Windows NT Workstation 3.51, Windows NT Workstation 4.0, Windows 2000 Professional, Windows XP (32–разрядный клиент). Для того чтобы воспользоваться возможностью дистанционной работы при помощи Windows Terminal Services, необходимо правильно устанавливать приложения, в частности, в них должна быть предусмотрена совместная работа нескольких пользователей (многопользовательский режим), поэтому для запуска некоторых приложений может понадобиться специальный сценарий совместимости приложений. Terminal Services хорошо подходит для развертывания отраслевых приложений для удаленных пользователей, филиалов или устройств–терминалов и снижает совокупную стоимость владения. Поскольку модель Terminal Services обеспечивает централизованное управление приложениями, это помогает администраторам сократить затраты времени и труда на разработку клиентского программного обеспечения, его обслуживание и обновление. В среде Terminal Services никакие приложения или данные пользователей не хранятся на клиентских машинах, что позволяет вплотную контролировать систему безопасности и надежность. Terminal Services можно эксплуатировать и как средство удаленного администрирования Windows. Данная технология

лучше всего подходит для ненасыщенных графикой приложений, которые не используют мультимедийные возможности операционной системы (звук, видео и т.п.). В сравнении с решением от Citrix, Windows Terminal Services выглядит менее производительным и удобным программным продуктом, хотя опираясь на свою долгую историю развития, эта технология является реальным альтернативным способом работы удаленных пользователей терминалов.

VNC

VNC – это еще один очень интересный программный продукт для удаленной работы пользователей, разработанный компанией RealVNC (сайт http://www.vnc.com). VNC (Virtual Network Computing - виртуальные сетевые вычисления) представляет собой действительно многоплатформенную среду клиент-сервер. Начиная с 1998 года, когда исходный код этого программного продукта стал свободно распространяемым, с сайта компании его загрузило более 20 миллионов человек и многие популярные дистрибутивы операционной системы Linux также включают в себя данный программный продукт. Для того чтобы понять, что лежит в основе работы VNC, следует рассмотреть схему на рис. 1.3.

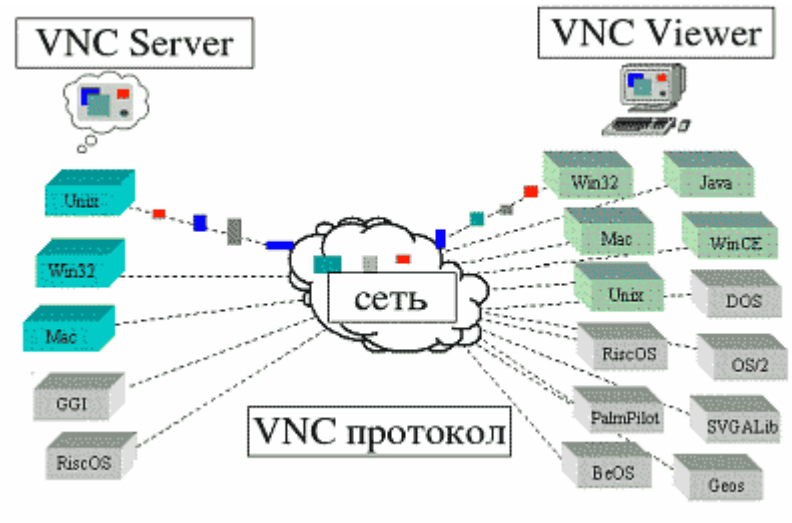

Рис. 1.3. Схема дистанционного доступа VNC

В сетевой вычислительной среде VNC присутствуют два понятия: VNC Server и VNC Viewer. VNC Server – это программа, которая работает на компьютере, что предоставляет удаленный доступ, а VNC Viewer – это клиентское приложение для работы с ресурсами удаленного сервера. Взаимодействие между этими двумя частями системы осуществляется

посредством специального VNC-протокола. В своей реализации протокол VNC представляет простую интерпретацию пользовательского графическому интерфейса. Он базируется на понятии удаленного фрейм-буфера, или как его еще иногда называют, RFB (remote framebuffer). Данный протокол позволяет серверу очень просто обновить изображение на клиенте, а поскольку идея фрейм-буфера может быть легко реализована в разных операционных системах и прикладных программах, то серверное и клиентское программное обеспечение VNC распространено на многих платформах. По сути, протокол VNC (он же, RFB-протокол), является протоколом для “тонкого” клиента, так как приложение VNC Viewer может работать на очень скромных аппаратных конфигурациях, включая карманные компьютеры (PDA).

При отображении и передаче графической информации по протоколу VNC используется понятие графических примитивов, таких как, например, прямоугольники. Но данный протокол выполняет дополнительную оптимизацию при передаче информации от сервера клиенту. Например, так как изображение фрейм-буфера хранится и на сервере и на клиенте, то при изменении информации на экране пересылается не все его изображение, а только те части, которые претерпели изменения. Подобным образом VNC протокол поступает и при передаче пиксельных изображений, то есть в зависимости от конкретной ситуации (аппаратного обеспечения, скорости соединения и т.п.), он выбирает схему кодирования графической информации. Например, неподвижные графические изображения могут передаваться в формате JPEG, а динамические видеоданные в формате MPEG. Современные интерфейсы операционных систем изобилуют графическими элементами, и рабочий стол пользователя часто напоминает по обилию красок новогоднюю елку, поэтому при передаче изображения от сервера клиенту, в сочетании используется очень много приемов и методов для уменьшения нагрузки на локальную сеть и повышения быстродействия. Очень часто при работе пользователя с графическим интерфейсом он выполняет действия, которые значительно изменяют изображение экрана, например, прокрутка содержимого окна. Но если рассмотреть этот процесс (прокрутку) более детально, то

окажется, что только незначительная часть экрана изменяется, а остальная просто сдвигается вверх или вниз (право/лево). Использование такой особенности выполнения прокрутки экрана также позволяет значительно снизить поток информации, передаваемой от сервера клиенту. В качестве примера аналогичной оптимизации следует сказать, что при передаче текста могут использоваться схемы кеширования изображений отдельных графических символов, так как внешне очертания букв одного шрифта и одного размера выглядят одинаково. Не следует однако забывать, что главным условием применения технологии оптимизации при передачи данных по протоколу VNC является наличие ее поддержки как со стороны сервера, так и со стороны клиента (VNC Viewer).

VNC протокол работает по принципу обновления фрейм-буфера, и инициатором этого обновления является приложение клиент (VNC Viewer). Под обновлением принято понимать замену старого изображения фрейм-буфера новым. А так как обновление всегда происходит по инициативе клиента, то VNC протокол смело можно назвать адаптивным, то есть отображение графических данных будет происходить с тем качеством, на которое рассчитывает клиент. Для медленного соединения или ограниченного в возможностях клиентского приложения скорость обновлений фрейм-буфера снижается, и тем самым падает нагрузка на локальную сеть, хотя качество работы (отображения графики приложением клиентом) снижается.

Протокол VNC содержит в себе также и ту часть, которая отвечает за передачу данных от устройств ввода клиента серверному приложению. При этом он базируется на стандартной клавиатуре и устройстве точечного ввода, таком как манипулятор “мышь”, но при желании система будет работать и с более специфическими устройствами. Данные пересылаются в случае нажатия на клавишу “мыши” или клавиатуры, а также при перемещении указателя “мышки” по экрану.

Протоколу VNC свойственны фазы установки соединения (setup) и его завершения (shutdown). При первом подключении клиента он проходит процедуру аутентификации, которая, как правило, заключается в передаче

имени пользователя и пароля, а также на скрытом от пользователя уровне происходит процесс согласования возможностей сервера и клиента, таких как разрешающая способность экрана, глубина цвета, схема кодирования графических образов и т.д. Исходя из природы соединений по протоколу VNC, которая не поддерживает понятие активного состояния, соединение может быть в любой момент прекращено клиентом, без какого-либо подтверждения на то сервером.

Проект VNC очень интересен с точки зрения его практического применения. Например, с его помощью вы можете администрировать удаленный сервер с Linux, работая на компьютере с операционной системой Microsoft Windows, или наоборот. Главное, что при этом вы не ограничиваете себя текстовой консолью, а можете наслаждаться всеми преимуществами графической среды. Использование клиентской программы VNC Viewer позволяет выполнять удаленные входы практически с любого компьютера. Отсутствие привязки к операционной системе и программному обеспечению делают этот проект действительно кроссплатформенным. Представьте себе ситуацию, когда принято решение перевести компьютеры с операционной системы Windows на Linux, но в организации используется программное обеспечение, несовместимое с Linux. В качестве компромиссного варианта можно использовать VNC для получения доступа к программам, которые не переведены в Linux. И таких примеров огромное множество. Хотя следует признать, что проект VNC не лишен изъянов. Так для работы VNC Viewer нужна операционная система, поэтому его сложно назвать полноценным “тонким” клиентом. Скорее всего, данное приложение следует рассматривать как способ кратковременного доступа к ресурсам удаленного компьютера, которое безусловно обладает широким спектром функциональных возможностей. Чтобы подчеркнуть большой потенциал проекта VNC, замечу, что на период написания этой книги были доступны сервера VNC для платформ: Windows 9x/2000/NT/XP (x86), Linux (x86), Solaris 2.5 (SPARC), HP-UX 11, Java, Novell Netware, OS/2, QNX и MacOS X, а клиенты VNC Viewer для еще большего числа платформ: Win32, Windows CE, Linux (разные

дистрибутивы с поддержкой X-Windows и даже SVGALIB), Amiga, BeOS, MacOS, FreeBSD, OpenBSD, NetBSD, CygWin32, Java, DOS, OS/2, PalmOS, Solaris, SCO, VMS, AIX и многих других.

NCD ThinPATH

NCD ThinPATH – интересный программный продукт для терминальных систем на базе серверов Microsoft Windows (сайт производителя http://www.ncd.com). В частности, данный программный продукт расширяет функциональные возможности бездисковых Windows терминалов и традиционных ПК при работе в удаленном режиме. Так, программное обеспечение NCD ThinPATH позволяет:

обеспечить поддержку локальных устройств для “тонких” клиентов, таких как, например, принтеры, через параллельный и последовательный порты;

использовать звук в работе приложений;

иметь доступ к локальным файлам на устройствах хранения информации, таких как жесткие диски, устройства чтения компакт-дисков и флоппи-дисководы;

расширить возможности таких технологий как Windows Terminal Services и Citrix ICA;

организовать доступ к локальным данным терминалов со стороны сервера.

Хотя NCD ThinPATH нельзя рассматривать как самодостаточную систему, следует отметить, что данное приложение является хорошим образцом технологического продукта, который расширяет возможности существующих на рынке программного обеспечения систем для работы удаленных пользователей.

NoMachine

NoMachine (сайт проекта http://www.nomachine.com) – проект итальянской компании Medialogic S.p.A. (сайт компании http://www.medialogic.it) представляет собой еще один вариант дистанционной работы (другое название !M). Как и в случае проекта VNC, данная разработка использует в своей работе два базовых компонента: виртуальный сервер (NX сервер) и клиент (NX клиент). NX сервер работает под управлением unix-подобных операционных

систем, хотя существуют экспериментальные сборки для платформы win32 (посредством использования CygWin). Клиент NX протестирован и может работать под управлением Windows 95/98/ME/NT/2000/2003/XP, Linux RedHat 8.0/9.0, Linux SuSE 8.1/8.2/9.0, Linux Mandrake 9.0/9.1, Linux Debian Stable (Woody) и SUN Solaris 9 (Sparc). NX клиенты для различных платформ выдержаны в одном стиле пользовательского интерфейса, что значительно облегчает их использование в различных программных средах. В зависимости от доступного сервера NX клиент может работать по протоколу RPD, VNC или даже использовать в своей работе X-протокол. Для повышения уровня безопасности используется SSH и SSL, что особенно актуально при работе через Интернет.

Проект NoMachine нельзя рассматривать как результат частного усовершенствования существующей технологии удаленного доступа к ресурсам сервера. В проекте NX мы имеем самодостаточный программный комплекс, который предоставляет полный спектр управления удаленным компьютером и даже такие функции, как доступ к общим каталогам и принтерам, а также поддержка мультимедиа.

Возможности проекта !M достаточно широки и не заканчиваются удаленным доступом и администрированием. Так, например, можно в одном сеансе Linux запустить из оконного менеджера KDE, окно Gnome сессии. Также удобно работать с приложениями Linux на ПК с Windows, при этом скорость работы значительно выше чем при использовании VNC. Тонкая настройка NX клиента позволяет использовать данную технологию на различных типах сетевых подключений, включая аналоговый модем, ISDN, ADSL, WAN и LAN, применять внешний сервер шрифтов и выбирать типы передачи графических изображений. На практике это означает, что производительность NX клиента можно оптимизировать для конкретной ситуации сеанса удаленного доступа. Благодаря проекту Thinstation (сайт http://thinstation.sourceforge.net) вы можете использовать NX клиент даже на бездисковых рабочих станциях.

Частично базируясь на технологиях OpenSource, NoMachine, тем не менее, остается коммерческим программным продуктом и, несмотря на доступность

клиентского программного обеспечения, за серверную версию вам все же придется заплатить.

Thinstation

Thinstation – довольно молодой и еще недостаточно стабильный проект (сайт http://thinstation.sourceforge.net) представляет собой операционную систему Linux, которая является базой для работы “тонкого” клиента (бездискового компьютера) с использованием таких протоколов как: Citrix ICA, MS Windows terminal services (RDP), Tarantella, X, telnet, tn5250, VMS term и SSH. Как видите, впечатляющая функциональность и при этом не стоит забывать, что загружаться такие “тонкие” клиенты могут как по сети, так и с локальных дискет, компакт-дисков, “винчестеров” и накопителей USB-Flash. В своей работе проект Thinstation не ограничивается ресурсами удаленного сервера, а может работать с локальными принтерами (LPT/USB) и устройствами хранения информации, такими как дискеты, компакт-диски, жесткие диски и флеш-накопители. Для уменьшения нагрузки на удаленный сервер и локальную сеть в проекте Thinstation активно используется запуск приложений на стороне клиента, что позволяет эффективно использовать ресурсы “тонкого” клиента для работы с такими популярными программными продуктами, как веб-браузер, медиа-плеер, калькулятор и т.п. Если вы серьезно задумываетесь над использованием бездисковых рабочих станций, то вам стоит более пристально взглянуть на проект Thinstation, тем более, что используя его, не обязательно работать в операционной системе Linux, а можно запускать только удаленные Windows сеансы (для этого понадобиться сервер с операционной системой от Microsoft).

LTSP

LTSP – эта аббревиатура расшифровывается как Linux Terminal Server Project и представляет собой систему удаленной работы для бездисковых компьютеров под управлением операционной системы Linux (официальный сайт http://www.ltsp.org). Так как в данной книге об этом проекте пойдет речь в ее третьей части, то не будем детально останавливаться на его достоинствах и недостатках. Если коротко, то проект LTSP можно охарактеризовать как

удачный симбиоз двух программных разработок: графической подсистемы X Window System и свободно распространяемой операционной системы Linux.

X terminal kit

X terminal kit – является простой интерпретацией идеи удаленной работы в операционной системе Linux (сайт http://www.solucorp.qc.ca/xterminals). Отличаясь спартанской лаконичностью и минимальными системными требованиями, эта технология может быть опробована в небольших организациях или домашними пользователями. Само понятие X-терминал раньше было связано с рабочими станциями от Sun Microsystems, но с выходом на рынок серверов операционной системы Linux мы имеем принципиально новое сочетание, которое без ограничений можно использовать на практике.

X-Windows и XDMCP

X-Windows и XDMCP- это сочетание изначально была рассчитано на удаленную работу пользователей. Структура X Window System позволяет задействовать все преимущества распределенной технологии клиент-сервер в графической среде. Распространенности X-Windows может позавидовать любой программный продукт, так как она портирована практически для всех существующих платформ. Правда есть у X-Windows и свои ограничения. Во-первых, X Window System не является операционной системой, а следовательно для ее работы на аппаратной базе требуется наличие последней. Во-вторых, несмотря на кроссплатформенность X-протокола, его конкретная реализация и использующие его приложения все равно привязаны к программному обеспечению и операционной системе, а следовательно переносимость X ограничивается только пользовательским интерфейсом. И в третьих, определенный архаизм, который сопровождает проект с 20-летней историей развития, также сдерживает использование данной безусловно перспективной технологии. Несмотря на то, что настроить удаленный сеанс в X-Windows не так и сложно (достаточно локально запустить X-сервер и в его параметрах указать адрес удаленного X-сервера, к которому необходимо подключиться), на практике пользователи сталкиваются с довольно серьезными

конфигурационными проблемами и без вспомогательных программных утилит такой подход сопряжен со значительными трудностями.

Приведенные выше программные комплексы, обеспечивающие удаленный доступ к серверным ресурсам, являются лишь вершиной айсберга, каплей в море сетевых вычислений, тонких клиентов и распределенных компьютерных сред. Однако они показывают, что данная область информационной технологии хотя и не доступна широкому кругу пользователей, все же успешно развивается и имеет миллионную аудиторию потребителей. Как и любая другая составляющая часть информационной технологии, она обеспечивает пользователей новыми возможности, которые в данном случае выражаются в более простом и надежном администрировании, а также снижении совокупной стоимости обслуживания одного автоматизированного рабочего места.

Достоинства и недостатки терминальных решений

Минусы

Как любое явление окружающего нас мира, бездисковый компьютер имеет как сильные так и слабые стороны. Начнем сначала с перечисления его слабостей. Главный недостаток – это отсутствие самодостаточности, то есть без локальной сети или в отсутствии сервера бездисковый ПК просто даже не загрузится. Тонкий клиент должен восприниматься как составная часть терминальной локальной сети и ни о каком существовании за ее пределами не может быть и речи. Другими словами, применение бездисковых компьютеров ограничено территорией распространения локальной сети с терминальным сервером, то есть дома вы вряд ли воспользуетесь преимуществами распределенных вычислений, так как для них одного персонального компьютера мало, нужен еще и сервер или высокоскоростной канал доступа к такому серверу. Из перечисленного выше кратко сформулируем первый и главный недостаток терминального компьютера: для работы нужен терминальный сервер, без подключения к нему терминал – груда металлолома.

Кинем взор в сторону администрирования терминальной локальной сети. Даже неподготовленному человеку ясно, что без администрирования она

работать не может. Это означает, что когда все настроено и отлажено, то работ по администрированию совсем немного, а пользователи просто включают свои терминалы и работают, но на первоначальном этапе внедрения “тонкие” клиенты требуют пристального внимания системных администраторов, особенно при этом "достается" их серверу. Поэтому к списку недостатков такого подхода запишем: требует администрирования.

Если бы обслуживание терминальной локальной сети заканчивалось бы простым администрированием, то это было бы пол беды. Проблема заключается в том, что администрировать нужно среду, которая значительно, я бы даже сказал принципиально, отличается от традиционной. Эта особенность предъявляет новые требования к обслуживающему персоналу, то есть к администраторам. Вполне вероятно, что внедрение бездисковых компьютеров потребует от системных администраторов повышения своей квалификации, а в ряде случаев даже освоения идеологически “враждебной” операционной системы. Конечно, речь не идет о каких то высших материях, недоступных пониманию простых смертных, но попотеть сисадмину придется - это точно. В итоге получаем третий немаловажный недостаток: качественно новое администрирование требует освоения новых областей знаний, а следовательно затрат времени, сил и средств.

В терминальной среде идет процесс централизации хранения и обработки информации. Несмотря на свою распределенную структуру, основная информация хранится и обрабатывается центральным сервером. При таком подходе все персональные терминалы становятся заложниками сервера не только в аппаратном, но и в информационном смысле. Представьте на секунду, что с информацией на сервере что-то случилось (например, “посыпался” жесткий диск). Все, приехали, пропали данные сразу всех пользователей! Если перефразировать известную пословицу, то в терминальной среде мы носим яйца (храним файлы) в одной корзине. Как результат, имеем еще один недостаток “тонких” клиентов – централизованное хранение информационных данных значительно увеличивает риск их потери и выдвигает серьезные требования к регулярному резервному копированию.

Если у вас еще нет в локальной сети выделенного сервера, тогда специально для вас появляется еще один недостаток - покупка дорогостоящего высокопроизводительного сервера. Без сервера вам не удастся использовать технологию сетевых компьютеров и обеспечить комфортную работу удаленных клиентов. Конечно, в образовательных целях в качестве сервера можно взять любой ПК, но для постоянной работы требуется наличие выделенного сервера с завышенными аппаратными требованиями. Кстати, этот сервер будет примером компьютера, которому практически любой конфигурации будет мало.

Так как в большинстве своем люди - законопослушные граждане, то они уже давно свыклись с мыслью, что за использование программного обеспечения, как правило, нужно платить. Программное обеспечение терминальной среды также не исключение из этого правила. Откройте любой каталог программных продуктов и ознакомьтесь со стоимостью приобретения лицензии на подобный софт. Эти числа, точнее их денежный эквивалент, обязательно нужно учитывать при принятии решения об использовании терминальной локальной сети. Брать в расчет следует также и будущее обновление программного обеспечения и возможность подключения дополнительных пользователей (стоимость добавочных лицензий), в случае расширения сферы деятельности. В общем минус налицо - затраты на приобретение программного обеспечения, его обновление и покупка дополнительных лицензий.

Все пользователи терминалов, и практически все запущенные на них программы выполняются на одном компьютере – терминал-сервере. Такое построение работы и совместное использование ресурсов приводят к возникновению пиков нагрузки на сервер, а следовательно замедлению работы всех пользователей удаленных терминалов. Конечно, увеличение нагрузки на сервер связано с особенностью одновременной работы пользователей, и если бы они работали по очереди, то вообще никаких проблем бы не было, но как объяснить это пользователям? Чтобы сгустить краски и отчетливей представить всю пагубность этой ситуации вообразите, что один пользователь терминала решил сыграть в “DOOM”, понятно: загрузка центрального процессора 100%,

но что делать в это время другим пользователям? Хорошо, в “DOOM” мы этому геймеру запретим играть в административном порядке, но задумайтесь так ли надежны операционные системы и программное обеспечение? Что мешает зависшей программе использовать 100% процессорного времени и доставлять неудобство остальным пользователям? Короче говоря, открытых и нерешенных вопросов в этом направлении много и еще один явный недостаток - совместная работа на одном компьютере всех пользователей сказывается на его производительности, причем пик нагрузки замедляет работу всех удаленных станций без исключения.

Сетевая загрузка предполагает распределение вычислений между терминалом (клиентом) и сервером. По сути, на терминале выполняется клиентская часть программного обеспечения, которая и обеспечивает доступ и взаимодействие с сервером. Чувствуете разницу между клиентской частью программного обеспечения терминала и полноценной операционной системой? В реальной жизни эта разница сказывается на возможности доступа к локальным устройствам терминала, его периферии. Безусловно, монитор, “мышка” и клавиатура работать будут без каких-либо трудностей, а вот с доступом к флоппи-дисководу, приводу для чтения компакт-дисков, принтеру, сканеру и т.п. придется повозиться. Конечно можно бесконечно дискутировать по поводу необходимости подключения графического планшета или другого экзотического устройства к терминальному рабочему месту, но факты - вещь упрямая, а по сему имеем сложности при доступе к локальным устройствам бездискового компьютера.

Как это не странно, но даже на рабочем месте пользователи воспринимают компьютеры как часть своей собственности. Даже таким виртуальным понятиям как “Рабочий стол” (имеется ввиду рабочий стол оконного менеджера) часто сопутствуют притяжательные местоимения: “А на моем рабочем столе ...” или “Ой, я тоже хочу на свой рабочий стол таку картинку!”. В принципе предосудительного в этом ничего нет, что плохого в том, что пользователь настраивает свой компьютер так, как ему удобно, но не всегда настройка под пользователя оправдана в среде терминалов. Как было сказано

ранее, у терминалов нет постоянного пользователя. То есть ввел свое имя и пароль, и работай со своими данными на любом терминале. Приведу в качестве иллюстрации реальный пример: ремонт в одной из комнат вынудил пользователя сесть работать за другой терминал, он набрал текст, отправил его на печать, а из принтера в ответ тишина. Нажал еще раз - результат аналогичный. Причина такого поведения принтера скрыта именно в настройках пользователя: он у себя в профиле настроил принтер по умолчанию, который остался в комнате закрытой на ремонт, и именно на этот принтер были направлены его задания сервером печати. Ясно, что в среде “тонких” клиентов привязка пользователя к рабочему месту не всегда гладко вписывается в “безликую” схему терминальных автоматизированных рабочих мест.

Так как без локальной сети работа сетевых компьютеров не мыслима, то к ней предъявляются завышенные требования по качеству и пропускной способности, но даже при наличии хорошо спроектированной локальной сети, среда терминалов интерпретирует передаваемую информацию по ней не так как это принято у традиционных персональных компьютеров. Приведу пример – прослушивание файла в формате mp3 через локальную сеть. При традиционном подходе нагрузка на локальную сеть будет ничтожной, так как в среднем плотность информационного потока в этом музыкальном формате находится на уровне 128 кбит/с. Но при запуске на удаленном бездисковом терминале проигрывателя mp3-файлов мы получаем совсем другой сетевой трафик, порядка 1500 кбит/с. Чем вызвано такое увеличение нагрузки на локальную сеть и почему эта нагрузка сохраняется даже при прослушивании несжатых аудиозаписей, хотя по логике должна еще больше увеличиваться. Ответ на этот вопрос содержится в клиент-серверной архитектуре терминальной локальной сети. Когда на терминале вы запустите mp3-плеер, то он будет работать на сервере, но звук то вы хотите слышать из колонок подключенных к терминалу, а следовательно по сети на терминал он будет передаваться не в mp3-формате, а в несжатом аудио. Также на терминале сложнее работать с видео и графикой, так как эти программы подразумевают перемещение значительных объемов видеоинформации по локальной сети с ее достаточно скромной пропускной

способностью. Отсюда вот вам еще один недостаток – локальная сеть – узкое место для больших объемов информации, и ее пропускная способность – одна из составляющих замедленной работы бездискового терминала по сравнению с традиционным персональным компьютером. Правда следует несколько реабилитировать пользователей терминалов в случае использования медиа-плееров. Существуют готовые решения которые позволяют запускать подобные программы локально, то есть использовать потенциал самих "тонких" клиентов и не тратить на музыку вычислительные ресурсы сервера и пропускную способность локальной сети.

Работа в терминальной среде выполняется при помощи обычных прикладных программ, но все же не все из них пригодны для работы на терминалах. Понятно, что из-за распределенной структуры вычислений некоторые программы работают лучше, а некоторые хуже, но следует признать также и тот факт, что некоторые программы работать не будут вообще. Это утверждение в первую очередь относится к таким программным продуктам, как игры. Даже при супер параметрах терминального сервера и гигабитной оптоволоконной локальной сети вам врядли удастся поиграть в современный 3D шутер на бездисковом “тонком” клиенте (с другой стороны, отсутствие игр в офисе – это мечта практически любого руководителя). Так же смело можно утверждать, что специальные утилиты для тонкого администрирования и настройки обычного ПК вряд ли подойдут в терминальном мире. А использование очень ресурсоемких приложений (трехмерное моделирование, обработка видео, компиляция программного обеспечения и т.п.) вообще лучше не использовать на терминалах, так как отобранные на выполнение этих задач ресурсы центрального процессора сервера негативно скажутся на общей производительности труда остальных пользователей терминальной локальной среды. Поэтому целесообразно смириться с тем, что часть программного обеспечения не пригодна для использования на терминалах.

Если в вашей локальной сети уже существовали средства учета сетевого трафика и велась его тарификация в разрезе IP-адресов, то с переносом информации на терминальную основу вы, скорее всего, столкнетесь с

необходимостью пересмотра методов учета и аудита локальной сети. Это связано с тем, что у терминальных решений все их пользователи работают практически на одном компьютере – сервере, и метод учета по IP-адресам не пройдет. Для правильной реализации системы автоматического слежения за использованием локальной сети нужно будет использовать одну из методик авторизации пользователей, а, как результат, имеем усложнение системы аудита сетевого трафика и его тарификации.

Да, негативных сторон довольно много. Но ни одна из них не является фатальной в развитии технологии сетевых компьютеров, так как этот подход имеет и ряд очень существенных достоинств.

Плюсы

Главным достоинством сетевого компьютера является его стоимость. При прочих равных условиях он всегда будет дешевле, чем полноценный традиционный персональный компьютер. Практика показывает, что терминалу не нужен ни мощный процессор, ни много оперативной памяти, без винчестера он также прекрасно обходится. Если речь идет о небольшом офисе, то экономия не выглядит убедительным аргументом, но там, где количество автоматизированных рабочих мест исчисляется десятками, а то и сотнями – факт уменьшения затрат на аппаратное обеспечение не вызывает сомнений. Поэтому реальный плюс терминальной среды – это удешевление стоимости одного рабочего места и суммарных затрат на компьютерную технику.

Если вы счастливый обладатель парка морально устаревших компьютеров, то для вас это станет огромным плюсом при переходе на терминальную основу, так как практически любой старый ПК можно переделать в терминал при минимальных затратах. Терминальный режим работы по сути продлит жизнь вашим компьютерам не на один год. И это при условии, что программное обеспечение будет установлено на них самое современное. Хотя терминальные решения возвращают нас как бы в прошлое (помните “супер” компьютеры размерами с комнату, в институтах в 70-х годах XX века), тем не менее они предоставляют реальный шанс использованию морально устаревшей техники для работы с современным программным обеспечением.

Из приведенного выше плюса вытекает еще один, влияние которого очень заметно в долгосрочной перспективе: нет необходимости выполнять постоянную модернизацию (апгрейд) клиентских компьютеров. В информационной технологии, чтобы получать преимущества от работы с современным программным обеспечением необходимо, чтобы и аппаратная часть ему соответствовала. Вспомните системные требования к компьютеру с операционной системой Windows 95: ей вполне хватало 486-го ПК с 8 Мб оперативной памяти, да и “винчестер” в 640 Мб ее также вполне устраивал. С появлением Windows 98 аппаратная “планка” была поднята выше, Windows 2000 – еще выше, и т.д. Исходя из такой тенденции парк компьютерной техники приходится постоянно менять, модернизировать, наращивать оперативную память, ставить быстрые большие жесткие диски и т.п., а все это заканчивается наличием постоянных затрат на обслуживание компьютерной техники. В случае использования терминальных компьютеров, они практически не нуждаются в модернизации, так как все, что нужно, за них делает сервер. Модернизация сервера конечно необходима, но во-первых, это один компьютер, а во-вторых, его характеристики рассчитываются всегда “с запасом”, чтобы обеспечить приемлемую продуктивность на несколько лет вперед.

Практически на 100% справедливо утверждение, что в терминале просто нечему ломаться. Бездисковый компьютер лишен механических деталей, а поэтому выход из строя какого-либо аппаратного компонента мало вероятен. Правда, в своей практике встречались клавиатуры, на которых за год работы стерлись клавиши и “мышки” с переломанными проводами, но, согласитесь, что такого рода поломку сложно считать серьезной. Их универсальность позволяет всегда содержать резервный терминал, пользователи которого появляются лишь на период замены “мышки” или монитора на их привычном рабочем месте. Поэтому у терминалов практически полностью отсутствует обслуживание и профилактический ремонт.

Очень сильной стороной терминалов является способность их к работе в условиях неблагоприятных для традиционных компьютеров. К таким факторам

относится жара, холод, вибрации, запыленность помешения и т.д. Конечно при -10 оС терминал вряд ли проработает долго, но запас прочности у него на порядок выше. Представьте себе ситуацию, когда в механическом цехе нужно поставить компьютер, а на обычном ПК “винчестер” из-за вибраций ломается раз в месяц. Терминал для этого идеальный случай, так как динамические вибрации ему нестрашны. Подобных примеров существует достаточно много, и поэтому смело можно утверждать, что терминал способен работать в неблагоприятных для традиционного персонального компьютера условиях.

Как уже упоминалось ранее, у терминала нет той “персональности”, которая всегда была присуща традиционному персональному компьютеру. По сути, это безликие однотипные рабочие станции, которые кроме внешнего вида, мало чем отличаются одна от другой. Пользователю должно быть абсолютно безразлично на каком терминале работать, где бы он ни ввел свое имя и пароль, он всегда получит привычное рабочее окружение. Такая универсальность позволяет пересмотреть подход к организации рабочих или учебных мест, оборудованных компьютерами. Например, в вашей организации 10 человек пользуются компьютерами, но не постоянно, а по мере необходимости. В терминальной среде каждый из них имеет свою учетную запись (имя пользователя и пароль). Следовательно, нет необходимости в закупке 10 терминалов, так как большую часть времени они будут попросту простаивать. В зависимости от интенсивности использования компьютеров вполне можно будет обойтись 5-6 равноценными терминалами. Универсальность терминалов позволяет также проще относиться к расположению компьютерной техники. Так как все они равнозначны, то полностью безразлично как и где их устанавливать, главное, чтобы по близости была локальная сеть. В результате имеем существенное преимущество в универсальном доступе к данным с любого терминального места. Конечно, вы можете сказать, что и на обычном персональном компьютере можно организовать подобный доступ и будете безусловно правы, но в терминальных решениях такая универсальная функциональность достигается значительно проще.

Только представьте себе, что все информация хранится в одном месте и управляется при помощи единой системы конфигурирования – это по сути мечта любого системного администратора. В терминальных средах все происходит именно так. Есть сервер, который нужно администрировать, и есть клиенты, которые в администрировании не нуждаются. Хранение всех данных на сервере значительно упрощает процесс управления такой средой. Программное обеспечение обновляется централизовано, доступ к пользовательским учетным записям также управляется сервером, общие каталоги, файлы все находится в одном месте. Поэтому, без сомнения, централизованность хранения информации приводит к упрощению процесса администрирования, что является не маловажным достоинством использования терминального сетевого окружения.

Терминальные решения жестко привязаны к каналам связи. Но производители программного обеспечения сделали большой шаг в сторону их экономного и главное рационального их использования. Для конечных пользователей это означает наличие возможности удаленной работы даже при низкой пропускной способности локальной сети интранет и даже глобальной Интернет. Сейчас существуют реальные терминальные решения, которые позволяют подключать к серверу пользователей, физически расположенных от него на очень большом расстоянии. То есть к положительным особенностям терминальной архитектуры следует добавить возможность подключения удаленных абонентов.

Свои недостатки терминальный подход может обратить в свои достоинства. Так, централизованная система хранения информации, с одной стороны, значительно увеличивает риск потери информации, а, с другой стороны, при соблюдении норм и правил систематического резервного копирования позволяет значительно улучшить ее надежность. Так, например, если резервные копии будут храниться в хорошо защищенном месте, то информации не страшны даже стихийные бедствия. Аргументы в сторону распределенного хранения данных на нескольких рабочих станциях в сети, также не выдерживают критики, поскольку в серьезных организациях всегда

существует потребность в совместном доступе к информации, а значит подразумевается наличие единого места ее хранения. Конечно, если не придерживаться стратегии резервного копирования, то выход из строя терминального сервера по своим масштабам очень близок к “падению” файл-сервера в традиционной локальной сети. С другой стороны, работа сервера терминалов отличается от работы рабочей станции (так, ему вполне присуща круглосуточная загруженность), а следовательно существуют приемы, которые позволяют выполнять полное или частичное резервное копирование информации в моменты простоя сервера (например, ночью, или в обеденный перерыв). В итоге, при правильном подходе вы не только получите надежную систему хранения важных корпоративных данных, но и сведете риск потери или разрушения данных практически к нулю.

Для иллюстрации повышенной надежности сохранности данных на терминалах приведу пример выключения электропитания при сохранении документа. Особенно эта ситуация актуальна для пользователей офисного пакета OpenOffice, так как сохраняет данные он довольно медленно. Так вот, представим, что пользователь терминала нажал на кнопку “Сохранить” и ждет завершения сохранения файла, а в этот момент пропадает электроэнергия. Что будет с файлом, который пытался сохранить пользователь? Для традиционного персонального компьютера ответ понятен: в следующий раз, когда пользователь включит компьютер, он увидит старый документ, каким он был до неудачного сохранения. В терминальной среде дела обстоят намного лучше, так как сохранение выполняет сервер (подключенный к источнику бесперебойного питания), и потеря контакта с клиентом не означает остановку выполнения его процессов (прикладных программ), так что файл будет успешно сохранен!

Несмотря на наличие большого числа коммерческого программного обеспечения для организации терминальных сред, в информационной индустрии нашлось место и для совершенно бесплатных проектов. Так, развитие движения за свободно распространяемое программное обеспечение поддерживаемое такими организациями как The Free Software Foundation (сайт

http://www.fsf.org) и Open Source Initiative (OSI) (сайт http://www.opensource.org) привело к появления достаточно надежных терминальных решений, например, на базе операционных систем типа Linux (проект Linux Terminal Server Project, детальную информацию смотрите на сайте http://www.ltsp.org). Конечно, противники свободного программного обеспечения всегда могут сказать, что OpenSource не может обеспечить надежность, разнообразие и качество, которое свойственно коммерческим разработкам, но с такой же степенью уверенности можно утверждать, что в целом ряде случаев открытость коллективной разработки и доступность исходного кода конечному потребителю позволяют выявить потенциальные проблемы производительности и безопасности, которые в коммерческих проектах могут остаться долгое время просто незамеченными. Немаловажным фактором остается и бюджетность любого технологического новшества. Хорошо, если организация располагает свободными финансовыми средствами, и готова потратить их на развитие своей информационной структуры. А как быть, если средств в обрез? Неужели отказываться от терминальных решений, выбрасывать старые персоналки и все равно тратить деньги на покупку новых компьютеров? Я лично приветствую развитие свободного программного обеспечения и не вижу ничего страшного в его интенсивном внедрении и использовании, так как на основании личного опыта давно убедился, что в некоторых случаях его надежность значительно выше, чем у коммерческих аналогов (яркий пример язык программирования Perl). По роду своей основной деятельности мне пришлось участвовать в частичном переводе локальной сети предприятия на работу в операционной системе Linux. По-началу у пользователей были значительные трудности с освоением нового программного продукта, особенно много нареканий было на OpenOffice (раньше у всех стоял Microsoft Office), по после 3-х месяцев работы в новых условиях все сотрудники предприятия успешно адаптировались и некоторым из них работа в Linux понравилась даже больше (особенно оценили пользователи функцию виртуальных рабочих столов, когда различные типы задач можно выполнять в разных виртуальных пространствах экрана). Поэтому дешевизна и доступность программного обеспечения терминальных локальных

систем является очень важным их преимуществом, и это особенно справедливо в отношении свободно распространяемого программного обеспечения под лицензиями GNU/GPL.

В информационном обществе существует стереотип, согласно которого терминалы всегда работают медленно. В качестве аргументов такого поведения приводятся факты сравнения пропускной способности локальной сети и локальных шин компьютера (FSB, PCI, AGP и т.д.). Как может терминал работать быстрее обычного персонального компьютера, когда при прочих равных он должен получить информацию с сервера, обработать ее и только потом отобразить на локальном мониторе. В таких размышлениях есть доля истинны, но не следует забывать, что прикладная программа, с которой работает пользователь терминала, фактически выполняется на сервере, а аппаратные возможности терминального сервера значительно превосходят характеристики среднестатистического персонального компьютера. Рассмотрим простой пример: вы запускаете текстовый редактор OpenOffice Writer в операционной системе Linux. В моем случае я сравнивал скорость запуска на терминале и традиционном ПК, так вот, обычная персоналка справилась за 15 секунд, а терминал - за 3. Как такое может быть, спросите вы? Ответ достаточно прост: терминал-сервер мощный компьютер, он имеет быстрый “винчестер”, много оперативной памяти и производительный центральный процессор, поэтому запуск программы на нем выполняется значительно быстрее, а пользователь терминала видит у себя на экране лишь результат работы сервера. И такое увеличение производительности наблюдается практически во всех операциях, будь-то открытие или закрытие файлов, печать на принтере, или выполнение сложных математических вычислений. Единственная область, где терминал проигрывает, это вывод значительных объемов графической информации на экран (например, работа с двухмерной графикой). Следовательно, при выполнении стандартного набора задач, на терминале наблюдается значительное повышение скорости работы прикладных программ, иными словами, пользователю просто становится удобней работать с ним.

Проектируя любую локальную сеть, нельзя не задумываться о ее будущем развитии и изменении структуры. В информационных технологиях все достаточно скоротечно, а поэтому всегда следует быть готовым к изменениям. Терминальные среды имеют, конечно, предел своих вычислительных возможностей, но между тем они отличаются гибкостью и масштабируемостью. Например, вы успешно внедрили терминальную сеть на 20 рабочих мест, но с ростом компании появилась необходимость в дополнительных 10 терминалах. Существующий терминальный сервер успешно справляется с новой нагрузкой, но его возможности не безграничны. Если понадобится добавить второй десяток компьютеров, то необязательно покупать новый сервер взамен старого, достаточно поставить его в дополнение к уже существующему. Тем самым вы сможете перераспределить нагрузку между 40 рабочими станциями в терминальной среде. В итоге, мы имеем случай гибкой системы с единым управлением, которая смело можеть записать в свои плюсы такой фактор как масштабируемость.

При использовании компьютеров мы часто забываем о негативном влиянии этих приборов на здоровье человека. Речь не идет о пресловутой радиации и излучении, здесь имеет место скорее совокупность факторов, таких, как качество монитора, удобство позы, в которой человек проводит свое время за компьютером, постоянный шум от работающего компьютера и т.д. Как было замечено раннее, терминал стоит значительно дешевле, чем традиционный персональный компьютер, а, следовательно, при покупке нового терминала можно больше средств потратить на хороший жидкокристаллический монитор, чтобы у пользователей не уставали от работы глаза. Также фирменные терминалы очень малы в размерах, что позволяет устанавливать их в непригодных для персональных компьютеров места, а стало быть, для пользователя остается больше свободного пространства, что немаловажно в "густонаселенных" офисах. Маломощные процессоры, которые стоят в терминалах, избавляют от необходимости использования активного охлаждения процессора (в большинстве случаев хватает обычного радиатора), также нет там и шумных “винчестеров”, а, как результат, уровень шума

значительно ниже чем у традиционного персонального компьютера. Поэтому, при равной стоимости аппаратного обеспечения персонального компьютера и терминала, у последнего качество и безопасность для человека будет на порядок выше.

Если вы на рабочем месте все время играете в игры, то не о какой коммерчески важной информации на вашем компьютере не может быть и речи (кстати, сообщите мне пожалуйста координаты вашего предприятия, а то я тоже хочу себе такую работу), но как правило, в любой организации есть что оберегать от посторонних глаз. Данные финансовой деятельности, бухгалтерия, содержание переговоров с поставщиками и заказчиками, вот далеко не полный список всего того, что хотелось бы сохранить в тайне, особенно от конкурентов. Вопросам защиты информации в последнее время уделяется значительное внимание, но в терминальных средах эту проблему решить проще и вот почему. Все локальные компьютеры практически лишены данных. Даже если злоумышленники вынесут терминал из вашего здания, то ни к какой информации они не получат доступ, так как на терминале ее просто нет, а хранится она на сервере и аутентификация пользователей выполняется там же. Согласитесь, что обеспечить безопасность одного компьютера (терминального сервера) значительно легче, чем каждого рабочего места в отдельности. Поэтому в терминальных компьютерных сетях защищать информацию от несанкционированного доступа значительно проще.

Только человек, поработавший на многопользовательском “персональном” компьютере, знает, как важна конфиденциальность личных данных. Например, ты весь день старался, печатал квартальный отчет, а какой-то нерадивый коллега взял и стер всю папку вместе с результатами твоего титанического труда. В терминальных средах к разграничению пользовательских данных подходят значительно серьезней, так как отсутствие такого контроля может привести к выходу всей системы из строя. А, стало быть, работая на терминале, можно не бояться того, что ваши личные данные станут достоянием ваших коллег. Конечно, остаются системные администраторы, но куда ж без них. Только представьте, какие это открывает новые горизонты для компьютерных

классов в учебных заведениях и Интернет клубах. Например, в университете, каждому учащемуся присваивается логин и пароль для работы в компьютерном классе. Автоматически отпадает необходимость в дискетах, так как ваши персональные данные будут мирно ждать вас в вашей личной папке, а в сочетании с универсальностью терминалов, будет даже безразлично за какой из компьютеров сесть работать. Скептики, конечно, могут возразить, что сложно будет управлять системой с тысячью пользовательскими профилями, но на помощь всегда придут дисковые квоты и средства администрирования. В общем итоге получаем, что разграничение личных и общих данных в терминальной среде осуществляется проще, удобнее и с большей степенью безопасности.

В одноранговых локальных сетях, с появлением терминального сервера, у пользователей появляется мощный и практически постоянно включенный компьютер. Так почему бы не нагрузить его дополнительной работой? Терминал-сервер может одновременно быть еще и файл-сервером, сервером печати, интернет-сервером, контроллером домена, DNS-сервером, почтовым сервером, прокси-сервером, ftp-сервером и даже веб-сервером. Конечно, можно поспорить о целесообразности совмещения всевозможных серверных приложений на одном компьютере, но почему нет? С другой стороны, терминал-сервер может заняться параллельно и другой полезной работой, например, учетом статистики телефонных переговоров офисной АТС. В любом случае на терминал сервер можно возложить выполнение большого количества дополнительных задач.

Обобщая недостатки и преимущества терминальных компьютерных сетей, хочется еще раз остановиться на вопросе сферы применения данной технологии. Положительное или отрицательное восприятие бездисковых сетевых компьютеров зависит от угла зрения на данную проблему. Чтобы успешно применять технологию “тонких” клиентов на практике, давайте с энтузиазмом смотреть на присущие ей преимущества, но в то же время не забывать о существующих трудностях и подводных камнях использования терминалов.

Чтобы сделать сравнение положительных и отрицательных сторон использования терминального подхода, ниже в таблице приведена обобщающая схема “За” и “Против”:

За	Против
удешевление стоимости одного рабочего места и суммарных затрат на компьютерную технику	без подключения к серверу терминал – груда металлолома
использование морально устаревшей техники для работы с современным программным обеспечением – по сути, “вторая жизнь” ваших старых компьютеров	требует администрирования
нет необходимости выполнять постоянную модернизацию (апгрейд) клиентских компьютеров	качественно новое администрирование требует освоения новых областей знаний, а, следовательно, затрат времени, сил и средств
практически полностью отсутствует обслуживание и профилактический ремонт (там просто нечему ломаться)	централизованное хранение данных информации значительно увеличивает риск ее потери и выдвигает серьезные требования к регулярному резервному копированию
терминал способен работать в неблагоприятных для традиционного персонального компьютера условиях	покупка дорогостоящего высокопроизводительного сервера
универсальный доступ к данным с любого терминала	затраты на приобретение программного обеспечения, его обновление и покупка дополнительных лицензий

централизованность хранения информации приводит к упрощению процесса администрирования	совместная работа на одном компьютере всех пользователей сказывается на его производительности, причем пик нагрузки замедляет работу всех удаленных станций без исключения
возможность подключения удаленных абонентов	сложности при доступе к локальным устройствам бездискового компьютера
надежная система хранения важных корпоративных данных, а также минимальный риск потери или разрушения данных	привязка пользователя к рабочему месту не всегда гладко вписывается в “безликую” схему терминальных автоматизированных рабочих мест
дешевизна и доступность программного обеспечения терминальных локальных систем	локальная сеть – узкое место для больших объемов информации, и ее пропускная способность - составляющая замедленной работы бездискового терминала по сравнению с традиционным ПК
значительное повышение скорости работы прикладных программ	часть программного обеспечения не пригодна для использования на терминалах
масштабируемость (легко увеличивается количество клиентов и мощность сервера)	усложнение системы аудита сетевого трафика и его тарификации
качество и безопасность терминалов для человека на порядок выше, чему обычных ПК	
защищать информацию от несанкционированного доступа значительно проще	

разграничение личных и общих данных в терминальной среде осуществляется проще, удобнее и с большей степенью безопасности	
на терминал сервер можно возложить выполнение большого количества дополнительных задач	

Общие сведения о используемых протоколах (TFTP, DHCP, NFS, UDP, TCP/IP, SSH)

Работа сетевых протоколов не зависит от нашего знания о них. Но если вы хорошо разбираетесь в принципах их функционирования, то это позволит настроить их на оптимальную производительность в соответствии с вашими потребностями. В любом случае, базовые теоретические знания не помешают. В данной главе я хочу бегло познакомить вас с основными сетевыми протоколами, которые применяются для обеспечения работы терминальных решений. Основное внимание здесь уделяется протоколам с открытыми стандартами, которые получили наибольшее распространение. Начнем наш краткий экскурс в сетевые протоколы с рассмотрения такого обобщенного понятия как семейство протоколов TCP/IP. В него входят:

IP

IP (Internet Protocol – протокол Интернета) был первоначально разработан в конце 70-х годов и, конечно, не был рассчитан на такую популярность. Главным ограничением протокола IPv4, как его сейчас принято называть, является ограниченность количества узлов сети (хостов). Так, для IPv4 максимальное теоретическое количество хостов не может превышать 4 миллиардов (232), а на практике это значение еще меньше. Исходя из такого ограничения, был разработан новый стандарт протокола IPv6, который уже утвержден, но, к сожалению, еще не получил широкого распространения. В новой версии протокола IP используются более длинные адреса в 16 байт (вместо 4 для IPv4), применяется гибкий формат заголовка, поддерживается

резервирование пропускной способности, а также внедрена поддержка расширяемости протокола (добавление новых функций).

ICMP

ICMP (Internet Control Message Protocol– межсетевой протокол управления сообщениями) предназначен для передачи низкоуровневых сообщений, например, сообщений об ошибках. Наиболее распространенное применение протокола ICMP – это, наверное, утилита ping, при помощи которой можно проверить функционирование сетевого интерфейса. Как и многие другие сервисные средства, ICMP стал инструментом хакерских атак на узлы Интернет, поэтому знание данного протокола может особенно пригодиться для защиты локальной сети от внешнего вторжения.

ARP

ARP (Address Resolution Protocol – протокол преобразования адресов) служит для определения соответствия MAC-адресов IP-адресам. Применяется ARP только в пределах одной сети.

RARP

RARP (Reverse Address Resolution Protocol – обратный протокол преобразования адресов) служит для определения соответствия IP-адреса MAC-адресу сетевого устройства. По сути, он выполняет операцию, обратную протоколу ARP. Как и для ARP, RARP применяется только в пределах одной сети.

TCP

TCP (Transmission Control Protocol – протокол управления передачей). TCP – это протокол транспортного уровня, который базируется на протоколе IP и обеспечивает надежную передачу данных с предварительным установлением связи между источником информации и ее приемником. При передаче данных по протоколу TCP создается канал связи путем передачи запроса и получения утвердительного ответа. Важно помнить, что по окончании передачи информации сеанс должен быть явно завершен. Канал передачи имеет повышенную надежность, так как приемник отсылает подтверждающий пакет о успешном получении данных. В возможностях протокола TCP заложены

функции управления потоком данных, а также доставка данных исключительной важности. Конечно, за надежность протокола TCP пользователь расплачивается уменьшенной скоростью передачи информации.

UDP

UDP (User Datagram Protocol – протокол пользовательских датаграмм), так же как и протокол TCP, обеспечивает пересылку данных (то есть является протоколом транспортного уровня), но в отличие от последнего он не выполняет проверку получения данных. Для подтверждения получения данных должны уже заботятся протоколы уровня приложений, которые используют данный транспортный протокол. По сравнению с TCP, UDP-протокол обеспечивает более высокую скорость передачи данных, но не гарантирует надежность доставки. Применяется UDP при передачи звука и видео через Интернет (если допускаются искажения информации, но важна скорость), сервером DNS, протоколом NTP (синхронизация времени через Интернет), а также протоколом передачи файлов TFTP.

Совершенно другую группу представляют протоколы уровня приложений. В организации распределенных терминальных компьютерных сетей применяют в частности такие протоколы, как NFS, TFTP, DHCP, SSH. Эти протоколы являются протоколами более высокого уровня и в своей работе полагаются на уже существующие протоколы транспортного уровня, такие как TCP/IP и UDP/IP.

NFS

NFS (Network File System – сетевая файловая система). Этот протокол был разработан фирмой Sun Microsystems (сайт http://www.sun.com) для обеспечения доступа к файлам и каталогам удаленного сервера. В основном используется UNIX-подобными операционными системами, хотя существуют его реализации и для других ОС. В терминальных решениях получил распространение благодаря возможности удаленной загрузки операционной системы через локальную сеть. В качестве транспортного уровня применяется протокол UDP или реже TCP. Принцип использования данного протокола довольно прост: на сервере выделяются каталоги для экспортирования (общего

использования) в локальной сети. К сожалению, данный протокол лишен средств защиты от несанкционированного доступа и не имеет в себе способов аутентификации пользователей. Разграничение доступа осуществляется только по доменным именам или IP-адресам компьютеров в сети. Пользователи принимаются по их идентификационным номерам, что требует дополнительного использования таких средств как NIS или LDAP. Попробую объяснить это более детально. Представьте ситуацию, что на сервере NFS экспортируется домашний каталог /home/masha пользователя masha, с идентификатором пользователя UID=507. Проблема безопасности в модели доступа NFS заключается в том, что если подключается компьютер с пользователем petya, у которого такой же UID=507, то он получит полный доступ к домашнему каталогу /home/masha, так как сервер будет считать, что это пользователь masha. Исходя из таких ограничений, NFS можно применять только в надежных сетях, то есть там, где вы точно доверяете пользователю удаленного компьютера.

TFTP

TFTP (Trivial File Transfer Protocol – тривиальный протокол передачи файлов) предназначен для передачи файлов по сети с использованием транспортного протокола UDP. В терминальных сетях он применяется для начальной загрузки ядра операционной системы. Использование в качестве транспортного протокола UDP накладывает определенные ограничения на качество соединения, но зато позволяет сделать программную реализацию протокола TFTP очень маленького размера. Например, загрузочная программа может поместиться в 8 Кбайт памяти BootROM сетевой карты. Конечно, столь малые размеры и простоту данного протокола компенсирует его низкая безопасность, поэтому применять данный протокол также рекомендуется только в проверенных сетях.

DHCP

DHCP (Dynamic Host Configuration Protocol – протокол динамического конфигурирования хостов). Данный протокол предназначен для передачи уникальных параметров конфигурации от сервера DHCP клиентам. К числу

таких параметров относится, например, IP-адрес сетевого адаптера, который часто привязан к MAC-адресу. В терминальных решениях на данный протокол возлагают также обязанности передачи клиенту дополнительных данных, таких, как название и расположение загрузочного образа операционной системы, специфические параметры работы с сетевой картой или параметры удаленной загрузки. Практически незаменимым данный протокол является в больших локальный сетях, так как ручное конфигурирование сетевых интерфейсов превращается в них в сущий кошмар.

SSH

SSH (Secure SHell – протокол защищенной оболочки) предназначен для выполнения команд на удаленной машине. Механизм аутентификации этого протокола обладает повышенной безопасностью, так как данные в зашифрованном виде передаются через общедоступную сеть. Кроме этого, данный протокол поддерживает сжатие пересылаемых данных и обработку графической информации через поддержку протокола X-Windows. Несмотря на то, что программное обеспечения SSH защищено патентами, существует его свободная реализация OpenSSH, которая обладает практически всеми преимуществами коммерческой версии данного программного продукта.

Администрирование и сопровождение терминальной инфраструктуры

Как наверное всегда, с одной стороны, администрирование терминальной инфраструктуры проще, а с другой – сложнее.

Начнем обзор сложности администрирования бездисковых рабочих станций. Как уже было сказано выше, сам терминал в обслуживании практически не нуждается, так как никакой полезной информации в нем нет, а следовательно, ломаться там просто нечему. Для внесения изменений в параметры работы бездискового терминала используется программное обеспечение сервера. Главная сложность при этом состоит в том, что сервер нельзя выключать и перегружать, так как это может отразиться на работе других пользователей: их терминалы просто выключатся. Эта ключевая, на мой взгляд особенность, заставляет системных администраторов по другому осуществлять свою деятельность. Например, обычно системный администратор

начинает вносить изменения в систему когда что-то перестает работать. В среде терминалов ему нужно заранее планировать все свои мероприятия так, чтобы внесения изменений не затронули работающих пользователей. Один из вариантов решения данной проблемы - перенос рабочего дня администратора на выходной день и применение средств автоматического реконфигурирования в нерабочее время (например, выполнение задач crontab).

Второй сложностью, которая подстерегает системного администратора в терминальной среде, это одновременная работа большого числа пользователей на одном компьютере. При неправильной настройке операционной системы и прикладного программного обеспечения может сложиться ситуация, когда один процесс будет потреблять все свободные ресурсы терминал-сервера и превратит работу остальных пользователей в настоящий ужас. В руках системного администратора должны быть инструменты, которые своевременно отлавливают такие программы и либо понижают их приоритет, либо вообще прекращают их работу.

Время от времени в программных продуктах находятся ошибки, которые приводят к аварийному завершению программных процессов. Операционная система не всегда в состоянии определить причину таких ошибок, и иногда “повисшие” в памяти процессы продолжают жить еще долгое время, потребляя ресурсы центрального процессора и оперативную память компьютера. Следовательно, системные администраторы серверов должны своевременно выявлять такие программы и освобождать используемые ими ресурсы. Хорошо было бы настроить эту работу по сборке “мусора” полностью в автоматическом режиме без человеческого вмешательства. Но иногда вам все же придется в ручную проверять результаты работы такой системы, хотя бы из соображений контроля.

Также не следует забывать и о удаленных пользователях терминалов. Установка программного обеспечения должна всегда быть функциональной обязанностью и быть под контролем только системного администратора. Если разрешить пользователям самим устанавливать нужные им программы, то можно с уверенностью утверждать, что через некоторое время терминальная

система превратиться в хаос. В традиционных локальных сетях такой большой опасности от действий рядового пользователя нет, так как при соблюдении жестких мер системной безопасности, максимум, что он может сделать – это навредить самому себе, поломав компьютер на своем рабочем месте (хотя, если он будет запускать программы содержащие вирусы, достанется всей локальной сети). При терминальном способе подключения компьютеров от устанавливаемых в системе программ зависит стабильность работы всех пользователей, поэтому делегировать полномочия по установке программных продуктов не рекомендуется ни в коем случае.

Все терминальные решения без исключений используют локальную сеть в качестве транспортного уровня, и без ее надежной работы отказываются функционировать. Такая зависимость от постоянного подключения к локальной сети создает целый ряд дополнительных проблем в жизни рядового сисадмина. Так, в моей практике был случай, когда пользователь компьютера вытащил из гнезда сетевой карты сетевой кабель (RJ-45) и пытался засунуть туда телефонный провод (RJ-12), и самое удивительное, что у него это получилось! Обнаружил я это лишь после того, как этот же пользователь озадачил меня вопросом: “Почему я на сетевом принтере напечатать ничего не могу?”. Если бы он работал на терминале, то извлечение сетевого кабеля привело бы к немедленной остановке его работы, так как связь с сервером была бы потеряна. Вспомните также о любимой привычке пользователей и их непосредственных руководителей постоянно менять расположение компьютеров, мебели, и даже рабочих кабинетов. А ведь первым условием расположения терминала – является наличие в этом месте именно подключения к локальной сети, соединяющей его с терминал-сервером.

Вообще, работа системных администраторов (не только тех, кто обслуживает терминальные системы) схожа с функционированием коммунальных служб. Пока все работает, про них не вспоминают, а как только что-то перестало функционировать: крики, вопли где сисадмин. Конечно, и сам системный администратор тоже должен осознавать эту схожеть, и постоянно выполнять профилактические мероприятия, планировать свои действия

наперед, предвидеть возможные потенциальные проблемы. Так как известно, что если долго не менять трубу, рано или поздно случится авария. В одной книге, посвященной системному администрированию операционной системы Linux, я встретил цитату человека, уже не первый год занимающегося выполнением функций системного администратора. Так вот, на вопрос, что его привлекает в этой работе он ответил: “Если бы я хотел спокойную работу, то стал бы авиадиспетчером.” Доля истины есть в его словах, так как в информационных технологиях все очень быстротечно, и от системного администратора требуется способность к быстрому принятию правильных решений, а также наличие способностей прогнозирования и выявления узких мест информационной системы, а также их оперативная ликвидация, причем в идеальном варианте, еще до того времени, когда они станут очевидны пользователям.

Но не будем о грустном. У администрирования терминальных решений есть и много плюсов. Главный из них, возможность концентрации усилий фактически только на одном компьютере – терминал сервере. Терминалы сами по себе, довольно безобидные и “безмозглые” создания, и воспринимать их нужно не более чем связку сетевого адаптера, монитора, “мышки” и клавиатуры. Основная работа выполняется на сервере, а следовательно, и администрировать нужно только его.

Упрощается и работа системного администратора с аппаратным обеспечением клиентов, так как вышедший из строя терминал заменяется довольно крупными элементами (монитор, клавиатура, “мышь” или системный блок). Имея в своем распоряжении запасной терминал, можно всегда оперативно реагировать на отказ работы клиентского оборудования. А если в отделе много терминалов, то пользователи для работы могут использовать любые из них. При замене оборудования, не нужно думать о переносе локальных файлов, восстановлении поврежденной информации с жестких дисков, и тому подобных ужасах, так как все данные хранятся на сервереи установка или использование нового терминала позволяет пользователю получить мгновенный доступ к его личным данным.

Средства администрирования терминальных систем, как правило, находятся на более высоком уровне, чем при использовании локальной сети традиционных персональных компьютеров. У системных администраторов под рукой всегда есть утилиты доступа к удаленным экранам пользователей и для того, чтобы ответить на вопрос конкретного пользователя, позвонившего по телефону, необязательно бежать через все здание. К тому же в терминальных решениях все та же централизованность значительно облегчает функции установки программных продуктов, их обновление и обслуживание. Так, установленная один раз программа, стает доступной для всех пользователей терминалов, и нет необходимости выполнять эту рутинную операцию для каждого клиента в отдельности.

Перед тем как “пересаживать” пользователей на терминалы, я бы порекомендовал любому системному администратору сначала на себе испытать все достоинства и недостатки терминального подхода, так как приобретенный таким способом опыт поможет ему в будущем лучше понимать пользователей и быстрее находить решения их, а следовательно, и системных проблем. Лично мне работа за терминалом нравиться по нескольким признакам. Во-первых, это работа в одной среде с пользователями, то есть существует возможность наиболее быстрого реагирования на их запросы. Во-вторых, терминал очень быстрый компьютер, а следовательно работать за ним удобно и это особенно справедливо для ресурсоемких приложений. В-третьих, отсутствие таких шумных деталей как "винчестер" и вентилятор к нему, делают рабочее место по-настоящему тихим. В-четвертых, с экономической точки зрения, использование терминалов это самый эффективный способ эксплуатации компьютерной техники, когда на одно рабочее место выделяется минимум средств. И наконец в-пятых, внедрение терминальной структуры позволяет более рационально использовать время системного администратора, а стало быть мое время.

Часть II. X-терминалы

Глава 2. Общие сведения о X-терминалах

Как это работает (краткое описание процесса работы и загрузки)

Из первой части книги вы узнали как и зачем появились бездисковые клиенты и откуда взялось понятие удаленной загрузки. Если немного пофантазировать в этом направлении, то в недалеком будущем мы вполне можем иметь ситуацию, когда просто не останется обычных персональных компьютеров. Представьте себе, что придя домой или на работу в офис, вы включаете свой компьютер не в локальную сеть, а сразу в глобальный Интернет (или как он там будет в будущем называться). Интернет к тому времени станет таким всепоглощающим, что заменит собой все телекоммуникации. То есть вместо телевидения и телефонной связи останется одна глобальная компьютерная сеть, причем такой пропускной способности, что просто не будет необходимости в жестких дисках, в локальном хранении информации отпадет надобность. Теперь хоть дома, хоть на работе вы будете имееть возможность работать со своими документами, более того вам не нужно заботиться о модернизации своего компьютера, так как всю информацию будет обрабатывать удаленный сервер, а ваш персональный компьютер только посылать ему команды и принимать от него ответы. Фактически главными

компонентами компьютера станут монитор, клавиатура и “мышка” (скорее всего аналогичные устройства из будущего, такие как голографический монитор и устройства речевого ввода). К тому времени все программное обеспечение будет бесплатным, так как информация должна быть свободно доступной, а корпорация Microsoft объявит о своем банкротстве ... Так, стоп. Что-то я чересчур расфантазировался. Однако если тщательно всмотреться в компьютер будущего из моей фантазии, то в принципиальном плане он очень сильно напоминает X-терминал. А так как теоретически вы знаете на что это должно быть похоже, не будем останавливаться на пройденном и перейдем к реальному использованию X-терминалов с операционной системой Linux.

Работа X-терминала в Linux несколько отличается от работы бездисковых компьютеров в других операционных системах. Для использования X-терминальных решений требуется: настроенный X-терминал сервер, локальная сеть и X-терминал клиенты. Когда пользователь включает X-терминал (клиентский компьютер) на нем автоматически запускается загрузочная программа, которая рассылает широковещательные запросы с целью получения информации от сервера (IP-адрес, ядро для загрузки по сети и т.п.). Сервер X-терминалов откликается на этот запрос и отсылает требуемую информацию X-терминал клиенту. Далее через локальную сеть на X-терминал клиент загружается с сервера, а затем и запускается ядро Linux, скомпилированное особым образом. Ядро выполняет на X-терминал клиенте привычную для него работу: инициализирует устройства, распределяет память, протоколирует все системные сообщения и т.д. После того как ядро полностью вступило во владение X-терминалом, инициализирующий скрипт монтирует по сети корневую файловую систему сервера. То есть фактически вместо локального жесткого диска в распоряжение X-терминала поступает жесткий диск (диски) сервера. Теперь именно с сервера загружается X-сервер системы X-Windows, но выполняется он на локальном оборудовании X-терминала. Затем он подключается к X-серверу терминального сервера для удаленного запуска приложений.

Этот процесс удаленных подключений необходимо пояснить более детально. Например, вы работая на X-терминале запустили браузер Mozilla. Реально происходит следующее: Mozilla запустился на сервере, но свои окошки через X-протокол он отрисовывает на X-терминале клиента. Этот пример и объясняет низкие требования к аппаратному обеспечению оборудования X-терминала. Основная работа X-терминала фактически заключается в обработке сетевых подключений и отрисовке на экране программных окошек. Дисковое пространство, оперативная память, использование центрального процессора – все это X-терминал берет от сервера. Практически, X-терминал - это реальное воплощение давней мечта всех пользователей персональных компьютеров, которые задавали вопрос: "Как подключить к компьютеру второй монитор и вторую клавиатуру?".

Описать насколько продуктивно работает X-терминал однозначно не получиться. Лучше один раз увидеть (например, я пишу эти строки используя X-терминал, а реальный пример внедрения X-терминальной сети вы найдете в приложении). Если быть кратким, то X-терминал имеет легкий системный блок, быстро загружается и не боится перебоев в электропитании. Практически все программы запускаются на нем также как и на обычном персональном компьютере (исключение составляют программные пакеты интенсивно использующие графику, например, Gimp). Так как программы выполняются на сервере, а аппаратные возможности сервера значительно превосходят параметры рядового персонального компьютера, то получается, что программы на X-терминале выполняются быстрее, а следовательно работать на нем удобнее.

Не всегда нужно использовать удаленную загрузку именно для X-терминалов работающих в графическом режиме. Терминал можно использовать и в текстовом режиме, правда в таком роле он больше подходит для функций системного администрирования нежели для обычной работы (а если признаться откровенно, то администрировать из X-Windows удобнее: запустил xterm и вперед).

Многие могут подумать, что для работы X-терминалу требуется локальная сеть супер качества. Сразу скажу, что это не так, но желательно наличие хорошей локальной сети Fast Ethernet 100 Мбит/с (хотя в моей практике X-терминалы работали и по коаксиальному кабелю 10 Мбит/с в полудуплексном режиме “Half Duplex”). Для того, чтобы проверить зависимость X-терминала от помех в локальной сети, я для эксперимента взял и отсоединил работающий X-терминал от сетевого кабеля. Как результат на экране X-терминала все замерло, только мышка продолжала реагировать на действия пользователя. Через пару минут я подключил сетевой кабель обратно. После 10 секундной паузы X-терминал продолжил работу как ни в чем не бывало. Для очистки совести должен признаться, что если отключить работающий X-терминал от сети на очень длительное время, то шанс его "реанимировать" простым подключением к локальной сети достаточно невелик. Скорее всего в этом случае лучше поможет его перезагрузка.

Удаленный запуск консольных программ

Если у вас возникли вопросы касательно надежности и эффективности удаленного запуска приложений, сейчас самое время проверить их в действии. В принципе, проверить удаленный запуск можно и на одном компьютере через сетевой интерфейс обратной петли (lo), но лучше все же задействовать локальную сеть, а следовательно для проверки нам понадобятся два компьютера.

Самая на мой взгляд незаслуженно распространенная программа удаленного доступа – это telnet. Именно незаслуженно, потому что она передает по сети данные (в частности имя пользователя и пароль) в не зашифрованном виде. Имея такую огромную дыру в безопасности она быстро нашла своего безопасного приемника SSH (Secure SHell – защищенная оболочка). На самом деле ssh скорее замена для rlogin и rsh, но не будем придираться к словам, а перейдем к практике. Именно на примере ssh и рассмотрим пример запуска удаленных приложений.

Один из компьютеров будет сервером, а другой клиентом. На сервере необходимо проверить установку серверных пакетов свободной реализации

SSH – OpenSSH. Делаем это командой (примеры данной главы ориентированы на дистрибутивы операционной системы Linux, которые используют менеджер пакетов RPM, в частности ASPLinux 9.0):

```
$ rpm -qa | grep ssh
openssh-clients-3.6.1p1-1.1asp
openssh-3.6.1p1-1.1asp
openssh-askpass-gnome-3.6.1p1-1.1asp
openssh-server-3.6.1p1-1.1asp
openssh-askpass-3.6.1p1-1.1asp
```

Если вы увидели что-то подобное у себя на экране значит пакет ssh-server у вас установлен и достаточно лишь проверить его конфигурацию. Если пакеты OpenSSH на вашем компьютере отсутствуют, то самое время их установить на сервер:

```
# rpm -ihv openssh-3.6.1p1-1.1asp
# rpm -ihv openssh-server-3.6.1p1-1.1asp
```

Сервер-демон sshd настраивается достаточно просто: редактированием файла /etc/ssh/sshd_config. Ниже приводится пример содержимого файла /etc/ssh/sshd_config:

```
#       $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.
#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::
# HostKey for protocol version 1
```

```
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 3600
#ServerKeyBits 768
# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO
# Authentication:

#LoginGraceTime 120
#PermitRootLogin yes
#StrictModes yes
#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys
# rhosts authentication should not be used
#RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
```

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#AFSTokenPassing no
# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no
# Set this to 'yes' to enable PAM keyboard-interactive authentication
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt no

#X11Forwarding no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#KeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression yes
#MaxStartups 10
# no default banner path
#Banner /some/path
#VerifyReverseMapping no
```

override default of no subsystems

Subsystem sftp /usr/libexec/openssh/sftp-server

В этом файле используются практически все значения по умолчанию, за исключением “X11Forwarding yes”, но это для запуска X-приложений. Как видите настраивать практически ничего не пришлось. Теперь проверяем работает ли демон sshd:

$ ps aux | grep sshd

root 712 0.0 0.0 3572 60 ? S Sep24 0:01 /usr/sbin/sshd

mikola 10794 0.0 0.0 3764 596 pts/1 S 09:30 0:00 grep sshd

Если работает, то все нормально. Если нет, запускаем его:

/sbin/service sshd start

Настройка сервера закончена. Теперь переходим к настройке клиента. Для начала проверяем установлена ли клиентская часть пакета OpenSSH: нас в частности интересуют пакеты openssh-clients и openssh. Если таких пакетов на импровизированном клиентском компьютере нет, самое время их установить:

rpm -ihv openssh-3.6.1p1-1.1asp

rpm -ihv openssh-clients-3.6.1p1-1.1asp

Надеюсь понятно, что в вашем любимом дистрибутиве операционной системы Linux расположение и версии этих пакетов могут значительно отличатся, но суть необходимости их установки от этого не измениться. Теперь вам доступна консольная команду удаленного подключения к серверу – ssh.

Синтаксис использования ssh очень простой, достаточно лишь знать доменное имя или IP-адрес сервера. Например, сервер к которому подключаюсь я имеет IP-адрес 192.168.0.20, поэтому я использую такую простую команду:

$ ssh 192.168.0.20

При первом подключении к серверу вам потребуется создать пару ключей и на экране вы увидите нечто подобное:

The authenticity of host '192.168.0.20 (192.168.0.20)' can't be established.

RSA key fingerprint is 92:00:65:d2:53:90:ec:42:69:0b:d0:97:51:24:5f:ed.

Are you sure you want to continue connecting (yes/no)?

Следует согласиться с предложением программы и набрать “yes”. После секундной паузы вы увидите на экране приглашение для ввода пароля пользователя удаленного сервера:

Warning: Permanently added '192.168.0.20' (RSA) to the list of known hosts.

pasha@192.168.0.20's password:

В данном случае необходимо ввести пароль пользователя pasha на сервере 192.168.0.20. Если вы все сделали правильно, то в терминале увидите приглашение оболочки удаленного сервера. Если и это вас не очень убедило (наверное я имею дело с настоящим скептиком), выполните команду проверки сетевых интерфейсов:

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:21:26:18:3B
          inet addr:192.168.0.20  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9052 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8408 errors:0 dropped:0 overruns:0 carrier:0
          collisions:738 txqueuelen:100
          RX bytes:6448118 (6.1 Mb)  TX bytes:1242932 (1.1 Mb)
          Interrupt:9 Base address:0xe400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:251 errors:0 dropped:0 overruns:0 frame:0
          TX packets:251 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:381023 (372.0 Kb)  TX bytes:381023 (372.0 Kb)
```

Как видите, IP-адрес сетевой карты (192.168.0.20) указывает на то, что вы действительно работаете на удаленном компьютере. Теперь вы можете запускать практически любое консольное приложение удаленного сервера,

например мой любимый файловый менеджер Midnight Commander (см. рис. 2.1).

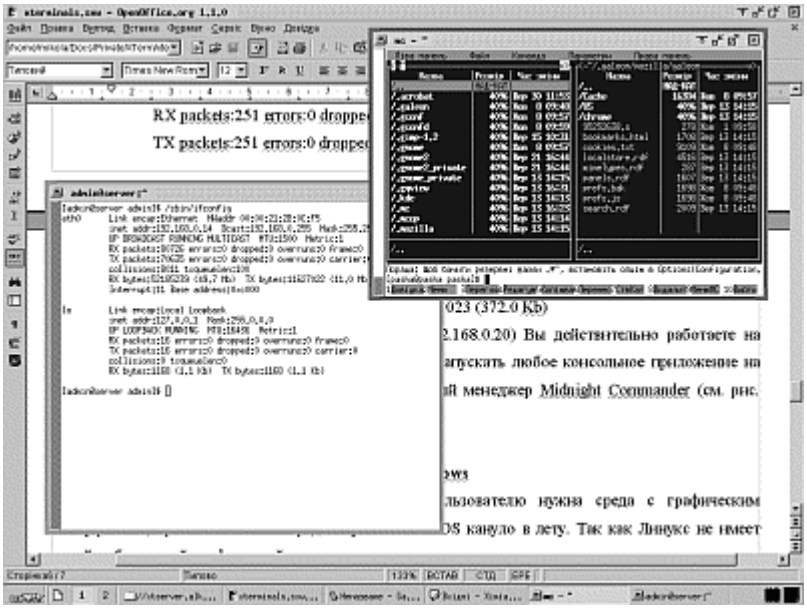

Рис. 2.1. Пример запуска Midnight Commander на удаленном компьютере

Возможности применения удаленного доступа довольно обширны. Так им постоянно пользуются системные администраторы для изменения параметров компьютеров пользователей. Не секрет, что сисадмины по своей природе народ ленивый и им совсем не хочется делать лишние телодвижения, поэтому им очень нравятся утилиты на подобие ssh. Также через ssh удобно администрировать удаленные серверы через Интернет. Иными словами удаленный доступ штука хорошая и нет никакой необходимости ограничивать его возможностями консольных приложений. Далее из этой книги вы узнаете как запускать на удаленном компьютере приложения с графическим интерфейсом системы X-Windows.

Следует помнить, что для ограничение доступа через протокол SSH в файле /etc/ssh/sshd_config могут использоваться параметры AllowGroups и AllowUsers, а также DenyGroups и DenyUsers. Из соображений системной безопасности целесообразно предоставлять доступ по SSH только тем пользователям, которым он действительно нужен, а не оставлять право удаленного входа для всех пользователей, как это сделано по умолчанию. Если при использовании команды ssh вы столкнулись с проблемами доступа по сети к удаленному серверу, то в первую очередь следует проверить наличие параметров ограничивающих доступ в файле /etc/ssh/sshd_config. Если это не

помогло, то нужно обратить внимание на правильность ввода пароля пользователя, а также посмотреть настройки межсетевого экрана (firewall), на предмет блокирования портов с номерами 22, 513-1023.

X-терминалы – бездисковые клиенты X-Windows

Для нормальной работы современному пользователю нужна среда с графическим интерфейсом, время консольных редакторов и MS DOS кануло в лету. Так как операционная система Linux не имеет своей собственной графической подсистемы, то приходится довольствоваться проектом системы X-Windows. И проект X-Windows не смотря на приличную историю развития (в информационных технологиях время иногда только вредит) имеет неоспоримые идеологические преимущества. Главным из которых является клиент-серверная архитектура. Зарождаясь во времена мейнфреймов, X-протокол распределял нагрузку между X-сервером и X-клиентом, что собственно и требуется сейчас для работы X-терминала.

Итак, для того чтобы понять преимущества архитектуры клиент-сервер системы X-Windows, давайте рассмотрим на примере, что происходит при запуске обычного графического приложения:

Пользователь отдает команду запустить программу;

Операционная система считывает з диска код программы и начинает выполнять его;

Программный код взаимодействуя с библиотеками системы X выводит на экран графическую информацию и выполняет другие полезные функции.

Так вот, персональный компьютер, который выполняет эту программу тратит все свои ресурсы как на полезный код (то что действительно выполняет программа без учета ее интерфейса) так и на графическую подсистему (рисование окошек на экране) и нельзя сказать, что графическая подсистема не влияет на скорость выполнения программы. В X-терминальном варианте все перечисленные операции выполняет сервер, а X-терминал только отображает графическую информацию. То есть, по большому счету, X-терминал выполняет одну программу – сервер X-Windows, в обязанности которой входит лишь обслуживание графической подсистемы. Такие ограниченные функциональные

обязанности значительно снижают требования к аппаратной базе для X-терминалов, а круг выполняемых на них задач зависит лишь от пропускной способности сети и мощности сервера.

В графических системах отличных от X-Windows такой трюк не получиться, так как их протоколы не рассчитаны на использование структуры клиент-сервер, а следовательно эффективность удаленного запуска графических приложений будет ниже (как вы уже знаете, удаленная работа в операционных системах не использующих X Window System более чем возможна, но соревноваться с X им очень тяжело).

Удаленный запуск GUI-приложений (XForwarding)

Только для того, чтобы вы поверили какая все-таки X-терминал крутая штука, я покажу вам как запускать X-приложения на удаленном компьютере. Когда лично я впервые увидел как это работает, то в очередной раз убедился что за операционной системой Linux будущее.

Для запуска нам понадобится все та же утилита ssh и возможно пакет openssh-askpass (для ввода пароля X-программ). Установим на клиентский компьютер пакет openssh-askpass, если он уже не установлен:

```
# rpm -ihv openssh-askpass-3.6.1p1-1.1asp
```

На сервере в файле /etc/ssh/sshd_config необходимо также добавить строчку “X11Forwarding yes” и перезапустить демон sshd:

```
# /sbin/service sshd restart
Shutting down sshd:                                [  OK  ]
Starting sshd:                                     [  OK  ]
```

Теперь с клиентского компьютера давайте запустим графическую программу, например, konqueror. Для этого достаточно набрать такую команду:

```
$ ssh -X -l mikola 192.168.0.20
```

В данном случае параметр -X означает принудительную поддержку клиентом X-протокола, так называемый "x-forwarding", а параметр -l mikola – пользователя удаленного компьютера (mikola). Если на удаленном компьютере имя пользователя совпадает с вашим текущим, то параметр -l использовать не

нужно. Вводим на запрос системы пароль и попадаем в удаленную систему. Теперь набираем команду запуска файлового менеджера:

$ konqueror

И на вашем рабочем столе появится окошко Konqueror (см. рис. 2.2.).

Рис. 2.2. Пример запуска программы с графическим интерфейсом (konqueror) на удаленном компьютере

Если оконный менеджер KDE не установлен на вашем импровизированном сервере, не беда, подойдет любая X-программа. Попробуйте запустить любое приложение с графическим интерфейсом и попробовать поработать с ним. Вам понравится скорость и удобство работы, а также та легкость доступа к информации на удаленном компьютере.

Глава 3. X-Windows

Описание X-протокола

Графическая система X-Windows для своих нужд использует X-протокол, который был разработан в середине 1980-го года для обеспечения прозрачного сетевого доступа к графическому интерфейсу пользователя и был ориентирован преимущественно на операционный системы семейства UNIX. X-протокол предоставляет управление графической информацией подобно другим альтернативным системам, наиболее известными из которых являются Microsoft Windows и Presentation Manager компании IBM.

Главное отличие X-протокола от упомянутых схожих разработок заключается в том, что он не ограничивает использование графического интерфейса только локальным компьютером, наоборот, X-протокол изначально разрабатывался для предоставления доступа удаленным сетевым клиентам в архитектуре клиент-сервер. Графический интерфейс пользователя согласно X-протоколу как бы разделяется на две части: одна часть уровня приложения, которую принято называть X-клиентом, и вторая часть уровня сервера – X-сервер. Исходя из такого подхода, приложения, которые используют в своей работе X-протокол, могут работать по такой схеме: X-клиент запускается на удаленной машине, а графическая информация передается локальному X-серверу, который, в свою очередь, и отображает ее пользователю.

Для обеспечения масштабируемости и переносимости системы X-Windows на различные аппаратные платформы X-протокол состоит из 2-х уровней: аппаратно-независимого и аппаратно-зависимого (см. рис. 3.1). Такая структурная схема позволяет вносить изменения только в одну часть программного комплекса (аппаратно-зависимый уровень) при переносе его на новую аппаратную основу.

Рис. 3.1. Два уровня X-протокола

X-протокол устанавливает клиент-серверные отношения между приложением и дисплеем (конечным интерфейсом пользователя). Такой подход позволяет приложению (X-клиенту) дистанционироваться от дисплея (X-сервера). X-протокол определяет также общий оконный пользовательский интерфейс в сочетании с асинхронной передачей информации через локальную сеть между X-клиентом и X-сервером. В результате конечный пользователь имеет практически полное отделение аппаратно-зависимого уровня и

особенностей реализации операционной системы от графического пользовательского интерфейса. Тем самым X-протокол скрывает аппаратный уровень от реализации X-клиента, что значительно облегчает его разработку и способствует легкому портированию X-клиентов. Схематически структуру работы X-протокола для конечного пользователя и разработчика можно представить, как изображено на рис. 3.2.

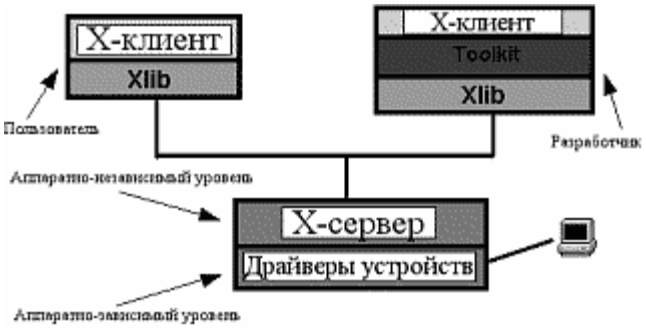

Рис. 3.2. Схема работы X-протокола между X-клиентом и X-сервером

Как видно на схеме, такой подход обеспечивает важные преимущества:

Удаленная работа через локальную сеть и работа на обычном персональном компьютере выглядит одинаково, как для конечного пользователя, так и для разработчика программного обеспечения;

X-сервер хорошо переносится на различные аппаратные платформы и операционные системы;

X-клиенты также имеют высокую степень переносимости (кросплатформенность);

X-протокол может работать совместно с любым потоковым транспортным протоколом: локальным или удаленным;

Производительность приложений не зависит от используемой графической подсистемы.

Конечно X-протокол далеко не идеален, и его использование сопряжено с целым рядом сложностей, которые и привели к появлению таких графических надстроек как Qt и Gtk. Сказывается и почтенный возраст X-Windows, так как зарождаясь на заре появления графических интерфейсов, она разумеется не могла досконально отобразить потребности современного пользователя компьютера.

Следует также сказать, что X Window System - один из самых больших и успешных проектов в истории компьютерной техники. Исторически сложилось, что два разработчика систем компьютерной графики – проект Athena (MIT) и W Windowing (Стэнфорд) - решили объединить свои усилия. В 1987 году официальным разработчиком системы стал X Consortium, который позже был переименован в X Open Group (официальный сайт проекта http://www.x.org).

Архитектура X-Windows

Фундаментальность построения X-Windows заключается в том, что изначально эта система была рассчитана на удаленное взаимодействие своих компонентов. Только представьте, какой это огромный потенциал, учитывая, что проект развивается уже более 15 лет. Опираясь на разные сетевые протоколы транспортного уровня, X-Windows может легко туннелировать соединения X-сервера и X-клиента, а также возможно даже туннелирование взаимодействий оборудования с X-сервером. Это означает, что кроме привычной удаленной работы X-сервера и X-клиента, возможно удаленное взаимодействие отдельных аппаратных компонентов данной системы. Проще говоря, монитор может находиться в одной комнате, клавиатура – в другой, а “мышка” - в другом городе (как говорится бессмысленно, но зато практически возможно).

На словах это звучит красиво, а на практике конечно же все гораздо более скромно. В использовании X-Windows чувствуется архаизм, который сопровождает любой проект со столь почтенной историей. Да это и понятно, так как нужно соблюдать совместимость с программными продуктами, написанными для предыдущих версий X. Значительно помогают преодолеть этот недостаток современные оконные менеджеры. “Голый” X-сервер, к счастью или к сожалению, не имеет функций для работы с пользователем. В его состав не входят ни рабочий стол, ни панель задач, ни главное меню. Все эти функции реализованы в оконном менеджере (например, KDE, Gnome, IceWM, XFce и т.д.), который в свою очередь взаимодействует с менеджером дисплея (например, xdm, gdm или kdm). И менеджер окон и менеджер дисплея являются X-клиентами, которые обеспечивают необходимую для пользователя

функциональность компьютера. Другими словами, современная система X-Windows представляет из себя многослойную структуру с большим количеством программ-посредников (см. рис. 3.3).

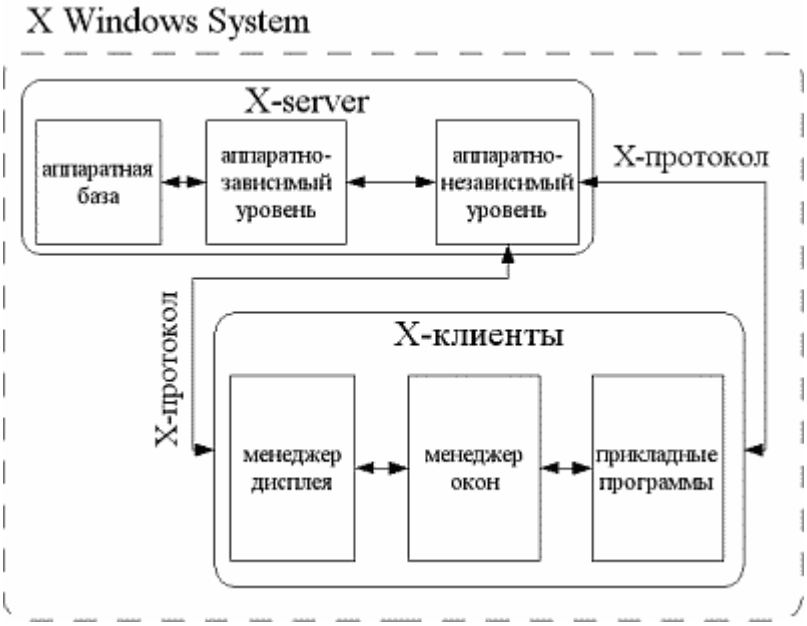

Рис. 3.3. Структура X-Windows

Другим важным отличием системы X-Windows является то, что она не есть частью операционной системы, а фактически запускается как приложение. Эта особенность позволяет использовать ее вместе со встроенными в операционную систему графическими интерфейсами. К тому же, такой подход изолирует интерфейс пользователя от ядра операционной системы и тем самым повышает его надежность.

X-сервер и X-клиент

Как уже было сказано ранее, в X Windows System информационный обмен осуществляется между X-клиентом и X-сервером. Давайте рассмотрим более детально эти две составляющие. Программное обеспечение, которое управляет монитором, “мышкой” и клавиатурой, принято называть X-сервером. Приложения, результат работы которых отображается X-сервером, называются X-клиентами. X-клиент отсылает запросы X-серверу, а X-сервер, который может одновременно работать с несколькими X-клиентами, отвечает X-клиенту нужной информацией, например, пользовательским вводом или возникшими ошибками.

X-сервер

Особенностью X-сервера является то, что он всегда запускается локально, то есть на том компьютере (терминале), на котором работает пользователь. В функции X-сервера входит:

Отображение на экране визуальной информации;

Ответы на информационные запросы пользователя;

Реагирование на ошибки при выполнении запросов;

Управление клавиатурой, “мышкой” и монитором как локально, так и по сети;

Создание, управление и удаление окон (отображение в окнах графической информации).

X-сервер может отображать окна различных X-клиентов, в том числе и запущенных на разных компьютерах.

X-клиент

X-клиент,как правило, работает в тесной взаимосвязи с графической библиотекой (например, Xlib) для обработки запросов X-протокола. Наиболее распространенными его функциями являются:

отправка сообщений X-серверу (например, запрос на рисование линии на экране);

обработка событий, поступающих от X-сервера (например, события от “мыши” и клавиатуры);

получение ошибок от X-сервера (например, ошибка доступа к удаленному серверу).

И X-клиент и X-сервер взаимодействуют посредством X-протокола, которому свойственны такие составляющие компоненты, как: запросы, ответы, события и сообщения об ошибках.

Запросы

X-клиент направляет запросы X-серверу, на выполнение определенных операций (например, запрос создания окна);

Для улучшения скорости обработки запросов, X-клиент обычно не ждет подтверждения получения его запроса X-сервером, полностью полагаясь на сетевой транспортный протокол;

X-запросы всегда кратны 4-м байтам.

Ответы

X-сервер отличает запросы, которые требуют ответа, и отвечает на них. Как было уже сказано выше, из соображений производительности далеко не все запросы требуют ответов;

X-ответ также, как и X-запрос, кратен 4-м байтам, но всегда не меньше 32-х байт.

События

X-сервер пересылает X-события клиентам, которые их ожидают. Это может быть ввод с клавиатуры или управление “мышкой”. Для минимизации сетевого трафика, X-клиентам отсылаются только ожидаемые события;

Длина X-событий всегда 32 байта.

Сообщения об ошибках

X-сервер может отвечать сообщениями об ошибках на запросы X-клиентов. Сообщения об ошибках очень схожи с событиями, но по другому обрабатываются;

Длина сообщения об ошибке 32 байта, и отправляется она специальной подпрограмме X-клиента, которая предназначена для обработки X-ошибок.

Базируясь на описанных выше понятиях, становиться понятно, что X-клиент – это программный продукт, работающий с графической информацией на более высоком абстрактном уровне. В отличие от него, X-сервер, а точнее его конкретная программная реализация, очень сильно зависит от того оборудования и операционной системы, для которых он спроектирован. Если анализировать особенности аппаратно-зависимого уровня X-сервера, то ему свойственны:

адаптация X-сервера к выбранной платформе;

учет порядка следования байтов и модификация его между компьютерами с разным порядком (MSB and LSB). Порядок следования байт учитывается при выполнении каждого X-запроса;

маскирование отличий аппаратного обеспечения и особенностей реализации операционных систем от X-клиентов;

работа с драйверами устройств, такими как клавиатура, “мышь” и монитор.

В зависимости от платформы X-сервер может быть приложением однопотоковым, то есть выполняться в линейном режиме, или же быть полностью многозадачной системой и использовать преимущества многопотоковости центрального процессора.

Сегодня X-сервера работают на трех основных типах компьютеров:

Персональные компьютеры– обладают достаточной вычислительной мощью для локального запуска X-сервера и X-клиентов. Также незначительный сетевой трафик может приходиться на работу с удаленными X-клиентами;

X-терминалы– выполняют по сути только одну программу X-сервер, которая загружается по сети с сервера, поэтому они славятся скромными системными требованиями. Как правило, не содержат жесткого диска. X-терминалы значительно дешевле персональных компьютеров и проще в обслуживании;

Гибридные ПК– совмещают в себе преимущества традиционных персональных компьютеров и широкие возможности работы с удаленным рабочим столом. В зависимости от задач пользователя могут работать как в локальном так и в сетевом режимах. Просты в использовании, но значительно сложнее в администрировании.

Настройка X-сервера и XDM/XDMCP

В своей работе я использую общедоступную версию X-сервера для платформы Intel, которая называется XFree86 (официальный сайт проекта http://www.xfree86.org), поэтому о ней и пойдет речь далее в этой книге. Если читатель до этого был знаком только с графической подсистемой операционных систем от Microsoft, то настройка X-сервера в XFree86 покажется ему неоправданно сложным занятием. Честно признаться, но поначалу я тоже был немало смущен кажущейся сложностью этого процесса. Сразу хочу обратить ваше внимание на главные отличия, которые сопутствуют настройки X-сервера:

Отсутствие единой программы конфигурации, а также несовершенство имеющихся утилит автоконфигурирования;

Все параметры работы X-сервера хранятся в одном текстовом файле;

Так как X-сервер не является частью ядра операционной системы, то он запускается как обычная программа (можно настроить запуск в скрипте инициализации);

Модульная, распределенная структура X-сервера, подразумевает использование также сопутствующих клиентских приложений: менеджера дисплея и оконного менеджера;

Не часто можно встретить драйвер устройства для X-сервера, который раскрывает весь потенциал его аппаратных возможностей. Но зато практически все видеокарты можно запустить в VESA-совместимом режиме;

При использовании специальных настроечных утилит от вас как от пользователя могут потребоваться дополнительные знания устройства компьютера, в частности параметров видеокарты, монитора, клавиатуры и манипулятора “мышь”.

Практически все дистрибутивы Linux при установке операционной системы запускают утилиту настройки X-сервера. Если вы не используете ультра-модную видеокарту или еще какой-нибудь эксклюзивный раритет, то в 90% случаев система сама успешно настроит X-сервер и при первом же старте компьютер перейдет в графический режим. Но не всегда эта процедура проходит гладко, и нужно быть готовым к выполнению ручной настройки конфигурационного файла X-сервера.

В версиях 4.x XFree86, главный конфигурационный файл X-сервера находится в каталоге /etc/X11/ и носит название XF86Config (иногда XF86Config-4). Начиная с 4-й версии, конфигурационные файлы проекта XFree86 претерпели значительных изменений, что не должно не радовать пользователей, так как теперь их структура стала значительно проще. Примерное содержание этого файла приведено ниже (данный файл взят с рабочего компьютера под управлением операционной системы ASPLinux 9.0, версия XFree86 4.3.0.):

```
# buildxconf *** delete this line to prevent buildxconf from modifying this file
#
```

```
# XF86Config auto-generated by buildxconf
#
# Copyright (c) 1999, Corel Corporation
# Robin Burgener, robinb@corel.com
#
# See 'man XF86Config' for info on the format of this file

Section "ServerLayout"
	Identifier     "XFree86 Configured"
	Screen      0  "Screen0" 0 0
	InputDevice    "Mouse0" "CorePointer"
	InputDevice    "Keyboard0" "CoreKeyboard"
	InputDevice    "DevInputMice" "AlwaysCore"
EndSection

Section "Module"
	Load  "GLcore"
	Load  "dbe"
	Load  "dri"
	Load  "extmod"
	Load  "fbdevhw"
	Load  "glx"
#	 Load  "pex5"
	Load  "record"
#	 Load  "xie"
#	 Load  "v4l"
	Load  "type1"
	Load  "freetype"
EndSection

Section "InputDevice"
```

```
	Identifier "Keyboard0"
	Driver  "keyboard"
	Option  "Protocol"      "Standard"
	Option  "XkbModel"      "pc105"
	Option  "XkbRules"      "xfree86"
	Option  "XkbLayout"     "us,ru,ua"
	Option  "XkbVariant"    ",winkeys,winkeys"
	Option  "XkbOptions"    "grp:ctrl_shift_toggle,grp_led:scroll"
EndSection

Section "InputDevice"
# If the normal CorePointer mouse is not a USB mouse then
# this input device can be used in AlwaysCore mode to let you
# also use USB mice at the same time.
	Identifier      "DevInputMice"
	Driver  "mouse"
	Option  "Protocol" "IMPS/2"
	Option  "Device" "/dev/input/mice"
	Option  "ZAxisMapping" "4 5"
	Option  "Emulate3Buttons" "no"
EndSection

Section "InputDevice"
	Identifier      "Mouse0"
	Driver  "mouse"
	Option  "Protocol"      "PS/2"
	Option  "Device"        "/dev/mouse"
# When using XQUEUE, comment out the above two lines, and uncomment
# the following line.

#Option "Protocol"      "Xqueue"
```

```
# Baudrate and SampleRate are only for some Logitech mice

#    BaudRate  9600
#    SampleRate  150

        Option  "ZAxisMapping"  "4 5"

# Emulate3Buttons is an option for 2-button Microsoft mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)
        Option  "Emulate3Buttons"       "off"

# ChordMiddle is an option for some 3-button Logitech mice

#    ChordMiddle
EndSection

Section "Files"
        RgbPath "/usr/X11R6/lib/X11/rgb"
        FontPath        "unix/:7100"
EndSection

Section "ServerFlags"
    AllowMouseOpenFail
EndSection

Section "Monitor"
        Identifier      "Monitor"
        VendorName      "HEI"
        ModelName       "Unknown"
        HorizSync       37-68
```

```
        VertRefresh      60-85
    #800x600 @ 85.0 Hz, 55.8 kHz hsync
        Modeline        "800x600"       60.75  800  864  928 1088   600  616  621
657 -HSync -VSync
    #640x480 @ 85.0 Hz, 43.3 kHz hsync
        Modeline       "640x480"      36    640  696  752  832   480  481  484  509 -
HSync -VSync
    EndSection

    Section "Monitor"
        #!Description   "Super VGA 1024x768 @ 60Hz"
        Identifier      "Super-Duper VGA"
        HorizSync       31.5-48.5
        VertRefresh     50-90
    # 1024x768 @ 60 Hz, 48.4 kHz hsync
        Modeline        "1024x768"      65    1024 1032 1176 1344   768  771  777
806 -hsync -vsync
    # 800x600 @ 60 Hz, 37.8 kHz hsync
        Modeline       "800x600"      40    800  840  968 1056   600  601  605  628
+hsync +vsync
    # 640x400 @ 70 Hz, 31.5 kHz hsync
        Modeline        "640x480"       25.175 640  664  760  800   480  491  493
525
    EndSection

    Section "Device"
        BusID   "PCI:1:0:0"
        Driver  "radeon"
        Identifier      "ATI|Radeon R200 QL [Radeon 8500 LE]"
        BoardName       "ATI Radeon 8500LE"
    EndSection
```

```
Section "Screen"
    Identifier     "Screen0"
    Device  "ATI|Radeon R200 QL [Radeon 8500 LE]"
    Monitor "Monitor"
    DefaultColorDepth      16
    SubSection "Display"
        Depth   4
        Modes   "800x600" "640x480"
        Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth   8
        Modes   "800x600" "640x480"
        Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth   15
        Modes   "800x600" "640x480"
        Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth   16
        Modes   "800x600" "640x480"
        Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth   24
        Modes   "800x600" "640x480"
        Virtual 800 600
    EndSubSection
```

```
        SubSection "Display"
            Depth   32
            Modes   "800x600" "640x480"
            Virtual 800 600
        EndSubSection
    EndSection

    Section "DRI"
        Mode 0666
    EndSection
```

Как видно из файла примера, структура его довольно проста и разбита на секции. Первой в файле /etc/X11/XF86Config расположена секция “ServerLayout”, в которой определяются используемые устройства ввода, а также параметры экрана X-сервера. В разделе “ServerLayout” приведены только названия соответствующих разделов, в которых заключена более подробная информация. Следует отметить, что в системе X Windows должны существовать два ключевых устройства ввода: клавиатура и манипулятор “мышь”. В разделе “ServerLayout” они обозначаются как "CoreKeyboard" и "CorePointer". Для приведенного выше примера файла XF86Config, X-сервер будет считать первичными клавиатуру с идентификатором “Keyboard0” и “мышку” с идентификатором “Mouse0”.

Настройка X-сервера и XDM/XDMCP (продолжение)

Раздел “Module” содержит список загружаемых модулей X-сервера. Как правило, обязательным является только модуль "extmod". Динамически загружаемые модули используются для расширения функциональности X-сервера. Например, модули, предназначенные для работы со шрифтами “type1” и “freetype”, позволяют X-серверу использовать в своей работе шрифты формата PostScript Type 1 и TrueType. Каждый загружаемый модуль выполняет определенную функцию, а разобраться с особенностями применения конкретных модулей и передачей им параметров (подраздел SubSection “имя модуля”) поможет сопровождающая сервер XFree86 документация.

Раздел “InputDevice” описывает устройства ввода, как правило, клавиатуру и манипулятор “мышь”. Количество таких разделов в файле XF86Config не ограничено, и каждый из них должен иметь уникальное значения параметра “Identifier”. В упомянутом выше разделе “ServerLayout” используется ссылка именно на это значение, так как параметр “Identifier” идентифицирует весь раздел. Если устройство ввода должно быть первичным для X-сервера, то в параметрах можно использовать опции:

Option "CorePointer" – для “мышки”

Option "CoreKeyboard" – для клавиатуры

Рассмотрим более детально методы и параметры настройки в XFree клавиатуры и “мышки”. Итак, для того, чтобы заставить вашу клавиатуру правильно работать с X-сервером, в разделе “InputDevice” следует, в первую очередь выбрать соответствующий драйвер, как правило, это:

Driver "keyboard"

Символьное устройство подключения клавиатуры указывать не нужно (PS/2, USB и т.д.), так как в компьютере работать может только одна клавиатура, выбор и инициализация которой осуществляется еще на этапе загрузки ядра системы. К важным параметрам настройки работы клавиатуры относятся такие:

Option "XkbLayout" "us,ru,ua"

Option "XkbVariant" ",winkeys,winkeys"

Option "XkbOptions" "grp:ctrl_shift_toggle,grp_led:scroll"

XkbLayout – отвечает за раскладку клавиатуры, и начиная с версии XFree 4.3, поддерживает неограниченное количество одновременно использующихся раскладок, которые достаточно перечислить через запятую. До выхода в свет версии 4.3 можно было использовать только две раскладки клавиатуры, что частично мешало использовать систему X-Windows многоязычным пользователям.

Параметр XkbVariant позволяет задать дополнительные опции для каждой из раскладок клавиатуры. Так в приведенном выше примере для русского (ru) и украинского (ua) языков используется расположение клавиш как это принято в

операционных системах Windows, а для английского языка (us – английский США) применяются правила поведения по умолчанию.

XkbOptions показывает, какая комбинация клавиш используется для смены расскладок клавиатуры (в примере Ctrl+Shift). Дополнительный параметр “grp_led:scroll” применяется для того, чтобы задействовать индикатор “Scroll Lock” на клавиатуре при переключении альтернативных раскладок.

Настройка манипулятора “мышь” более замысловата, так как устройства точечного ввода информации отличаются большим конструктивным разнообразием. В разделе “InputDevice” после обязательного параметра “Identifier” необходимо указать драйвер манипулятора “мышь”, например:

Driver "mouse"

Далее, в зависимости от типа “мышки”, следует указать символьное устройство подключения ее к компьютеру. Для “мышей” с интерфейсом PS/2 это:

Option “Device” “/dev/psaux”

Для USB “мышек”:

Option “Device” “/dev/input/mice”

Для “мышей” подключенных к последовательному порту COM1:

Option “Device” “/dev/ttyS0”

Для “мышей” подключенных к последовательному порту COM2:

Option “Device” “/dev/ttyS1”

Теперь в зависимости от устройства подключения и наличия колесика-скроллера, следует указать используемый протокол. Несмотря на то, что протоколов довольно много, в подавляющем количестве случаев вас вполне удовлетворит один из следующих вариантов.

Для обычной PS/2 или USB “мышки” без скроллера используем такую запись:

Option “Protocol” “PS/2”

Для “мышки” PS/2 или USB со скроллером:

Option “Protocol” “IMPS/2”

Для COM “мышки” без скроллера:

Option “Protocol” “Microsoft”

Для COM “мышки” со скроллером:

Option “Protocol” “IntelliMouse”

Пользователи 2-х кнопочных “мышей” могут воспользоваться эмуляцией нажатия на третью кнопку “мыши”, путем применения параметра "Emulate3Buttons". Если в конфигурационном файле X-сервера в разделе “InputDevice”, который посвящен настройке “мыши” встретится запись:

Option "Emulate3Buttons" "on"

То она означает, что одновременное нажатие левой и правой кнопки “мыши” эквивалентно нажатию на среднюю кнопку.

Для того, чтобы изменить скорость движения указателя “мыши” по экрану, используется параметр "Resolution", например:

Option "Resolution" "400"

Увеличивая его числовое значение, вы тем самым вынуждаете “мышку” двигаться быстрее, и наоборот.

Остальные параметры не так существенны, а поэтому применяются редко. Как видите, в настройках “мышки” и клавиатуры в XFree86 нет ничего сложного и сверхъестественного.

Настройка X-сервера и XDM/XDMCP (продолжение)

Раздел “Files” отвечает за определение пути к каталогам, в которых хранятся используемые X-сервером файлы. Основными параметрами данного раздела являются:

FontPath – определение способа использования шрифтов (может использоваться адрес сервера шрифтов либо путь к каталогу);

RGBPath – путь к базе данных цветового преобразования;

ModulePath – указывает каталог размещения загружаемых динамических модулей (см. также раздел “Module”).

Раздел “ServerFlags” задает глобальные параметры поведения X-сервера. В приведенном выше примере файла /etc/X11/XF86Config используется только одно специальное значение: AllowMouseOpenFail, которое означает

возможность запуска X-сервера даже при ошибке инициализации “мышки” (без “мышки”). Кроме этого параметра вас могут заинтересовать следующие:

Option "DontVTSwitch" – запрещает переключение виртуальных консолей нажатием комбинации клавиш Ctrl+Alt+Fn, где Fn – функциональная клавиша F1, F2, F3 и т.д.;

Option "DontZap" – запрещает использование комбинации клавиш Ctrl+Alt+BackSpace для принудительного завершения работы X-сервера. Теоретически это повышает безопасность использования X-сервера, зато пользователь теряет возможность самостоятельно завершить работу “повисшего” X-сервера;

Option "DontZoom" – данная опция отменяет действие комбинаций клавиш Ctrl+Alt+<Серый плюс> и Ctrl+Alt+<Серый минус>, которые предназначены для переключения видеорежимов, которые перечислены в разделе “Screen”;

Option "XkbDisable" – запрещает использование расширений клавиатуры;

Option "BlankTime" "время в минутах" – определяет через сколько минут бездействия пользователя следует запустить хранитель экрана. Значение по умолчанию 10 минут;

Option "StandbyTime" "время в минутах" – указывает через сколько времени пользовательского бездействия монитор и видеокарта совместимые с режимом VESA DPMS могут переходить в ждущий режим (Standby). Для того, чтобы эта функция использовалась необходимо наличие параметра "DPMS" в разделе "Monitor". Значение, используемое X-сервером, по умолчанию - 20 минут;

Option "SuspendTime" "время в минутах" – данный параметр также будет работать только для монитора и видеокарты, совместимыми с режимом VESA DPMS (параметр "DPMS" в разделе "Monitor"). Он указывает через какое время отсутствия пользовательской активности перевести монитор в режим Suspend. Значение по умолчанию 30 минут;

Option "OffTime" "время в минутах" – соответствует времени выключения монитора в минутах при отсутствии активных действий пользователя (работа с “мышкой” и клавиатурой). Значение по умолчанию 40 минут, но данная

функция также работает только для монитора и видеокарты, которые совместимы с режимом VESA DPMS;

Option "Pixmap" “пиксельный формат” – определяет пиксельный формат для глубины цвета в 24 бита. Возможные значения 24 и 32. Значение по умолчанию 32, но некоторые видеодрайверы не совсем корректно работают с таким форматом;

Option "NoPM" – запрещает обработку событий от подсистемы электросбережения. По умолчанию будут использоваться функции управления электропитанием (power management) для систем, в которых реализована их поддержка.

В разделе “Monitor” приводится описание используемого монитора. Это довольно специфический раздел, так как для его ручного создания требуется довольно неплохое знание технических характеристик монитора. Начинается он с указания обязательного параметра Identifier, значение которого будет потом использовано в разделе "Screen". Парамерты VendorName и ModelName, отвечающие за название производителя монитора и его модель, носят скорее декларативный, чем информационных характер, а поэтому указывать их рекомендуют, но не настаивают на этом.

Параметры HorizSync и VertRefresh отвечают за частоты горизонтальной синхронизации и вертикальной разверстки монитора. Значения этих параметров задаются в виде числового диапазона, причем HorizSync по умолчанию принимает значения в килогерцах, а VertRefresh в герцах. Настоятельно рекомендуется использовать те значения, которые записаны в техническом паспорте монитора, или получены из компетентных источников (например, с официального сайта производителя).

Для того чтобы правильно рассчитать разрешающую способность монитора по вертикали и горизонтали, используется параметр DisplaySize, например:

DisplaySize 310 230

Приведенная выше опция соответствует монитору с габаритными размерами видимой области экрана 310 x 230 мм, а при известной

разрешающей способности экрана, например 1024 x 768 точек, это дает основание точно определить значение разрешающей способности монитора DPI (количество точек на дюйм).

Для выполнения цветовой коррекции средствами X-сервера существует параметр Gamma. Он может принимать значения от 10.0 до 0.1, причем по умолчанию равен 1.0. Цветовая коррекция может применяться как ко всем RGB компонентам, так и к каждой в отдельности. Например, чтобы увеличить в два раза значения яркости цветовой гаммы для всего спектра, достаточно ввести в конфигурационный файл такой параметр:

Gamma 2.0

Если же требуется выполнить тонкую настройку каждого цвета в отдельности, то следует воспользоваться таким вариантом:

Gamma 1.1 0.9 1.3

В приведенном выше примере коррекция красного канала равна 1.1, зеленого 0.9, а синего 1.3 соответственно. Следует также помнить, что не все видеодрайверы XFree86 используют цветовую коррекцию.

Для описания технических параметров работы монитора при определенном разрешении экрана используются два аналогичных параметра Mode (длинная форма) и ModeLine (короткая форма записи). Описание особенности применения этих параметров выходит за рамки данной книги, так как это довольно узкая и технически сложная тема. Использование опций Mode и ModeLine позволяет оптимально настроить монитор и даже работать в нестандартных разрешениях, но применение неправильных данных может также послужить причиной выхода его из строя. Поэтому, если вы точно не уверены, что вносите изменения, которые соответствуют возможностям вашего монитора, лучше воздержитесь от подобных настроек. Более подробную информацию о возможностях настройки монитора ищите в документации к проекту XFree86.

Очень полезным дополнением файла XF86Config может стать такая строчка в разделе “Monitor”, если монитор соответствует спецификации VESA DPMS:

Options “DPMS”

Данный параметр позволяет задействовать энергосберегающие функции монитора и не только сэкономить ваши деньги на оплате электроэнергии, но и как следствие, значительно продлить жизнь монитору.

Настройка X-сервера и XDM/XDMCP (продолжение)

Раздел "Device" отвечает за настройку работы видеоадаптера в X-Windows. После обязательного параметра Identifier следует не менее обязательный параметр Driver, который определяет соответствующий видеокарте драйвер. Список доступных вашей версии XFree86 драйверов можно просмотреть выполнив команду:

```
$ ls /usr/X11R6/lib/modules/drivers
apm_drv.o       dummy_drv.o    nsc_drv.o          sis_drv.o
ark_drv.o       fbdev_drv.o    nv_drv.o           tdfx_drv.o
ati_drv.o       glint_drv.o    r128_drv.o         tga_drv.o
atimisc_drv.o   i128_drv.o     radeon_drv.o       trident_drv.o
chips_drv.o     i740_drv.o     rendition_drv.o    tseng_drv.o
cirrus_alpine.o i810_drv.o     s3_drv.o           vesa_drv.o
cirrus_drv.o    linux          s3virge_drv.o      vga_drv.o
cirrus_laguna.o mga_drv.o      savage_drv.o
cyrix_drv.o     neomagic_drv.o siliconmotion_drv.o
```

Название видеодрайвера следует писать без суффикса “_drv.o”, например:

Driver "radeon"

Универсальные драйверы vga и vesa, предназначены для работы соответственно в режимах VGA и VESA-совместимых. Их следует использовать лишь в случае крайней необходимости, если не получится запустить X-сервер с родным драйвером видеоадаптера.

Параметр BusID отвечает за расположение видеокарты в определенном слоте расширения персонального компьютера. Для видеоадаптеров с интерфейсом PCI/AGP имеет формат PCI:номер шины:номер устройства:функция. Например, для AGP-карт он записывается в такой форме:

BusID "PCI:1:0:0"

В случае использования нескольких видеоадаптеров или видеоадаптеров с функцией multi-head применение параметра BusID обязательно. Также данный параметр может пригодиться при настройке систем со встроенной видео-картой. Часто в моей практике мне доводилось сталкиваться с компьютерами, которые имели интегрированное видео, которое не поддерживалось проектом XFree86 версии 4.x. Устанавливая внешнюю видео-карту необходимо правильно прописать параметр BusID для использования ее совместно с X-сервером.

Параметр Screen отвечает за номер монитора для данной конфигурации видеоадаптера, и применяется лишь в том случае, когда ваша видео-карта может одновременно выводить разное изображение на два и больше мониторов и вы используете эту возможность. По умолчанию считается, что видео-карта работает с монитором, которому присвоен условный нулевой номер. При настройки системы X-Windows на работу с несколькими мониторами, для каждого их них следует создавать отдельный раздел "Device" с уникальным номером Screen.

Параметр VideoRam позволяет задать объем доступной видеопамяти в килобайтах. Применять данный параметр следует лишь в крайних случаях, так как большинство видео-карт автоматически, при помощи своего драйвера, определяют это значение.

ChipID и ChipRev позволяют пользователю принудительно установить номер и версию видеочипа, который установлен на видео-карте, и тем самым перекрыть значения, полученные автоопределением. Необходимость в применении этих параметров иногда возникает при работе с новыми устройствами, для которых еще не написаны соответствующие драйвера, а используя номер и версию видеочипа предыдущего поколения, иногда получается заставить их работать в X-Windows, конечно только в том случае, если для видеоадаптеров справедливо утверджение об обратной совместимости.

Для задания особых параметров работы видеокарты можно использовать опцию Option, которая может принимать как стандартные значения для любого

драйвера X-сервера, так и специфические, узнать о которых можно из документации, сопровождающей видеодрайвер. Список общедоступных значений параметра Option в XFree86 версии 4.x достаточно большой, но в реальной работе пользователя заинтересуют лишь некоторые:

Option "NoAccel" – запрещает использование XAA (X Acceleration Architecture), то есть акселерацию выполнения 2-х мерных операций X-сервером. Применять данный параметр рекомендуется лишь в крайнем случае некорректного поведения видеодрайвера, например, при появлении на экране артефактов (искажения изображения на мониторе). Данная команда эквивалентна запрещению использования таких составляющих архитектуры XAA, как:

XaaNoCPUToScreenColorExpandFill

XaaNoColor8x8PatternFillRect

XaaNoColor8x8PatternFillTrap

XaaNoDashedBresenhamLine

XaaNoDashedTwoPointLine

XaaNoImageWriteRect

XaaNoMono8x8PatternFillRect

XaaNoMono8x8PatternFillTrap

XaaNoOffscreenPixmaps

XaaNoPixmapCache

XaaNoScanlineCPUToScreenColorExpandFill

XaaNoScanlineImageWriteRect

XaaNoScreenToScreenColorExpandFill

XaaNoScreenToScreenCopy

XaaNoSolidBresenhamLine

XaaNoSolidFillRect

XaaNoSolidFillTrap

XaaNoSolidHorVertLine

XaaNoSolidTwoPointLine

Option "NoMTRR" – запрещает использование технологии MTRR (Memory Type Range Register), которая значительно увеличивает скорость отображения видеоинформации. Причина использования этой опции – некорректное поведение некоторого аппаратного обеспечения и ошибочная реализация поддержки MTRR определенными видеодрайверами системы XFree.

Раздел "Screen" предназначен для консолидации информации в разделах “Device” и “Monitor”. Для этого за уже обязательным параметром Identifier следует использовать параметры Device и Monitor, со ссылками на соответствующие разделы файла конфигурации.

Параметр DefaultDepth (синоним DefaultColorDepth) указывает на глубину цвета (в битах) по умолчанию, и может принимать значение 1, 4, 8, 15, 16, 24 и 32 (на практике, редко используются значения ниже 16). Если опустить данный параметр, то X-сервером будет задействовано значение по умолчанию, которое присуще драйверу видеоадаптера.

Параметр DefaultFbBpp указывает на глубину цвета для фрейм-буфера.

Каждый раздел "Screen" должен содержать один или более подразделов "Display", которые отвечают за параметры работы монитора и видеокарты при различной глубине цвета. Данный подраздел может использовать такие парамеры:

Depth – определяет глубину цвета в битах для подраздела;

FbBpp – соответствует формату хранения изображения во фрейм-буфере;

Weight – применяется только для драйверов, которые поддерживают различные наборы распределения количества бит на RGB цвет. Характерен только для режима 16 бит;

Virtual – задает действительное разрешение экрана, которое будет использоваться. Например, если используется такой параметр:

Virtual 800 600

А пользователь попытался перевести монитор в режим 1024x768, то реальным останется режим 800x600, при этом на экране монитора будет отображаться рабочий стол значительно большего размера (1024x768),

частично с прокруткой. Лучше проиллюстрировать данную ситуацию поможет рисунок 3.4:

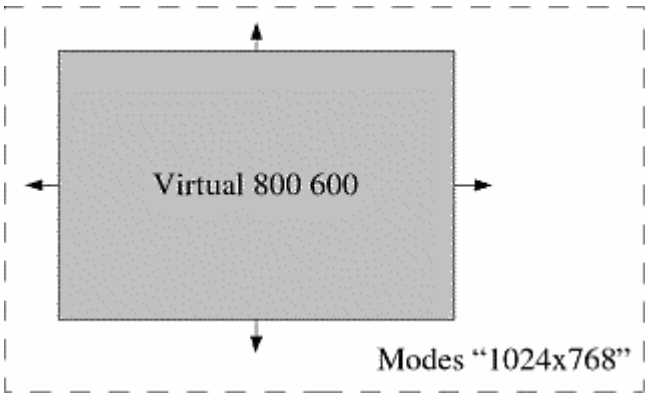

Рис. 3.4. Применение параметра Virtual

ViewPort – помогает задать координаты верхнего левого угла экрана, когда реальное разрешение экрана не совпадает с виртуальным. Если данный параметр опущено, то изображение рабочего стола будет отцентрировано в виртуальном экране.

Modes – указывает все допустимые режимы для подраздела "Display". Если значений больше одного, то они перечисляются через пробел, например:

Modes "1024x768" "800x600" "640x480"

Имена режимов работы монитора могут быть как встроенными VESA, так и нестандартными из раздела “Monitor”. При инициализации X-сервера будет выбран первый правильный режим параметра Modes из подраздела "Display", который отвечает глубине цвета установленной по умолчанию (параметр DefaultDepth).

Другие параметра раздела “Screen” и подраздела "Display" применяют редко, поэтому не будем на них останавливаться.

Последним разделом в нашем примере файла конфигурации X-сервера идет “DRI”. Его параметры определяют особенности использования Direct Rendering Infrastructure, которая отвечает за 3-х мерные функции X Window System, а следовательно интересна только очень узкому кругу специалистов. Дополнительную информацию по поводу применения технологии DRI можно найти на сайте http://www.xfree86.org/current/DRI.html.

Надеюсь, что изложенных выше знаний вам хватит для внесения изменений в файл /etc/X11/XF86Config, но прежде чем приступить к его правке, я настоятельно рекомендую сделать резервную копию этого файла:

```
# cd /etc/X11/
# cp XF86Config XF86Config.BACKUP
```

За запуск графического X-сервера сразу при старте операционной системы Linux отвечает файл /etc/inittab, а в частности такая его строка:

```
$ grep initd /etc/inittab
#   0 - halt (Do NOT set initdefault to this)
#   6 - reboot (Do NOT set initdefault to this)
id:5:initdefault:
```

В дистрибутивах операционной системы Linux основанных на RedHat графическим уровнем выполнения является пятый. Именно на нем происходит запуск X-сервера в сценарии /etc/X11/prefdm. Если вы столкнулись с ситуацией, когда из текстового режима вам вручную нужно запустить X-сервер, то вы всегда можете воспользоваться сценарием startx.

Менеджер дисплея и XDMCP

Сам по себе X-сервер мало чем может помочь конечному пользователю. Поэтому для выполнения функций аутентификации, запуска графических сессий и управления экранами в рамках проекта X Window System был разработан менеджер дисплея – XMD (X Display Manager – менеджер X дисплея). Данный программный продукт поддерживает также работу с протоколом XDMCP (X Display Manager Control Protocol – протокол управления менеджером X дисплея) – специальным протоколом управления менеджером дисплея. Согласно спецификации X Open Group стандарт XDMCP позволяет использовать менеджер дисплея для подключения к X-серверу удаленных X-клиентов, тем самым обеспечивая распределенность выполнения графических приложений. Конфигурационный файл, контролирующий применение протокола XDMCP, называется /etc/X11/xdm/Xaccess, и в нем по умолчанию запрещается подключаться к X-серверу удаленным пользователям

(более подробно о возможностях удаленного доступа к X-серверу можно узнать из главы "Настройка сервера").

Рассказывать о возможностях применения системы X-Windows можно еще очень долго. Важно понять, что данная система предоставляет пользователю, наверное, самые широкие возможности по настройке и управлению, из всех графических систем такого класса. Но для полноценного использования этого потенциала конечному пользователю и администратору следует хорошо разобраться в принципах функционирования этой системы, чтобы потом с успехом применить полученные теоретические знания на практике.

Глава 4. Аппаратная база X-терминальной сети

Системные требования к серверу X-терминалов

Выбор сервера X-терминалов довольно ответственный шаг, так как это практически единственный случай в настройки X-терминальной системы, когда вам придется купить новый компьютер, хотя вполне возможно, что у вас уже имеется подходящая техника. Как говорит один мой знакомый системный администратор, мощность компьютера измеряется не в мегагерцах, а в штуках процессоров на материнской плате, и по отношению к серверу X-терминалов он безусловно прав. Посудите сами, сервер X-терминалов предназначен для одновременной работы нескольких пользователей, а многопроцессорная система лучше справляется с параллельными вычислениями чем ее однопроцессорный аналог. Как обычно при выборе любого компьютера все упирается в экономическую целесообразность и те задачи, которыми пользователи будут нагружать ваш сервер. Например, вы планируете установить X-терминал сервер в локальной сети с 10-клиентами. Если ваши пользователи будут пользоваться текстовым редактором, электронной почтой и файловым менеджером, то им вполне подойдет сервер с минимальными характеристиками (производительностью), а если они будут заниматься 3-х мерным моделированием, то скорее всего им и 2-х процессорного Xeon-а будет мало. Хорошей идеей будет приобретение компьютера с техническими характеристиками на класс выше, чем те что уже есть у вас в распоряжении, так как тем самым вы обеспечите себе запас производительности на несколько лет

вперед (ну хотя бы на год, два). Несмотря на довольно размытые рамки требований к серверу X-терминалов все же можно зафиксировать начальные условия:

Оперативная память - самое критичное ограничение для сервера X-терминалов. Если памяти будет мало, то операционная система Linux вынуждена будет постоянно использовать виртуальную память из swap-раздела, а это приведет к значительному снижению скорости работы сервера, а следовательно и клиентов. Я бы рекомендовал рассчитывать оперативную память сервера по такой простой формуле:

Память сервера = 512 Мбайт + Память для одного клиента * Количество клиентов

Память, дополнительно выделяемая одному клиенту, опять же зависит от тех задач, которые он выполняет. В системах, что мне довелось использовать она колебалась от 16 до 128 Мбайт, то есть, если ваш сервер X-терминалов обслуживает 5 клиентов и дополнительная память каждому клиенту составляет 64 Мбайт, то минимальное значение оперативной памяти сервера должно равняться: 512 + 64 * 5 = 832 Мбайт. Хотя с возрастанием количества клиентов эта зависимость может и не так строго соблюдаться, но в любом случае 512 Мбайт – это базовый минимум для сервера, и устанавливать меньший объем памяти можно только в тестовых или учебных целях;

Процессор – второй по важности компонент сервера X-терминалов. Он должен быть быстрый и желательно чтобы их было много (шутка, с долей истины). Конечно, чем процессоров больше тем лучше, но как правило, решение о покупки нового компьютера принимает не системный администратор, а его непосредственный начальник (директор организации), поэтому если вам не дадут добро на покупку 4-х процессорного "монстра", не стоит отчаиваться, так как и с одним процессором может получится очень хороший сервер X-терминалов. Из общих характеристик следует отдать предпочтение процессору который имеет: больше кеш памяти, большую скорость шины FSB и номинальную частоту. Сервер X-терминалов будет работать круглосуточно (или весь рабочий день), а следовательно к

температурному режиму процессора следует относится с большой осторожностью. Если выбирать между быстрым и надежным процессором, то предпочтение лучше отдать последнему. Для операционной системы Linux вполне подойдут даже 64-х битные процессоры, так как все программное обеспечение можно собрать (скомпилировать) самому, а 64-х битные вычисления в будущем будут очень кстати. Если вернуться к сегодняшним реалиям, то для однопроцессорного компьютера я посоветовал бы в качестве центрального процессора использовать Intel Pentium 4, а для многопроцессорных систем – Intel Xeon, хотя это дело вкуса. Я думаю понятно, что лучше если это будет x86-совместимый процессор, так как эти процессоры занимают господствующее положение на рынке и именно на них эксплуатируется большинство Линукс-систем. В случае возникновения проблемы, решить ее будет гораздо легче для Intel-совместимого процессора, нежели для Alpha или скажем PowerPC. Ни в коем случае не хочу обидеть почитателей продукции компании Advanced Micro Devices, Inc. (официальный сайт компании http://www.amd.com/), так как данной фирмой выпущено очень много достойных изделий. Вообще в последнее время компания AMD значительно улучшила качество выпускаемых ею чипов, а, например, ее подход в организации многопроцессорных систем вообще заслуживает всяческих похвал. Но лично мне ближе по идеологии продукция с логотипом "Intel Inside" (официальный сайт Intel Corp. - http://www.intel.com/), так как я на собственном опыте убедился в ее очень высокой надежности. Чтобы еще раз развеять миф о гипертрофированных требованиях к мощности центрального процессора сервера X-терминалов хочу привести один реальный пример из жизни. Так, в моей практике был случай, когда сервер X-терминалов работал на компьютере, где в качестве центрального процессора использовался Intel Celeron Tualatin 1ГГц. Причем так продолжалось в течении года и четыре его пользователя были вполне удовлетворены его работой.

Когда основной материал данной книге был подготовлен, на компьютерном рынке массово начали появляться двухядерные процессоры (Intel и AMD практически одновременно объявили о начале поставок). С точки

зрения особенностей эксплуатации сервера X-терминалов двухядерные процессоры – это идеальный выбор в соотношении цена/производительность. В отличие от многопроцессорных систем они не так дороги, но при этом показывают очень хорошую производительность в многозадачных средах, а именно такой и является работа сервера X-терминалов;

Жесткий диск – также очень важный компонент сервера X-терминалов, а в организациях, где информация стоит дорого – самый важный. Первый попавшийся под руку “винчестер” вряд ли будет подходящим выбором. Лучше если это будет накопитель SCSI, так как этот интерфейс лучше подходит для выполнения параллельных операций чтения/записи, но учитывая стоимость таких накопителей, можно ограничиться и жестким диском с интерфейсом IDE или SATA. Так как на жестком диске сервера X-терминалов будет храниться информация всех пользователей, то внезапный его отказ поставит их, а следовательно и системного администратора, в трудное положение. Неплохой альтернативой одному “винчестеру” будет RAID-массив (в режиме зеркала), или на худой конец два жестких диска – один рабочий, а второй – для резервного копирования информации. Так как скорость работы винчестера тоже очень важна, а быстрые диски сильно греются, то дополнительный вентилятор никогда не будет лишним. Кэш памяти для упреждающего чтения тоже пусть будет как можно больше, так как работа в многопользовательской системе довольно значительно нагружает дисковую подсистему, а пользователи почему-то любят сохранять свои файлы одновременно и, как правило, в 16 часов 55 минут;

Материнская плата – подойдет любая, но лучше серьезного, проверенного производителя, а не NoName Inc. Вообще к материнской плате сервера X-терминалов, как и к другим его компонентам следует предъявлять несколько завышенные требования, так как выход одного из них из строя практически парализует работу всех X-терминал клиентов. Подумайте, во что обойдутся восстановительные работы перед принятием окончательного решения о покупке дешевого комплектующего. При прочих равных, предпочтение следует отдавать той материнской плате, которая имеет много свободных слотов

расширения (PCI, PCI-E), так как возможно в будущем к серверу X-терминалов будут добавляться дополнительные сетевые интерфейсы для подключения новых “тонких” клиентов. Если материнская плата имеет интегрированную сетевую карту (практически норма на сегодняшний день), то это также стоит только приветствовать, так как в сервере X-терминалов много сетевых карт не бывает;

Источник бесперебойного питания – обязательный атрибут любого сервера, а сервер X-терминалов тоже не исключение из этого правила. Без источника бесперебойного питания ваш сервер X-терминалов автоматически переходит в группу риска. Хотя, если вам нравится восстанавливать операционную систему, проверять разделы жестких дисков, и отвечать на вопросы пользователей типа: “Куда пропал мой квартальный отчет?”, то можете попробовать работать без него. Источник бесперебойного питания должен обеспечивать работу сервера в течении времени, необходимого для нормального завершения его работы. Благо сейчас в комплекте с "бесперебойниками" идет программное обеспечения под операционную систему Linux, которое позволяет автоматизировать процесс выключения компьютера даже если рядом никого нет. После приобретения источника бесперебойного питания, следует обязательно проверить его работу в "полевых" условиях. Для этого достаточно подключить к нему сервер X-терминалов и после полной разрядки батарей отключить питание. Время, в течении которого сервер проработает на автономном энергопотреблении и будет той критической отметкой надежности источника бесперебойного питания. Разделив это значение на 3 (запас прочности) мы получим время после которого необходимо осуществлять автоматическое завершение работы сервера X-терминалов;

Блок питания и корпус – подойдут тоже любые, но так как сервер будет работать круглосуточно, то лучше чтобы блок питания был хорошего качества. Лично я бы не гнался за Ваттами, а большее внимание обратил бы на стабильность подаваемого питания. Также всем пользователям персональных компьютеров известна ситуация, когда через полгода после покупки

компьютера его блок питания начинает издавать жуткие звуки при включении, первопричиной которых является скопившаяся в нем пыль. В любом случае приготовьтесь регулярно чистить ваш сервер X-терминалов, так как пыль является причиной не только неприятных звуков, но еще и выхода из строя аппаратного обеспечения. Желательно выбирать корпус по просторнее, так как в таком системном блоке легче обеспечить хорошую вентиляцию, а следовательно, продлить жизнь и повысить надежность работы сервера X-терминалов. Наличие дополнительных вентиляторов также не повредит, главное знать меру, так как 1 или 2 – это хорошо, а вот уже больше 3-х - перебор;

Сетевые карты. Лично я не сторонник покупки сетевых карт производства фирмы 3Com для использовании в сервере X-терминалов. Эта компания выпускает отличную продукцию, но анализируя весь свой опыт работы в операционной системе Linux я не припомню случая неправильной работы сетевой карты. Да и за весь период моей работы с персональными компьютерами из строя выходили только те карты, которые соединялись коаксиальным кабелем. Поэтому вам подойдет практически любая сетевая карта, главное условие – наличие драйвера для этой карты (соответствующего модуля ядра в операционной системе Linux). Понятно, что скорость работы карты очень важна (чем больше тем лучше). Настоятельно не рекомендую использовать Ethernet карты рассчитанные на 10 Мбит/с. Их вы можете поставить на X-терминал клиент, но для сервера нужны Fast Ethernet адаптеры со скоростью работы в 100 Мбит/с или даже 1000 Мбит/с;

Флоппи-дисковод, привод CD-ROM, звуковая карта – для сервера практически бесполезны. Привод CD-ROM может понадобиться при установке операционной системы и программного обеспечения на сервер, а в дальнейшей эксплуатации в нем отпадет надобность (так что после настройки сервера смело вынимайте из него CD-ROM). Непосредственно работать на сервере не рекомендуется, так как я лично был свидетелем того, как пользователь сервера выключал его уходя на обеденный перерыв, под крики “У меня монитор погас!” остальных пользователей X-терминалов. Хотя, с другой стороны,

привод CD-ROM лично я все же оставил бы, так как он может понадобится для выполнения профилактических или восстановительных работ;

Монитор, “мышка” и клавиатура также нужны серверу только для первичной установки и настройки программного обеспечения. Все остальные манипуляции с ним можно производить через средства удаленного доступа (например, ssh). Хотя клавиатуру я бы оставил, мало ли что, и к тому же некоторые компьютеры очень отрицательно относятся к отсутствию клавиатуры показывая на экране сообщения типа: “Клавиатура не найдена, для продолжения нажмите любую клавишу”. Очень помогает в работе системному администратору KVM-коммутатор, который позволяет подключить к одному монитору, клавиатуре и “мышке” сразу несколько персональных компьютеров. Управляя сервером X-терминалов при помощи KVM-коммутатора, системный администратор всегда может оперативно выполнять административные задачи, при этом параллельно работая за другим компьютером.

Помещение, где будет установлен сервер также нужно брать в расчет. Желательно, чтобы это был отдельный, закрываемый на ключ кабинет, но подойдет и небольшой закрытый коммуникационный шкаф. Сервер X-терминалов пусть и не является сервером в традиционном понимании этого слова (например, он может не использовать сервера баз данных), но централизованная обработка и хранение информации на лицо, а следовательно требования значительно выше чем к рядовому персональному компьютеру.

Если вы не уверенны в том, что точно осознали требования к серверу X-терминалов, то в тестовых целях для первого раза можете взять любой имеющийся в вашем распоряжении компьютер оснащенных, хотя бы, 256 Мбайтами оперативной памяти, и попробовать на нем настроить сервер X-терминалов. После этого подключитесь к нему с X-терминал клиента и попробуйте как все работает. Думаю это придаст вам уверенности в том, что требования к X-терминал серверу могут быть весьма скромными. В моей практике был случай, когда я только начал разбираться с X-терминалами и для экспериментов начальство выделило мне персональный компьютер менеджеров, на котором работало поочередно трое человек. Мне сказали, что

когда никто за компьютером не работает, я могу с ним вволю экспериментировать. Системные требования этого компьютера были очень скромные: Intel Pentium II 350 МГц, 256 Мбайт ОЗУ, жесткий диск Samsung 20Gb 5400 об/мин и сетевая карта Ethernet с чипсетом RTL 8029 AS на 10 Мбит/с. Руководствуясь полученными директивами, я все потихоньку настроил и на X-терминале, который стоял в другой комнате (на втором этаже), я начал тестировать работу сего нововведения. По-началу все было хорошо, но потом X-терминал начал явно тормозить, а индикатор загрузки центрального процессора стал на отметку в 100%. Я решил выяснить, что же такое делают пользователи моего первого импровизированного “сервера”. После спуска на первый этаж я обнаружил их мирно играющими в “Return to castle Wolfenstein”.

В заключении рассказа о X-терминал сервере приведу несколько замечаний по поводу требований к его аппаратному обеспечению. Все клиенты X-терминалов работая на сервере одновременно используют его ресурсы. И если они не играют в 3-х мерные игры, то загрузка центрального процессора далека от 100%. Для офисной работы она находится на уровне 3% и возрастает только в моменты запуска программ и открытия/закрытия файлов. Подчиняясь теории вероятности люди иногда работают в разное время, а иногда одновременно, и в такие моменты нагрузка на сервер резко увеличивается. Помню в одной статье посвященной X-терминалам был предложен “выход” из этого положения: пользователям достаточно договориться и асинхронно нажимать клавиши на клавиатуре, то есть нажали вы, затем ваш сосед и так далее. Конечно, это шутка, но она затрагивает одну очень важную проблему: возникновение пиков нагрузки сервера. Если вспомнить теорию массового обслуживания, то работа сервера X-терминалов очень напоминает магазин с кассовыми аппаратами на выходе. Пользователи – это покупатели, а тележки с товаром это приложения которые они используют. Кассовые аппараты это центральные процессоры сервера, чем их больше тем и быстрее и качественнее происходит обслуживание покупателей (выполнение программ пользователей), но и дороже обходится их эксплуатация (каждому кассиру нужно платить зарплату). Если покупатель купил много товара, он как бы соответствует

ресурсоемкому приложению, и обслуживание такого покупателя значительно замедлит обслуживание других, особенно в случае магазина с одном кассой (сервер с одним процессором).

Если посмотреть на эту проблему с точки зрения пользователей, то пик нагрузки на сервер выглядит следующим образом. Например, для открытия большого многостраничного документа электронной таблицы в табличном процессоре OpenOffice Calc требуется 20 секунд. При этом ресурсы вашего сервера задействованы на 100%. Если подобные документы открывают одновременно 2 пользователя, то сервер поделит свои ресурсы между ними и на открытие документов оба затратят по 40 секунд (а то и больше за счет накладных расходов многозадачности), если же одновременно работающих пользователей будет 3, то каждый из них свой документ будет открывать больше минуты. Увеличение количества процессоров сервера значительно сглаживает эти скачки интенсивной загрузки, но не избавляет от них. Иначе говоря, работая на X-терминале следует быть готовым к колебательному характеру загрузки сервера и фактически работы X-терминала. В один период времени ваш X-терминал будет работать быстрее, а иногда его работа будет замедляться, но даже при значительной нагрузке на сервер все приложения получат равные порции процессорного времени, а следовательно будут правильно выполняться.

Кроме нагрузки на центральный процессор сервера, значительно замедляет работу X-терминал сервера и механизм виртуальной памяти. Это еще раз подчеркивает необходимость использования в сервере X-терминалов значительного объема оперативной памяти. Именно значительного, а не огромного. Никому не секрет, что работать в среде X-Windows на персональном компьютере с операционной системой Linux, у которого объем оперативной памяти меньший чем 256 Мбайт не очень комфортно. Но это вовсе не означает, что для работы 10 X-терминалов сервер должен иметь оперативную память равную 2,5 Гбайт (256 Мбайт * 10). Дело в том, что все современное программное обеспечение в своей работе использует разделяемые библиотеки. А программный код этих библиотек загружается в оперативную

память сервера только в единственном экземпляре для использования его всему работающими программами. Поэтому, если все ваши пользователи X-терминалов запустят OpenOffice, то дополнительно потребуется памяти только на хранение их документов, а динамические библиотеки будут использоваться ими совместно. Благодаря такой структуре хранения информации программы на X-терминалах запускаются значительно быстрее, чем аналогичные на традиционных персональных компьютерах. И связано это не только со скоростью сервера, но и с использованием уже загруженных в память разделяемых библиотек (прирост скорости наблюдается за счет отсутствия необходимости повторной загрузки кода разделяемых библиотек в оперативную память сервера X-терминалов).

Есть еще один вид замедления работы на X-терминале – возникновение сетевых “пробок”, но он связан с загрузкой канала локальной сети и про это пойдет в разделе "Аудит сети X-терминалов" главы "Администрирование сети X-терминалов".

Требования к аппаратному обеспечению клиентов

Хотя требования к X-терминал клиенту (дальше просто X-терминалу) очень простые, но из любого компьютера может и не получится полноценный X-терминал, поэтому не спешите доставать из чулана “Поиск”. Для X-терминала необходимо наличие следующих компонентов:

Процессор – от 386 и выше (лучше 486 или Pentium);

Материнская плата – желательно с шиной PCI;

Память – любого типа (FPM, EDO, SDRAM, DDR, ...) от 8 Мбайт (лучше от 16 Мбайт);

Видеоадаптер – PCI, AGP, PCI Express или интегрированный на материнской плате (желательно, чтобы он поддерживался проектом XFree86 4.0 и выше);

Сетевая карта – от 10 Мбит/с с возможностью загрузки по сети (BootROM или PXE);

Корпус с блоком питания – любой работающий;

Монитор – VGA 640x480 или лучший (желательно чтобы он поддерживал разрешающую способность 800x600 и выше);

Клавиатура – любая;

Манипулятор “мышь” - лучше со скроллером. Интерфейсы: COM, PS/2 или USB.

Как видите требования довольно скромны, но все же ограничения есть. Во-первых, это процессор x86-совместимый, так как для процессоров другой архитектуры могут возникнуть проблемы с загрузкой по сети. Во-вторых, ниже 386-го процессора в X-терминалах использовать нельзя. В-третьих, материнская карта без интегрированного видео и без PCI слотов не подойдет для X-терминала.

Если вы окончательно решили использовать X-терминалы в своей работе, то прежде чем бежать в магазин, рекомендую осмотреться вокруг и обратить свой взор к уже имеющемуся парку компьютерной техники. Весьма вероятно, что среди ваших компьютеров уже есть "потенциальные кандидаты на покой", то есть на обретение второй жизни в X-терминале.

Делаем терминал из персональных компьютеров серии 386/486/Pentium

Считаем, что вы нашли персональный компьютер "преклонных годов", который согласился стать X-терминалом. Тут конечно раздолье для специалистов учебных заведений, так как эти организации издавна славились обилием "доисторической" техники (заходи в любой учебный компьютерный класс и выбирай).

Методика переделки персонального компьютера в X-терминал довольно проста и напоминает хирургическую операцию по удалению лишней оперативной памяти, “винчестера”, привода CD-ROM и возможно флоппи-дисковода и пересадки сетевой карты.

Будем последовательными. Разбираем системный блок персонального компьютера (см. рис. 4.1). Извлекаем от туда винчестер и привод компакт-дисков вместе с интерфейсными шлейфами. Флоппи-дисковод оставляем, так как он будет использован для проверки сетевой загрузки. Смотрим на модули памяти: необходимый минимум составляет 8 Мбайт, но лучше использовать 16

Мбайт и выше, так как в такой конфигурации не нужно будет использовать сетевой swap-файл. В принципе, памяти больше 64 Мбайт оставлять не нужно, но если у вас модуль DIMM SDRAM 128 Мбайт, а другого просто нет, то такой тоже сгодится.

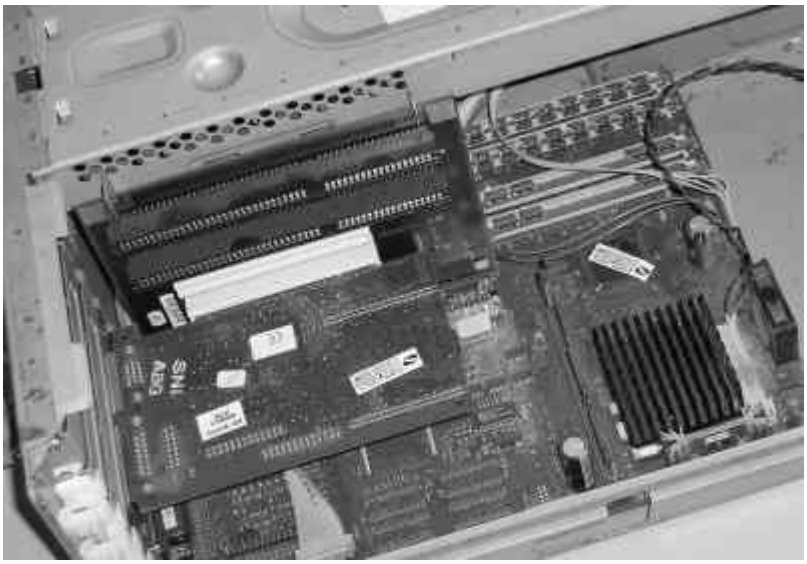

Рис. 4.1. "Внутренности" потенциального X-терминала

Очень внимательно изучаем видео-карту. От ее качества очень сильно зависит успешная работа на X-терминале. Где-то мне встретилось мнение, что для X-терминала следует подбирать особо тщательно монитор и видеокарту, так как от качества монитора зависит усталость глаз пользователя, а от видеокарты – работа в конкретном графическом режиме и частота регенерации экрана. Конечно, следует исходит из того, что доступно и есть у нас в наличии, но в некоторых случаях следует быть готовым к замене видеоадаптера. Так, в нашей стране, и я подозреваю в остальном мире тоже, огромное распространение получили видеоадаптеры PCI с чипом s3trio (благодаря своей дешевизне). Так вот, большинство из этих видео-карт не лучшие кандидаты на роль видеоадаптера в X-терминале. Эта проблема скрыта в драйверах проекта XFree86. Для старых карт семейства s3 необходим старый X-сервер версии 3.3.6, а он кроме проблем с безопасностью имеет еще и проблемы использования сглаженных шрифтов и шрифтов типа TrueType. Иначе говоря для видеокарт не все так просто и хорошо, но практически любую карту можно заставить работать в VESA совместимом режиме без использования аппаратной акселерации.

Чтобы было проще разобраться подходит видеоадаптер для X-терминала или нет, ниже приведено 2 таблицы совместимости видеокарт (в эти списоки входят только старые видеокарты, так как вряд ли ваш X-терминал будет оснащен Radeon X800). В первой Таблице 4.1 перечислены видеоадаптеры, которые поддерживаются сервером XFree версии 4.x и поэтому они являются прекрасным выбором для работы в X-терминале. В Таблице 4.2 собраны видеоадаптеры, для работы которым требуется X-сервер версии 3.3.6, а следовательно их использование возможно, но сопряжено с определенными неудобствами (отсутствие поддержки сглаженных шрифтов и шрифтов TrueType, потенциальные проблемы безопасности и т.д.).

Таблица 4.1. Список видеоадаптеров, использующих сервер XFree86 версии 4.x

Производитель	Драйвер	Тип видеочипа (марка, модель)
ATI	ati	210888CXMach64 CX
		210888GXMach 64 GX (WinTurbo)
		215CT222Mach 64 CT
		215VT222Mach 64 VT VIDEO XPRESSION
		68800AXMach 32
		DRAGE PRO 2X AGP ATI ALL IN WONDER PRO (8MB
		Mach64 GT Rage 3D II Graphics Accelerator
		Mach64 LT
		Mach64 ST
		Mach64 VT3
		Mach64 VT4 PCI
		Mach64ET
		Mach64VT
		MobilityM3 AGP 2x
		R200Radeon 2 - AGP
		RadeonDDR/SDR
		RadeonVE

		Rage128 4x
		Rage128 GL AGP 2x
		Rage128 GL PCI
		Rage128 Pro PA PCI
		Rage128 Pro PB AGP 2x
		Rage128 Pro PC AGP 4x
		Rage128 Pro PD PCI
		Rage128 Pro PE AGP 2x
		Rage128 Pro PF AGP 4x
		Rage128 Pro PG PCI
		Rage128 Pro PH AGP
		Rage128 Pro PI AGP
		Rage128 Pro PJ PCI
		Rage128 Pro PK AGP TMDS
		Rage128 Pro PL AGP TMDS
		Rage128 Pro PM PCI
		Rage128 Pro PN AGP
		Rage128 Pro PO AGP
		Rage128 Pro PP PCI TMDS
		Rage128 Pro PQ AGP TMDS
		Rage128 Pro PR AGP TMDS
		Rage128 Pro PS PCI
		Rage128 Pro PT AGP
		Rage128 Pro PV PCI TMDS
		Rage128 Pro PW AGP TMDS
		Rage128 Pro PX AGP TMDS
		Rage128 SE PCI
		Rage128 SF AGP 2x
		Rage128 SG AGP 4x
		Rage128 VR AGP 2x

		Rage128 VR PCI
		Rage3D II+
		Rage3D IIC
		Rage3D IIC PCI Graphics Accelerator
		Rage3D LT Pro AGP 133 MHz
		Rage3D LT Pro AGP 66 MHz
		Rage3D LT Pro PCI
		Rage3D LT-G
		Rage3D Pro
		RageL Mobility AGP 2x
		RageL Mobility PCI
		RageMobility 128 AGP 4x
		RageP/M Mobility AGP 2x
		RageP/M Mobility PCI
		RageXC AGP 2x
		RageXC PCI
		RageXC PCI-66
		RageXL AGP 2x
		RageXL PCI
		RageXL PCI-66
Tseng Labs	tseng	ET4000W32P-A GUIAccelerator
		ET4000W32P-B GUIAccelerator
		ET4000W32P-C GUIAccelerator
		ET4000W32P-D GUIAccelerator
		ET6000Graphics/Multimedia Engine
		ET6300
DEC	tga	DC21030 PCI GraphicsAccelerator
		DC21130 PCIIntegrated Graphics & Video Accel
		TGA2
Cirrus Logic	cirrus	CL-GD5432 Alpine GUIAccelerator

		CL-GD5434 Alpine GUIAccelerator
		CL-GD5436 Alpine GUIAccelerator
		CL-GD5436U
		CL-GD5446 64-bitVisualMedia Accelerator
		CL-GD5462 LagunaVisualMedia graphics accelerator
		CL-GD5464 Laguna 3DVisualMedia Graphics Accel
		CL-GD5464 Laguna BD
		CL-GD5465 Laguna 3DVisualMedia Graphics Accel
		CL-GD5480 64-bitSGRAM GUI accelerator
		CL-GD7548GUI-Accelerated XGA/SVGA LCD Controller
IBM	vga	GXT-150P GrqaphicsAdaptor
TridentMicrosystems	trident	32-bit GUIAccelerator
		3D Image 9850 AGP
		3DImage 9750 PCI/AGPtrident dgi
		Blade 3D PCI/AGP
		Cyber9382
		Cyber9383
		Cyber9385
		Cyber9386 VideoAccelerator
		Cyber9388 VideoAccelerator
		Cyber9397 VideoAccelerator
		Cyber9397DVD VideoAccelerator
		Cyber9520 VideoAccelerator
		Cyber9525 VideoAccelerator
		Cyber9540 VideoAcclerator
		CyberBlade i1 AGP
		CyberBlade i1 AGP 51
		CyberBlade i7

		CyberBlade i7 AGP
		CyberBlade XP
		CyberBlade-A i1
		ProVIDIA 9685
		TGUI9320 32-bit GUIAccelerator
		TGUI9420 DGi GUIAccelerator
		TGUI9440 DGi GUIAcclerator
		TGUI9460 32-bit GUIAccelerator
		TGUI9470
		TGUI9660XGi GUIAccelerator
		TGUI9680 GUIAccelerator
		TGUI9682 MultimediaAccelerator
		TGUI9683 GUIAccelerator
		TGUI975? Image GUIAccelerator
		TGUI9753 VideoAccelerator
		TGUI9753 Wave VideoAccelerator
		TGUI9783
		TGUI9785
Matrox	mga	Fusion G450 AGP
		Fusion Plus G800 AGP
		Meteor 2/MC VideoCapture Card
		MGA 1064SG 64-bitgraphics chip
		MGA-1164SG Mystique220 (AGP)
		MGA-2064W MilleniumGUI Accelerator
		MGA-21164W MilleniumII
		MGA-G100
		MGA-G100 ProductivaG100 Multi-Monitor
		MGA-G200Millennium/Mystique G200 AGP
		MGA-G200BMillennium/Mystique G200 AGP
		MGA-G800

		MGA-I Impression
		MGA-PX2085Ultima/Atlas GUI Accelerator
		MGA2164WA-B MatroxMillenium II AGP
		Mistral GUI+3DAccelerator
Chips andTechnologies	chips	64310 Wingine DGX -DRAM Graphics Accelerator
		65540 Flat Panel/CRTVGA Controller
		65545 Flat panel/crtVGA Cntrlr
		65548 GUI Accelerator
		65550 LCD/CRTcontroller
		65554 Flat Panel/LCDCRT GUI Accelerator
		65555 VGA GUIAccelerator
		65560 PCI FlatPanel/CRT VGA Accelerator
		68554 GUI Controller
		68554 HiQVision FlatPanel/CRT GUI Controller
		68555 GUI Controller
		69000 AGP/PCI FlatPanel/CRT VGA Accelerator
		69030 AGP/PCI FlatPanel/CRT VGA Accelerator
Silicon IntegratedSystems	sis	SiS 315
		SiS 6201/02 PCITrue-Color Graphics Accelerator
		SiS 6205 PCI Graphics& Video Accelerator
		SiS 6215 PCI Graphics& Video Accelerator
		SiS 6225 PCI Graphics& Video Accelerator
		SiS 635
		SiS 730x
		SiS 735
		SiS 740
		SiS 86C326 AGP/PCIGraphics & Video Accelerator
		SiS300/305/630 GUIAccelerator+3D
		SiS3052D/3D/Video/DVD Accelerator
		SiS530/620 Integrated3D VGA Controller

		SiS540 AGP
		SiS5597/98 OnboardGraphics Controller
		SiS6205 PCI VGAController
		SiS6236 Graphics
		SiS630 AGP
Number Nine VisualTechnology	i128	I128s2 Imagine 128Series 2
		Imagine 128 GUIAccelerator
		Revolution IVRevolution IV
		T2R Revolution 3D
Cyrix Corporation	cyrix	Cx5530 VideoController
NeomagicCorporation	neomagic	Graphics Cntrlr
		NM2097 GraphicController NeoMagic MagicGraph128ZV+
		NM2160 MagicGraph128XD
		NM2200 MagicMedia256AV
		NM2230 MagicMedia256AV+
		NM2360 MagicMedia256ZX/256M6D
		NM2380 MagicMedia256XL+
Nvidia Corporation	nv	GUI Accelerator
		Mutara V08 (NV2)
		NV10 GeForce 256
		NV10 GeForce 256 DDR
		NV10 GeForce 256 GL(Quadro)
		NV10 GeForce 256Ultra
		NV11 DDR GeForce2 MXDDR
		NV11 GeForce 2 MX
		NV11 GeForce2 Go
		NV11 GL GeForce2 MXGL (Quadro2 MXR)
		NV15 BladerunnerGeForce2 GTS Ultra
		NV15 DDR GeForce2 GTSDDR

		NV15 GeForce2 GTS/Pro
		NV15 GL GeForce2 GTSGL (Quadro2 Pro)
		NV20 GeForce3
		NV20BR GeForce 3Ultra
		NV20GL Quadro DCC
		NV2A GeForce 3Integrated (Xbox)
		Riva 128 Riva 128accelerator
		Riva 128 ZX
		Riva TNT AGP
		Riva TNT2 (NV5)
		Riva TNT2 (Pro)
		RIVA TNT2 Aladdin
		Riva TNT2 M64 RivaTNT2 Model 64
		Riva TNT2 Ultra
		Targa 1000 VideoCapture & Editing card
		Vanta (LT)
		VANTA (NV6)
Integrated MicroSolutions	imstt	IMS Twin Turbo 128
		MS Twin Turbo 3D
Intergraph	fbdev	CyberPro 2010
Creative Labs	nv	GeForce 256 DDR NvidaCorp. Video
	tdfx	Banshee 3D BlasterBanshee PCI CT6760
AllianceSemiconductor	apm	6410 6422 GUIAccelerator
		AT25 ProMotion-AT3D
		GUI Accelerator
		ProMotion 3210VGA/AVI Playback Accelerator
		ProMotion AT25
		ProMotion aT3D
		Provideo 6422
		ProVideo 6424ProMotion AT24 GUI Accelerator

Rendition Inc	rendition	Verite 1000 3DBlaster
		Verite 2000
3dfx InteractiveInc	tdfx	Voodoo Banshee VoodooBanshee
		Voodoo Voodoo 3DAcceleration Chip
		Voodoo2 Voodoo 2 3DAccelerator
		Voodoo3 All Voodoo3chips, 3000
		Voodoo4
		Voodoo5
Silicon Motion	siliconmotion	SM710 LynxEM
		SM712 LynxEM+
		SM720 Lynx3DM
		SM810 LynxE
		SM811 LynxE
		SM820 Lynx3D
		SM910 Lynx
NVidia / SGSThompson	nv	DAC64 EDGE 3D
		NV4 Riva TNT GUI+3DAccelerator
		NV5 Riva TNT2
		NV5 Riva TNT2 /TNT2Pro
		NV6 Vanta
		NVA0 Riva TNT2Aladdin
		NVM64 Riva TNT2 Model64
		NVULTRA Riva TNT2Ultra
		NVVANTA Vanta / VantaLT
		RIVA 128 Riva 1282D/3D GUI Accelerator
		RIVA 128ZX 2D/3D GUIAccelerator
3dlabs Inc. Ltd	glint	3C0SX 2D+3DAccelerator
		3D Extreme PermediaII 2D+3D Accelerator
		GLint 300SX 3DAccelerator
		GLint 500TX Sapphire3D Accelerator

		GLint Delta Geometryprocessor
		GLint Gamma G1
		GLint MX 3DAccelerator
		GLint VGA
		Permedia 2D+3DAccelerator
		Permedia 3
		Permedia II 2D+3DAccelerator
		Permedia2v 2d+3dchipset, integrated ramdac
S3 Incorporated	s3virge	386C325 Virge 3D GUIAccelerator
	s3virge	386C359 ViRGE /GX2+Macrovision
	savage	3Savege4 ProSavagePM133
	s3virge	82C375/86C385 ViRGE/DX & /GX
	s3virge	85C260 ViRGE/M3(ViRGE/MX)
	s3virge	86C240 ViRGE/MXC
	s3virge	86C260 ViRGE/M5(ViRGE/MX)
	savage	86C270/274/290/294Savage MX/IX/MX+MV/IX+MV
	s3virge	86C280 ViRGE /MX+Macrovision
	s3virge	86C357/86C359 ViRGE/GX2 & /GX2+
	s3virge	86C359 ViRGE /GX2+
	s3virge	86C362/86C368Trio3D2x & Trio3D2x+ AGP
	s3virge	86C365/366 Trio3D
	savage	86C390 Savage3D/MV
	savage	86C390/391 Savage3D
	savage	86C394-397 Savage 4
	savage	86C395B ProSavage
	savage	86C410 Savage 2000
	savage	86C508 SuperSavage128/MX
	savage	86C544 SuperSavage128/IX
	savage	86C553 SuperSavage128/IX DDR
	savage	86C564 SuperSavage/IX

	savage	86C573 SuperSavage/IXDDR
	savage	86C583SuperSavage/IXC SDR
	savage	86C594SuperSavage/IXC DDR
	s3	86C764/765 Trio64/64V+ GUI Accelerator
	s3	86C964 Vision 964 GUIAccelerator VRAM rev. 0
	s3	86C964 Vision 964-PGUI Accelerator VRAM rev. 1
	s3	86C968 Vision 968 GUIAccelerator VRAM rev. 0
	s3virge	86C988 ViRGE/VX 3DGUI Accelerator
	s3	86CM65? Aurora 64V+
	savage	Via VT8233a chipsetsavage video
Intel Coorporation	i740	82740 AGP GraphicsAccelerator
	i810	82810E GraphicsDevice
		82810E GraphicsDevice (FSB 133 MHz)
		82810E GraphicsDevice (FSB 133 MHz)
		82810M-DC100 Systemand Graphics Controller
		82815 InternalGraphics Device
		i810-0
		i810-1
		i810-DC100-0
		i810-DC100-1
		i815-0
		i815-1
		i840
		i850

Таблица 4.2. Список видеоадаптеров, использующих сервер XFree86 версии 3.3.6

Производитель	Сервер	Тип видеочипа (марка, модель)
Avance Logic	XF86_SVGA	iALG2032/2064 alg203263067s1
		ALG2364A
		ALG2301 GUIAccelerator

		ALG2302 GUIAccelerator
		AL2364 GUIAccelerator
		ALG2564A/25128A
Cirrus Logic	XF86_SVGA	CL-GD7555 Flat PanelGUI Accelerator
		CL-GD7556 64-bitAccelerated LCD/CRT Controller
		CL-GD7542 Nordic GUIAccelerator
		CL-GD7543 Viking GUIAccelerator
		CL-GD7541 Nordic-liteVGA Cntrlr
TridentMicrosystems	XF86_SVGA	TGUI9430 GUIAccelerator
NVidia / SGSThompson	XF86_SVGA	NV1 EDGE 3DAccelerator
Avance Logic Inc.	XF86_SVGA	ALG2301 GUIAccelerator
		ALG2302 GUIAccelerator
S3 Incorporated	XF86_S3	86C551 Plato/PX
		86C732 Trio 32 GUIAccelerator rev. 0
		86C764 Trio 32/64 GUIAccelerator v3
		86C767 Trio 64UV+ GUIAccelerator
		86CM66 Aurora128
		86C868 Vision 868 GUIAccelerator VRAM rev. 0-3
		386C928 Vision 928GUI Accelerator VRAM rev. 0-3
		86C864 Vision 864 GUIAccelerator DRAM rev. 0-3
		86C964 Vision 964-PGUI Accelerator DRAM rev 2-3
		86C968 Vision 968 GUIAccelerator VRAM rev. 1-3
		86C775 Trio64V2/DX
		86C775/86C785 Trio64V2 DX/GX
		TrioV Family
		86C765 Trio64V+comatible
		386C765 Trio64V+compatible

Делаем терминал из персональных компьютеров серии 386/486/Pentium (продолжение)

Видеокарта должна хорошо сочетаться с монитором X-терминала хотя бы по разрешающей способности, глубине цвета и частоте регенерации. Расчет

минимального количества видеопамяти, которое должно быть у видеоадаптера выполнить совсем не сложно. Так, для глубины цвета HiColor (16 бит на пиксель) и TrueColor (24 бита на пиксель) вам поможет следующая таблица (см. Таблица 4.3):

Таблица 4.3. Минимальные требования к памяти видеоадаптеров

Типмонитора	Диагональ	Рекомендуемая разрешающая способность	Память видеоадаптера при глубине цвета 16 бит	Память видеоадаптера при глубине цвета 24 бита
Лучеваятрубка	14"	640x480	1 Мб	1 Мб
	15"	800x600	1 Мб	2 Мб
	17"	1024x768	2 Мб	3 Мб
	19"	1280x1024	3 Мб	4 Мб
Матрица	13"	800x600	1 Мб	2 Мб
	14"	800x600	1 Мб	2 Мб
	15"	1024x768	2 Мб	3 Мб
	17"	1280x1024	3 Мб	4 Мб

Конечно видеопамяти на видеоадаптере лучше иметь побольше (при избытке видеопамяти происходит частичное ускорение работы видеоподсистемы), но так как мы переделываем старый персональный компьютер в X-терминал, то выбирать особо не приходиться. Хорошо, если вы имеете в своем распоряжении несколько старых компьютеров, тогда меняя их комплектующие можно собрать несколько полноценных X-терминалов. При замене видеоадаптера важно учитывать на какой частоте он сможет работать при заданном разрешении экрана. Особенно это важно для старых видеоадаптеров, так как некоторые из них могут только мечтать работать на частоте в 85 Гц. А мерцание монитора очень расстраивает пользователей (поверьте мне на слово). Но у старых адаптеров появился новый союзник в лице LCD-мониторов, так как их частота регенерации редко превосходит 75 Гц, а очень часто нормальная картинка наблюдается лишь при стандартных 60 Гц. За приобретение плоских LCD мониторов говорит и тот факт, что так как мы значительно экономим на покупке нового компьютера, то почему бы часть этих

средств не пустит на покупку хорошего монитора, тем более что плоский монитор в офисе становиться стандартом респектабельности.

С относительно новыми видеокартами (год выпуска начиная с 1998-го) таких проблем нет. Они прекрасно справляются с любой разрешающей способностью, поддерживают практически любую частоту регенерации монитора и редко имеют меньше 8 Мбайт видеопамяти. Тем более, что все видеоадаптеры с AGP-шиной, которые попадали мне в руки, прекрасно работали как в обычных (традиционных) компьютерах с операционной системой Linux так и в X-терминалах.

Если ваш персональный компьютер оборудован видеокартой с интерфейсом ISA, но на материнской плате есть свободный слот PCI, то рекомендую приобрести для этого X-терминала новый видеоадаптер с интерфейсом PCI. Так как скорость работы ISA видеоадаптера вас вряд ли устроит, да и с настройкой его вы точно намучитесь.

Теперь обратимся к сетевому адаптеру. Желательно, чтобы он имел интерфейс PCI (ISA устройства работают медленнее и их настройку выполнять значительно труднее) и работал со скоростью 100 Мбит/с. Операционная система Linux поддерживает практически все сетевые карты, поэтому особых проблем с их настройкой быть не должно. Если по каким-либо причинам нет возможности установить сетевую карту на 100 Мбит/с, то вполне можно использовать сетевой адаптер работающий на скорости в 10 Мбит/с. На работу офисных приложений это практически не повлияет, а вот работа с графикой – значительно замедлится.

Из приведенного выше материала вы должны были понять, что переделать из старого компьютера X-терминал не составляет особого труда, и это ничуть не сложнее обычной модернизации персонального компьютера. Перед сборкой X-терминала не забудь-те почистить все компьютерные комплектующие, так как пыль враг компьютера номер 2, после врага номер 1 – пользователя!

Собираем X-терминал сами

Если по каким-либо причинам в вашем распоряжении нет старого компьютера, но X-терминал вам необходим, то как вариант вы можете собрать

его сами. Для осуществления этого плана вам понадобятся объявления о продаже б/у компьютерной техники в газетах и сети Интернет, а также соответствующие магазины и радио-базары.

Перед тем как собрать терминал следует определится с тем, как он будет загружаться по сети, другими словами на каком носителе памяти будет располагаться его загрузочный образ. Как правило, для загрузки X-терминала по сети используются возможности сетевых карт. Для этого сетевая карта должна иметь функции загрузки по сети: BootROM, PXE или аналогичную технологию. Наиболее распространены карты с BootROM и отличить такую карту очень просто (см. рис. 4.2): она имеет специальный разъем для вставки микросхемы, содержащей загрузочный образ. Как на самой микросхеме, так и на разъеме есть ключ, который помогает правильно ориентировать чип на плате сетевого адаптера (см. рис. 4.3).

Рис. 4.2. Сетевая карта с пустым разъемом BootROM

Рис. 4.3. Сетевая карта с уже установленной микросхемой EPROM

После выбора сетевой карты необходимо найти микросхему для прошивки загрузочного образа. Для этого необходимо пойти в ближайший магазин радиодеталей и купить микросхему с названием 27Cxxx, где xxx – размер чипа в килобитах. Например, 27C64 – соответствует микросхеме емкостью в 64 килобита или 8 Кбайт. Для загрузочного образа, как правило, такой емкости мало, поэтому лучше выбрать микросхему типа 27C128 (16 Кбайт), 27C256 (32 Кбайт) или 27C512 (64 Кбайт). Буква “C” в названии микросхемы, как правило, означает, что информацию можно записывать однократно. Если буквы “C” нет, то у вас многоразовый чип, записать информацию на который можно несколько раз (обычно записывать более одного раза не нужно, а одноразовые чипы значительно дешевле). Если признаться честно, то с маркировкой микросхем для BootROM все не так просто. Во-первых, в начале названия микросхемы могут идти дополнительные малозначащие буквы, которые указывают принадлежность изделия определенному производителю (например, W27C256-45). Во-вторых, в названии чипа через дефис часто записывают особенности той или иной микросхемы. Вы можете даже встретить многоразовый чип, который начинается на "27C", например, M27C512F-12F1. В-третьих, перед покупкой, лучше дополнительно поинтересоваться техническими особенностями микросхемы в магазине радиодеталей.

В загрузочный чип сетевой карты необходимо записать загрузочный образ, который соответствует чипсету сетевого адаптера (про то как выбрать правильный образ вы узнаете из главы “Настройка терминалов”). Записывать

образы в микросхему EPROM нужно с помощью специального устройства – программатора. Если у вас нет программатора, то не нужно отчаиваться и тут же бежать в магазин покупать его. В любой серьезной компьютерной конторе есть не только программатор, но и специалист по работе с ним. Там вам за умеренную плату (в районе 4$) с радостью "прошьют" нужный образ.

Получается, что создание сетевой карты с “прошитым” загрузочный образом стоит совсем не много. По ценам 2004 года микросхема 27C512 стоила 1,5 $, ее прошивка 4 $, а сетевая карта PCI со скоростью 100 Мбит/с на чипсете rtl8139 - 6 $. То есть в итоге имеем 12 $.

Когда выбор сетевого адаптера закончен, переходим к дальнейшей сборки X-терминала – изучению предложений рынка компьютерной техники.

В ближайшем магазине б/у компьютеров я нашел подходящую “четверку” для переделки ее в X-терминал. Характеристики ее были следующие: i486 2DX (2 PCI, 3 ISA), 66 МГц, интегрированное видео 1 Мбайт Cirrus Logic, ОЗУ - SIMM FPM 16 Мбайт, флоппи-дисковод 3,25”, корпус desktop AT 180W, клавиатура и мышка PS/2. Цена такого раритетного “чуда” - 20$. Добавляем сюда 12$ за сетевую карту с возможностью загрузки по сети и имеем системник X-терминала за 20+12 = 32$.

Вот пример интересного объявления в Интернете: “MB - Socket-7, AT, i430TX "ASUS" (3-ISA, 4-PCI, 4-SIMM) и CPU-Pentium-166MMX (work-180mhz) Cooler – 12$, Video-PCI - 1mb "S3Trio64v " - 4$, Case - AT-230w – 4$, SIMM EDO 8MB 72pin (2шт) – 5$, keyboard, mouse – 6$”. Как видно из этого объявления, для X-терминала недостает только загрузочной сетевой карты (+12$). Итого получаем неплохой X-терминал на базе Pentium 166MMX всего за (12+4+4+5+6) +12 = 43$. Как видите, цена вполне приемлемая, учитывая тот факт, что этот компьютер будет работать с самыми современными приложениями.

Дальше в газете нашел такое объявление: “P II, без HDD, FDD, CDROM – цена 56 $”. После звонка по указанному телефонному номеру узнаю детальную конфигурацию: Intel Pentium II 266 МГц Slot 1, материнская плата с чипсетом Intel 440BX, видеоадаптер AGP s3 Virge GX 4Mb с TV-Out, память DIMM

SDRAM 32 Мбайта и 64 Мбайта, звуковая карта ISA ESS 1868, корпус AT 200W. Очень неплохая конфигурация для X-терминала. Так как этот компьютер продается практически по комплектующим, то можно договорится о покупке без звуковой карты (минус 2$) и без DIMM 64Мб (-8 $). В итоге получаем практически готовый системный блок X-терминала за 56-2-8 = 46 $. Конечно в него нужно добавить сетевую карту с загрузочным образом (+12$), клавиатуру (+10$), мышку (+8$). Но все равно, согласитесь, X-терминал без монитора получился по цене 46+12+10+8 = 76$.

Существует достаточно укоренившаяся точка зрения, что покупать б/у (старые) компьютерные комплектующие опасно, так как есть очень большая вероятность выхода их из строя. Спешу вас обрадовать, что к X-терминалу это не относиться, так как в его составе нет механических деталей (CD-ROM, “винчестер”), которые действительно подвержены износу. Зато у старых комплектующих значительно выше “запас прочности”, так как они выпускались по устаревшей, но достаточно надежной технологии. Любой специалист по компьютерному железу подтвердит вам, что у "старых" комплектующих качество выполнения лучше, чем у современных аналогов, хотя они значительно уступают последним в производительности. А так как производительность X-терминала не так уж и важна, то собрав его из старого компьютерного “металлолома” вы получите практически вечный компьютер. Единственное, что не рекомендуется покупать б/у это: мониторы, мышки и клавиатуры. Мониторы имеют довольно ограниченный срок службы (порядка 7 лет), а эксплуатируя эту технику достаточно долго вы подвергаете свое здоровье неоправданному риску. Насчет клавиатуры и “мышки” – это дело вкуса. Стоят они не дорого и я честно не вижу объективной причины работать на клавиатуре со стертыми (или ещё хуже наклеенными сверху) клавишами и использовать плохо ездящую "мышку" с западающими кнопками. По поводу “мыши” есть одно предостережение – если в вашем будущем X-терминале нет разъема PS/2, то скорее всего вам придется отказаться от использования оптической “мышки” (нет в продаже переходника для оптической “мышки” с PS/2 на COM и никогда не будет!).

Когда эта глава уже была закончена мне по работе пришлось приобрести в магазине б/у техники ещё один X-терминал – старый системный блок Scenic Pro D5 производства Siemens Nixdorf. В его комплект входили: процессор – Intel Pentium 166 МГц с пассивным охлаждением, память – 32 Мбайт (SIMM), материнская плата с чипсетом 430VX от Intel, интегрированный звук Creative SoundBlaster Vibra 16C, флоппи дисковод, видеоадаптер PCI Matrox Mystique 2 Мбайта и корпус типа desktop AT. Так вот, это чудо техники досталось мне всего за 35$.

Готовые терминалы: что можно купить за деньги?

В мире существует довольно много моделей готовых тонких клиентов (Thin Client). Большинство из них рассчитано на использования в сетях Windows для подключения к Citrix MetaFrame или Windows Terminal Server (Windows Terminal Services). Но есть и такие, которые ориентированы на работу в качестве X-терминалов, например, сетевые компьютеры NCD.

В качестве примера, хочу описать возможности модели сетевого компьютера NCD серии NC900. Процессор: QED RM5231 @ 165 МГц, оперативная память – от 16 до 80 Мбайт (256 Мбайт максимум), видеоадаптер - S3 Savage4 4 Мбайта, сетевая карта - 10/100BaseT Ethernet. Поддерживается 3 варианта загрузки: с удаленного сервера и с 2-х флеш-карт. Так как этот тонкий клиент использует не очень быстрый процессор, то и охлаждение в нем пассивное (нет вентиляторов) из-за чего в работе терминал не издает неприятного шума, характерного для традиционных персональных компьютеров. Такой бездисковый клиент можно использовать не только в качестве X-терминала, но и для работы с сервером Windows через протокол Citrix ICA.

На рисунке 4.4 приведена страница Интернет-магазина, которая содержит информацию по сетевым компьютерам. Как видите, обилие предлагаемых моделей впечатляет. Правда и цены явно не отечественные.

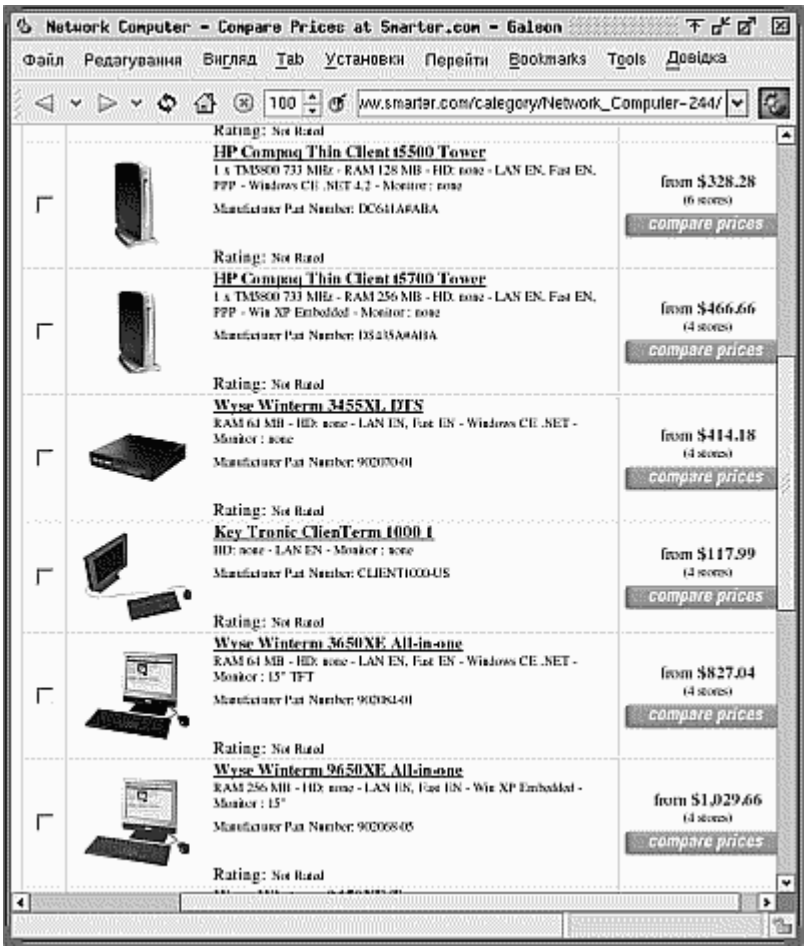

Рис. 4.4. Пример запроса по покупке сетевого компьютера через Интернет

Разнообразие предлагаемых на мировом рынке бездисковых сетевых компьютеров было вызвано настоящим бумом сетевых технологий, который можно было наблюдать в конце 90-х годов прошлого века. Тогда казалось, что эра персональных компьютеров заканчивается и что ей на смену придет эра сетевых вычислений. Но как оказалось на практике, внедрение сетевых компьютеров сопряжено с необходимостью обслуживания локальной сети и сервера, а следовательно автоматически отпала группа домашних пользователей и пользователей персональных компьютеров небольших учреждений. Играть на тонких клиентах можно разве что в "сапера" и тетрис, а следовательно игроманы также начали крутить носом. В общем сегмент рынка заинтересованный в тонких клиентах составляют большие компании, компьютеры которых объединены в локальную сеть, а также образовательные структуру. Усугубило ситуацию (распространение сетевых компьютеров) и стоимость программного обеспечения для работы сервера тонких клиентов. Если вам интересно, поинтересуйтесь сколько стоит Windows Terminal Server у компании Microsoft Corp. В общем все эти факторы не то чтобы остановили

развитие и популяризацию тонких клиентов, а скорее сделали так, что информационная отрасль как-бы забыла о их существовании. Тонкие клиенты продолжают выпускаться, но уже не с тем энтузиазмом и без намеков, что за бездисковыми компьютерами будущее.

С появлением операционной системы Linux ситуация немного изменилась в лучшую сторону. Так как эта операционная система свободно распространяется, то у конечных пользователей появилось больше возможностей овладеть технологией построения локальных сетей с использованием X-терминальных решений. А после того, как Linux “обросла” необходимым программным обеспечением ее смело можно рекомендовать в качестве настольной офисной операционной системы. К тому же появилась реальная возможность “пристроить” морально устаревшую технику, на которой точно не получится запустить современные дистрибутивы операционной системы Linux.

У нас в стране тонкие клиенты очень редко можно встретить в продаже, разве что в магазине техники б/у, продукция которого в свое время была выброшена на свалку в Германии или в другой развитой стране. Так что если вы не найдете готового X-терминала в продаже – не отчаивайтесь, так как собрать его из подручных деталей не составит особого труда.

Но возможно вы все же захотите приобрести готовый X-терминал. Тогда Вам следует обратить внимание на модели hp Compaq t5305, LTSP Term 150e и LTSP Term J-225. На момент написания книги они стоили 299,95 $, 294,95 $ и 319,95 $ соответственно. Да, цены немного кусаются, но посмотрите какие вы получаете взамен характеристики (см. Таблицы 4.4, 4.5 и 4.6):

Рис. 4.5. Внешний вид hp Compaq t5305

Таблица 4.4. Технические характеристики "тонкого клиента" hp Compaq t5305

Модель	hp Compaq t5305
Методсетевой загрузки	PXEчерез 10/100 BaseT Ethernet
Процессор	TransmetaCrusoe TM5800 733 MHz (без вентилятора, x86 совместимый)
Оперативнаяпамять	128Мбайт
Видеоадаптер	8MBATI Rage XC с максимальной разрешающей способностью 1600x1200
Дополнительныевозможности	COM-портLPT-порт4 USBверсии 1.1Аудиовыход для наушников и вход для подключения микрофона
Габаритныеразмеры (высота x ширина x глубина)	55 X205 X 195 мм (без подставки)230 X100 X 195 мм (на подставке)
Вес	1.39кг (сам блок)2.73кг (закрепляющая подставка)

Рис. 4.6. Внешний вид LTSP Term 150e

Таблица 4.5. Технические характеристики "тонкого клиента" LTSP Term 150e

Модель	LTSP Term 150e
Методсетевой загрузки	МодульEtherboot через сетевой адаптер 10/100 BaseT Realtek 8139C FastEthernet
Процессор	533MHz Via Eden (без вентилятора, x86 совместимый)
Оперативнаяпамять	128Мбайт (максимум 256 Мбайт)
Видеоадаптер	интегрированныйTrident VGA, использующий до 8 Мбайт памяти, с максимальнойразрешающей способностью 1600x1200
Дополнительныевозможности	2COM-портаLPT-порт2 USBверсии 1.1Аудиовыход для наушников и вход для подключения микрофона
Габаритныеразмеры (высота x ширина x глубина)	51 X231 X 211 мм
Вес	1.5кг (сам блок)2.6кг (закрепляющая подставка)

Рис. 4.7. Внешний вид LTSP Term J-225

Таблица 4.6. Технические характеристики "тонкого клиента" LTSP Term J-225

Модель	LTSP Term J-225
Метод сетевой загрузки	МодульPXE через сетевой адаптер 10/100 BaseT Fast Ethernet, Wake-On-Lan
Процессор	533MHz Via Eden (без вентилятора, x86 совместимый)
Оперативная память	256Мбайт
Видеоадаптер	интегрированныйTrident VGA, использующий до 8 Мбайт памяти, с максимальнойразрешающей способностью 1600x1200
Дополнительные возможности	COM-портLPT-порт4 USBверсии 2.0Аудиовыход для наушников, аудио вход и вход для подключения микрофона
Потребляемая мощность	36Ватт (максимум)
Габаритные размеры (высота x ширина x глубина)	67 X186 X 168 мм
Вес	0,85кг

Не нужно думать, что бездисковые "тонкие" клиенты – это дело прошлого. В качестве контраргумента можете ознакомится с новейшей разработкой фирмы Hewlett-Packard (официальный сайт http://www.hp.com/) - бездисковым "тонким" клиентом t5525. На веб-сранице http://h18004.www1.hp.com/products/thinclients/t5525/ представлена более подробная информация, но я думаю вас, в частности, порадует VIA Eden 800 МГц и 256 Мбайт Flash-памяти к 128 Мбайт DDR SDRAM ОЗУ.

Требования к локальной сети (проектирование сети на оптимальную загрузку)

Если у вас уже имеется локальная сеть, сейчас самое время проверить ее на соответствие требованиям сети X-терминалов. В отличие от локальной сети традиционных персональных компьютеров, где локальная сеть используется только в случае необходимости (печать на принтере, доступ к общим сетевым ресурсам, работа в сети Интернет и т.п.), в сети X-терминалов сеть эксплуатируется постоянно. Индикатор сетевой активности на сетевых картах X-терминалов никогда не выключается. Это связано с тем, что между сервером и клиентом (X-терминалом) постоянно поддерживается активное соединение, и любые проблемы передачи пакетов в сети очень негативно отражаются на работе X-терминальной среды. К тому же нагрузка на локальную компьютерную сеть значительно выше, чем в случае использование в сети традиционного персонального компьютера.

Сразу хочу предостеречь пользователей, которые используют в локальной сети коаксиальный кабель. При построении сети по принципу цепочки, ее суммарная пропускная способность ограничена 10 Мбит/с в режиме Half/Duplex. Для сети X-терминалов, такая сеть не подходит и сейчас самое время задуматься о переводе локальной сети на витую пару. Конечно, X-терминальная структура будет работать и через коаксиальный кабель, но вас вряд ли удовлетворит ее скорость и надежность (разрыв любого ее участка делит локальную сеть на два изолированных сегмента).

Рис. 4.8. Структура локальной сети с выделенным сервером и 10 рабочими станциями

Для локальной сети из витой пары, тоже не все так однозначно. Рассмотрим пример построения локальной сети на 10 рабочих станций и один

выделенный сервер (см. рис. 4.8). Как видно из рисунка, между сервером и сетевым концентратором лежит всего один сетевой кабель. Если этот кабель подключен к серверной сетевой карте со скоростью в 100 Мбит/с, то именно эта величина определяет максимальную пропускную способность локальной сети. С большей чем в 100 Мбит/с скоростью сервер не может отправлять и принимать данные из сети. Сетевые адаптеры рабочих станций не влияют на общую пропускную способность такой сети, так как скорость их работы определяет лишь комфортность работы на данном клиенте. Из схемы видно, что достаточно увеличить скорость работы сетевого адаптера сервера, для увеличения суммарной пропускной способности всей локальной компьютерной сети.

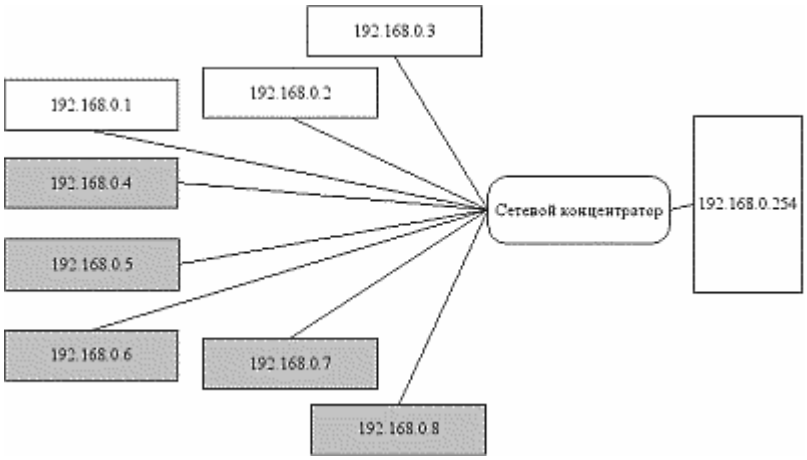

Рис. 4.9. Локальная сеть в адресном пространстве 192.168.0.0/255.255.255.0 с выделенным сервером и 10 рабочими станциями

Давайте посмотрим на тот же пример построения локальной сети, но с точки зрения IP-адресов составляющих ее компьютеров (см. рис. 4.9). Представим также, что в этой локальной сети X-терминалами стали рабочие станции с IP-адресами 192.168.0.4 – 192.168.0.8. Теперь мы имеем как бы 2 подсети с совершенно разными требованиями: традиционные персональные компьютеры и X-терминалы. Чтобы изолировать их взаимное влияние друг на друга, следует физически разделить локальную сеть на сегменты. Помогает в этом то, что для X-терминалов фактически неважен IP-адрес сетевого адаптера, так как для работающих на нем приложений используется IP-адрес сервера, следовательно для сегмента X-терминалов целесообразно ввести свою сетевую адресацию (см. рис. 4.10). Значение IP-адреса X-терминала имеет значение

только во время его загрузки, а следовательно вам не нужно будет менять параметры маршрутизации вашей локальной сети для работы X-терминалов с измененной адресацией.

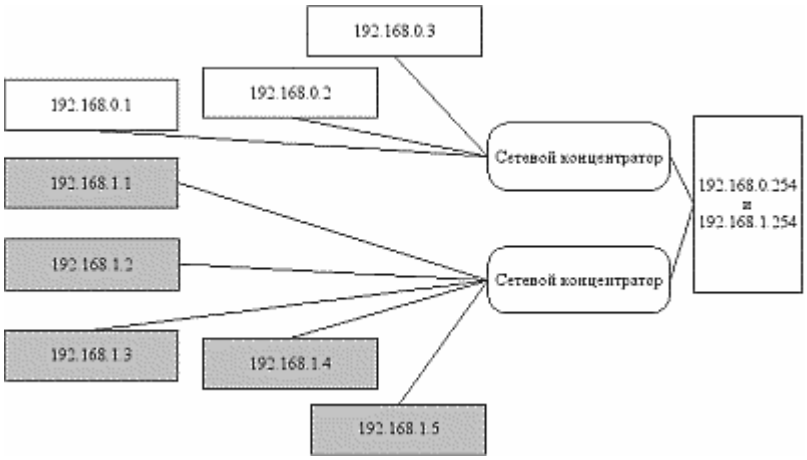

Рис. 4.10. Локальная сеть, разбитая на 2 сегмента 192.168.0.0/255.255.255.0 и 192.168.1.0/255.255.255.0

Как видно из схемы 4.10 сервер теперь имеет два сетевых интерфейса, с адресами 192.168.0.254 и 192.168.1.254. Сетевая карта с адресом 192.168.1.254 используется в сервере для работы с X-терминальными клиентами, а сетевой адаптер с адресом 192.168.0.254 – для связи сервера с остальными компьютерами локальной сети. Фактически мы увеличили скорость канала между сервером и локальной сетью в двое, только теперь нагрузка распределена поровну между двумя сетями 192.168.0.x и 192.168.1.x (каждая из сетей получила свои 100 Мбит/с). Такой экстенсивный подход (добавление сетевых интерфейсов сервера) позволяет значительно снизить сетевой трафик в X-терминальных решениях. По сути каждый X-терминал забирает свою часть сетевого трафика, который проходит через серверный сетевой интерфейс. То есть, для сети из 5-ти X-терминалов при скорости 100 Мбит/с на каждый из них придется всего по 100/5 = 20 Мбит/с. В принципе не так и мало, но что если в сети используется 20 X-терминалов (100/20 = 5 Мбит/с), а если 40 (100/40 = 2,5 Мбит/с уже совсем немного)? Конечно, такого равномерного распределения никогда не будет (исключение составляют системы с принудительным разделением трафика через маршрутизатор), каждый X-терминал будет стараться по максимуму использовать сетевой трафик, и только в минуты максимальной загрузки на долю каждого X-терминала будет приходится

пропорциональная часть суммарного сетевого трафика. Но, тем не менее, следует проектировать построение локальной сети X-терминалов с учетом пиков сетевой загрузки, и постараться их свести к минимуму. Для этих целей рекомендуется использование дополнительных сетевых интерфейсов при увеличении количества X-терминалов. Например, вы планируете создать локальную сеть на 16 X-терминалов. Предпочтительнее будет использовать 2 сетевые карты в сервере и два отдельных сетевых коммутатора для 2-х сегментов по 8 X-терминалов в каждом, чем 1 сетевой адаптер и один сетевой коммутатор (см. рис. 4.11).

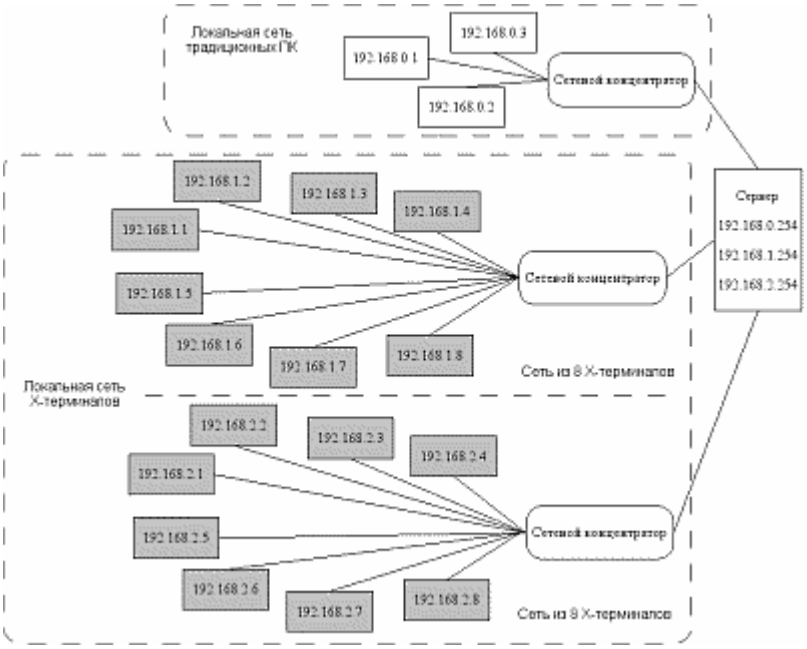

Рис. 4.11. Локальная сеть с 3-мя обычными персональными компьютерами и 16-ю X-терминалами

Определить нормативную сетевую нагрузку на X-терминал можно только экспериментальным путем, так как она очень сильно зависит от задач, которые выполняются на X-терминале (например, для офисных приложений меньше, для интенсивно использующих графику – больше). Более подробно об этом вы сможете узнать из главы "Администрирование сети X-терминалов". Рекомендуется строить локальную сеть так, чтобы доля сетевого трафика каждого X-терминала не была меньше чем 10 Мбит/с (хотя в каждом конкретном случае это число может быть больше или меньше). Исходя из такого эмпирического значения легко рассчитать пропускную способность сервера X-терминалов для любого количества "тонких" клиентов. Например,

для локальной сети с 16-ю X-терминалами, сервер должен иметь сетевой канал на уровне 16*10 = 160 Мбит/с. Как результат имеем, что для сети с 16-ю X-терминалами необходимы: 2 серверных сетевых интерфейса по 100 Мбит/с или одна сетевая карта на 1000 Мбит/с.

В случае использование гигабитных 1000BASE-T сетевых адаптеров необходимо помнить и о использовании специальных коммутаторов, рассчитанных на работу с такой скоростью, так как подключив его к обычному 10/100 Мбит/с он будет работать на скорости всего в 100 Мбит/с.

Интересно проанализировать направление движения сетевого трафика между X-терминалами и сервером. Как показывает статистика сетевой активности, в X-терминальных решениях преобладает трафик от сервера к X-терминалу в соотношении примерно 1 к 14, то есть на один пакет от X-терминала к серверу, приходиться 14 пакетов от сервера к X-терминалу (хотя это соответствие очень зависит от задач, что выполняются на X-терминале). Поэтому когда будете изучать загруженность вашей X-терминальной сети особое внимание следует обратить на размер исходящего трафика от сервера, и входящего для каждого клиента X-терминала.

Глава 5. Выбор операционной системы для сервера

Ключевые моменты выбора операционной системы

Для X-терминал сервера подойдет практически любой дистрибутив Linux, так как работа X-терминалов только частично зависит от серверной операционной системы и в основном использует возможности X-Windows. Тем не менее определенные нюансы эксплуатации имеются и хорошо быть с ними знакомым еще на этапе выбора операционной системы для сервера.

Во-первых, при выборе дистрибутива для сервера X-терминалов следует обратить особое внимание на стабильность его работы. Стабильность – это главная характеристика, которой должна обладать выбранная вами сборка операционной системы Linux. Важность стабильной работы для сервера X-терминалов заключается в том, что на нем одновременно работают сразу несколько пользователей, и проблемы в работе серверной операционной системы коснутся также одновременно всех их. Те же критерии стабильной

работы следует применять и к программному обеспечению сервера, а конкретно к тем программам, которые будут использовать в своей работе пользователи. Так, программа нерационально использующая процессорное время и оперативную память сервера, повлияет на удобство работы всех пользователей X-терминалов, даже тех, которые ее никогда не используют.

Во-вторых, для серверного программного обеспечения важна скорость, так как опять же сервер предназначен для коллективного использования и чем быстрее работают программы на нем, тем лучше его общая производительность и, соответственно, тем качественнее получается программная среда каждого отдельного пользователя. В этом плане у операционных систем Linux есть значительное преимущество, так как исходный код составляющих их приложений доступен и всегда можно выполнить компиляцию пакетов с оптимизацией под архитектуру центрального процессора сервера.

Никогда не следует забывать о безопасности работы сервера. Вопрос обеспечения безопасности особенно остро стоит там, где хранящаяся на сервере информация представляет собой важные данные и имеет реальную стоимость.

Парадоксом использования X-терминалов является то, что сервер должен быть надежным программно-аппаратным комплексом, но работать на нем должно не серверное, а пользовательское программное обеспечение. Если вы привыкли к классическому понятию UNIX-сервера, на который работают очень тщательно проверенные программные продукты, выполняющие исконно серверные функции: веб-сервера, FTP-сервера, прокси-сервера, DNS-сервера и т.д., то для сервера X-терминаллв это не характерно. Согласен, что часть серверных программ на нем также будет функционировать, но вместе с этим там будут и такие программные продукты, как текстовый редактор, электронная таблица, веб-браузер, файловый менеджер и т.п. Схематически этот симбиоз сервера и клиентского компьютера можно увидеть на рис. 5.1.

Рис. 5.1. Работа серверного и клиентского программного обеспечения на X-терминал сервере

Кое-кто может увидеть в этом большую проблему безопасности работы X-терминал сервера. Однако, не следует забывать, что "представляющее угрозу" программное обеспечение запускается от имени пользователя, а, стало быть, причинить вред оно может только ему самому, а не всей системе в целом.

Такая гибридная природа X-терминал сервера накладывает определенные ограничения на выбор истинно серверных дистрибутивов операционной системы Linux. Не следует забывать, что кроме надежной серверной части, X-терминал серверу нужен и достаточно широкий набор пользовательского программного обеспечения, которое, как правило, не включается в серверные дистрибутивы.

Отдельный вопрос в выборе серверной операционной системы относится и к поддержке работы в графическом режиме, а также наличие в дистрибутиве пакетов X Window System. Как вы уже знаете, X-терминал сервер должен работать в графическом режиме, а, стало быть, без X-Windows ему никак не обойтись. Различные дистрибутивы Linux используют индивидуальные методики инициализации системы, а поэтому уровни выполнения, которые соответствуют графическому режиму, у них также различаются. Так, для систем исторически базирующихся на RedHat, графический уровень

выполнения соответствует числу 5, а, например, для операционной системы Debian GNU/Linux, это число 3. Без перевода X-терминал сервера в графический режим, X-терминалы просто не смогут подключиться к менеджеру дисплея, и ни о какой дистанционной работе не сможет быть и речи.

Очень важна версия X-Windows, которая будет установлена на X-терминал сервере. Желательно, чтобы это была довольно новая сборка и поддерживала все современное оборудование. В последнее время, наметилась тенденция, когда практически все дистрибутивы операционной системы Linux переключились на использование версии от X.org, хотя еще несколько лет назад использовали пакеты от XFree86. Будем считать, что все, что происходит - к лучшему, и постепенно в распоряжении пользователей операционной системы Linux окажется самая надежная и технически совершенная графическая подсистема.

Перед тем, как перейти к обзору доступных дистрибутивов операционной системы Linux, следует напомнить один простой факт. Наиболее подходящей операционной системой является та, которую хорошо знает системный администратор. С этим чисто человеческим фактором следует всегда считаться, так как без надлежащего администрирования любая, пусть даже очень хорошая операционная система, обречена на печальный финал. Например, если вы прекрасно разбираетесь во FreeBSD, то после определенных усилий по настройке сервера, вам удастся запустить систему X-терминалов, в которой X-терминал сервер будет обслуживать именно ваша любимая операционная система, а не Linux, как это предполагалось заранее. Мне сложно анализировать плюсы и минусы такого подхода, а еще сложнее предсказать как будет работать такая система, но администрировать вам ее точно будет легче.

Как известно, системные администраторы такие же люди как и мы с вами (тут конечно я немного перебрал), и им свойственно эмоциональное отношение к обслуживаемым операционным системам. На практике это выливается в то, что одна система им нравится, а другая – нет. Чем лучше отношение к операционной системе со стороны системного администратора, тем лучше он ее знает и хочет еще глубже в ней разобраться. В подтверждение моих слов,

посетите любой форум, посвященный использованию операционных систем, и задайте вопрос: "Что лучше: Linux или Windows?". Если на этом форуме достаточно либеральный модератор, то в ответ вы увидите настоящую бурю высказываний приверженцев как одной, так и другой операционной системы. А если поставить более специализированный вопрос: "Какой из дистрибутивов Linux лучше?", то тут я вообще вам не завидую, так как это будет настоящий шквал эмоций в духе "RedHat forever, сам дурак!". Этим примером я еще раз хочу подчеркнуть субъективность взглядов на используемое программное обеспечение со стороны пользователей. Разбираясь в новой операционной системе, проводя долгие часы в чтении документации, и объясняя конечным пользователям как работать с тем или иным программным продуктом, системный администратор поневоле начинает воспринимать накопленный опыт как свое интеллектуальное достояние. А подумайте сами, кому хочется добровольно расставаться со своей собственностью, пусть даже интеллектуальной? В любом случае при знакомстве с операционной системой Linux за правильно нужно использовать пословицу, которая говорит, что знания за плечами не носят. Поэтому настройтесь на новые идеи и позитивно воспринимайте любую информацию о развитии этой операционной системы. К тому же, все дистрибутивы Linux используют одно ядро (с небольшими изменения) и очень схожий набор прикладных программ, поэтому реальных отличий не так и много, а там где они имеют место, их нужно воспринимать как разнообразие и возможность выбора, которые всегда были присущи сообществу свободно распространяемого программного обеспечения.

Ключевые моменты выбора операционной системы (продолжение)

Итак, перейдем непосредственно к обзору дистрибутивов операционной системы Linux, доступных на сегодняшний момент, в разрезе совместимости с X-терминалами. Конечно, рассмотреть все дистрибутивы операционной системы Linux физически невозможно, поэтому для общей информации можете посетить также страницу сайта http://lwn.net/Distributions/, чтобы ознакомиться с более полным списком. Принимая во внимание тот факт, что дистрибутивы

Linux создаются компаниями или группами лиц, целесообразно группировать их по этому признаку.

ASPLinux

В своей работе я использую дистрибутив от российского разработчика операционной системы Linux – компании ASPLinux (сайт http://www.asplinux.ru). Так, X-терминал сервер, который я администрирую, работает под управлением ASPLinux 9 Express, и я вполне удовлетворен его производительностью. Несмотря на то, что этот дистрибутив далеко не новый и состоит всего из 3-х компакт дисков, он включает в себя операционную систему и довольно обширный набор прикладных программ. Если вы имеете достаточный опыт работы в операционной системе Linux и вам не нужна дополнительная документация, то также можете остановить свой выбор на этом облегченном дистрибутиве (сборке Express). Кроме этого, компания ASPLinux предлагает своим пользователям дистрибутивы в сборке Standard, Deluxe и Greenhorn. На период написания книги, доступными были версии 9.2 и 10 операционной системы ASPLinux. Так, редакция Standard кроме самой операционной системы и базового набора прикладных программ содержит также руководство по установке и руководство пользователя, а также дополнительные программы (включая коммерческие демонстрационные версии) и игры. Наиболее полной считается редакция Deluxe, в которая отличается от Standard наличием исходных текстов пакетов операционной системы, компакт диска с документацией, книги "Руководство администратора", а также коммерческого программного продукта Acronis OS Selector. Для знакомства с операционной системой ASPLinux компания предлагает также загрузочный диск Greenhorn, который позволяет работать с операционной системой даже не устанавливая ее на компьютер. Редакция Greenhorn является незаменимым помощником при выполнении восстановительных работ для любого системного администратора.

Для X-терминал сервера интерес может представлять дистрибутив ASPLinux Server II, оптимизированный для создания корпоративных серверов различных классов и поддерживающий различные серверные архитектуры,

включая многопроцессорные, и системы с большими объемами памяти. Построенный на ядре операционной системы Linux версии .4.21, он включает в себя много функций ядра версии 2.6, таких как поддержка NPTL, IPSec, асинхронный ввод-вывод, балансировщик загрузки сети IPvs, списки управления доступом (ACL), поддержка файловой системы XFS и некоторых других. Кроме этого, в состав данного дистрибутива входит программный пакет Acronis True Image Server for Linux, который позволяет выполнять полное резервное копирование и восстановление данных для Linux-сервера. А так как на X-терминал сервере используется схема централизованного хранения информации всех пользователей X-терминалов, то это немаловажный плюс. В редакцию ASPLinux Server II также попала поддержка электронных ключей HASP — аппаратной системы для защиты данных от несанкционированного использования.

Дистрибутивы от ASPLinux интересны для конечных пользователей еще и тем, что в команде их разработчиков участвует коллектив Black Cat Linux Team: разработчики наиболее популярного дистрибутива Linux на постсоветском пространстве. Так, 19 марта 2001 года компания ASPLinux, разработчик российского дистрибутива GNU/Linux, и Black Cat Linux Team объявили об объединении. По условиям соглашения все права на товарный знак Black Cat Linux перешли к компании ASPLinux. Объединенная команда сосредоточила усилия на выпуске дистрибутива ASPLinux, в котором соединились лучшие черты двух дистрибутивов. Леонид Кантер, лидер Black Cat Linux Team, возглавил команду разработчиков дистрибутива. Компания ASPLinux продолжила поддержку многочисленных пользователей дистрибутива Black Cat Linux.

Продукты под торговой маркой ASPLinux появляются на рынке с завидным постоянством. Также регулярно выпускаются и обновленные версии пакетов, которые входят в состав этих дистрибутивов. Используя средство автоматического обновления yum, можно поддерживать систему в оптимальном состоянии и использовать самое современное программное обеспечение. Практически все из предлагаемых дистрибутивов ASPLinux,

кроме Greenhorn подойдут в качестве серверной платформы для организации сети бездисковых X-терминалов. Преимуществом ASPLinux для организации X-терминальной среды является еще и тот факт, что прикладное программное обеспечение этого дистрибутива, как правило, собрано для работы в архитектуре i386, а следовательно, его можно использовать для локального запуска даже на морально устаревших X-терминалах.

Локализация операционной системы Linux является особенно сильной стороной дистрибутива ASPLinux. Особенно хорошо реализована поддержка русского языка, причем как для кодовой страницы KOI8-R, так и для CP1251. На родной язык переведена не только программа установки, но также большая часть прикладного программного обеспечения и даже главные разделы справочной документации man. Украиноязычным пользователям понравится возможность работы в украинском варианте данной операционной системы (кодовые страницы KOI8-U или CP1251). Хороший уровень поддержки национальных языков должен прийтись по душе рядовым пользователям, которые смогут более эффективно использовать программное обеспечение в своей работе.

RedHat Linux

RedHat Linux – это наверное самая популярная версия операционной системы Linux. Данный дистрибутив разрабатывается одноименной компанией Red Hat (сайт http://www.redhat.com), основанной еще в 1993 году. Базируясь на платформе свободно распространяемого программного обеспечения, компания Red Hat смогла предложить своим пользователям хорошую операционную систему и неплохое сопровождение. В одно время можно было даже перефразировать известное словосочетание: говорим Linux, думаем Red Hat. Имея лицензионное соглашение GNU/GPL данный программный продукт нашел множество почитателей во всем мире. Но такое свободное распространение было вплоть до выхода операционной системы RedHat Linux версии 9, а теперь уже осталось в прошлом. Сейчас компания практикует исключительно платное предоставление своей операционной системы, используя модель подписки. То есть оплачивая подписку на сервисное

обслуживание, клиент получает возможность скачать программное обеспечение с сайтов компании или получить его на сменных носителях (CD-R, DVD-R). Возможно уверовав в свое действительно лидирующее положение на рынке операционных систем Linux, компания прекратила предоставление бесплатного доступа к создаваемому ею программному обеспечению. Причины, побудившие компанию к такому шагу, вполне очевидны. С одной стороны, программный продукт пользуется действительно большой популярностью, а, с другой стороны, затраты на предоставление технической поддержки пользователям постоянно возрастают. Как видите, выбор довольно не велик: или ухудшить качество технической поддержки, а как следствие, и самого дистрибутива, или сосредоточиться на поддержке пользователей и разработке, но повысить стоимость платы за использование своего программного продукта. Как известно, компания Red Hat выбрала второе направление развития. Возможно, сам продукт от такого решения и выиграл, но часть пользователей предыдущих версий RedHat Linux были огорчены таким решением. Выход из сложившейся ситуации не заставил себя долго ждать. На базе сообщество разработчиков и пользователей RedHat Linux появился перспективный проект Fedora Core, который развивается отчасти благодаря спонсорской поддержке компании Red Hat. На момент написания книги, последней версией программного продукта компании Red Hat был Red Hat Enterprise Linux v.4, выпущенный для всеобщего обозрения в феврале 2005. Линейка операционный систем Red Hat Enterprise Linux предназначена, как для высокопроизводительных серверов, так и для обычных рабочих станций. Цена подписки, мягко говоря кусается. Так, Red Hat Enterprise Linux WS в начальной комплектации (Basic Edition) стоит 179 $, вот вам и свободно распространяемое программное обеспечение. Правда, следует отметить, что дистрибутиву от Red Hat присущи также и положительные черты. В частности, он будет хорошим выбором для работы на загруженном сервере. Следует также помнить, что часть программного обеспечения может устанавливаться только на сертифицированные операционные системы Linux, одной из которых как раз и является Red Hat Enterprise Linux.

Fedora Core

Fedora Core Linux (сайт http://fedora.redhat.com/) - это наследник культового дистрибутива RedHat Linux. Он создан сообществом Fedora Project при участии и спонсорской поддержке компании Red Hat. Дистрибутив Fedora Core одинаково хорошо подходит для установки, как на домашний или офисный компьютер, так и на мощный сервер. Fedora Core Linux включает в себя удобную программу для установки и полный набор приложений - игры, офисные пакеты, серверные приложения и т.д. Последняя версия данного дистрибутива (на период написания книги) Fedora Core 3 включает в себя технологию SELinux, и продолжает политику использования только самого нового программного обеспечения. В третьей версии данного дистрибутива статический каталог /dev был заменен на динамический, контролируемый системой udev. Fedora Core 3 может работать на 2-х принципиально различных аппаратных платформах: 32-х разрядных Intel x86 и 64-х разрядных AMD64/EM64T. Дистрибутив от сообщества Fedora Project версии 3 для архитектуры x86 оптимизирован под использование в качестве центрального процессором Intel Pentium 4, и это обстоятельство нужно учитывать при выборе аппаратного обеспечения под данную сборку. Конечно, работать Fedora Core 3 будет и на процессорах других производителей, включая ADM и VIA, но менее эффективно. Сообщество разработчиков Fedora Core трудится очень продуктивно, что позволяет выпускать им новые версии своего продукта через небольшие промежутки времени. Такая поспешность радует пользователей-новаторов, но также делает данный дистрибутив операционной системы Linux менее надежным. При выборе Fedora Core в качестве серверной платформы для X-терминал сервера, системному администратору следует особенно тщательно проверить прикладные программы, которые будут использовать пользователи на предмет устойчивой и надежной работы.

Debian GNU/Linux

Про дистрибутив Debian говорят, что это сердце сообщества операционных систем Linux. Создатели данного программного продукта действительно хорошо постарались, чтобы сделать его по настоящему свободным (отсюда и

приставка GNU в названии). С технической точки зрения, Debian - продвинутый дистрибутив, и его вряд ли можно порекомендовать новичку. Debian GNU/Linux значительно отличается от дистрибутивов, основанных на Red Hat, и менеджером пакетов, и уровнями выполнения и даже содержимым и расположением конфигурационных файлов. Это по-настоящему "хакерская" операционная система в хорошем понимании данного слова. До недавнего времени разработка этого дистрибутива велась в двух направлениях: с одной стороны, вносились изменения в существующее программное обеспечение и тем самым оно попадало в раздел пакетов разработки (unstable distribution), а с другой стороны, существующие пакеты тщательно тестировались на предмет ошибок и после этого попадали в раздел стабильных пакетов (stable distribution). Такая ситуация привела к тому, что выход очередной версии данной операционной системы вызывал у некоторых пользователей недовольство стабильными, но часто морально устаревшими программными пакетами, которые входили в ее состав. Для разрешения подобной ситуации была введен еще один промежуточный раздел тестируемых пакетов (testing distribution), которые значительно надежнее, чем пакеты в разработке, но все же не так тщательно проверены как стабильные версии. Debian по праву может считаться самым большим из дистрибутивов операционной системы Linux. Так, в версии Debian GNU/Linux 3.0 содержится почти 9000 пакетов.

Для серверной платформы X-терминалов данный дистрибутив конечно подойдет, но перед его использованием желательно убедиться в том, что системный администратор обладает достаточными знаниями и желанием работать с Debian. Так как не следует забывать, что у данной операционной системы есть как преданные поклонники, так и яростные противники.

SuSE Linux

SuSE Linux – это наиболее продаваемый в Европе дистрибутив операционной системы Linux (сайт http://www.suse.com), благодаря, наверное, продуманному набору прикладных программ (включая даже коммерческие) и хорошей сбалансированной настройке. После покупки его всемирно известной компанией Novell, увидел свет дистрибутив SuSE Linux 9.2 Professional.

Субъективно, SuSE Linux – это не серверный дистрибутив, а скорее всего операционная система для рабочей станции, хотя и для работы на сервере он тоже прекрасно подойдет. Отличительными особенностями SuSE Linux является пожалуй наличие единой утилиты конфигурирования системы YaST, которая значительно облегчает администрирование данной операционной системы. По мнению пользователей, SuSE – это наиболее отшлифованный дистрибутив, который можно начать использовать даже без достаточного опыта. Несмотря на то, что это коммерческий дистрибутив (версия SuSE Linux 9.2 Professional продается по цене 89,95 $), он нашел достаточно поклонников по всему миру, которые по достоинству оценили его положительные качества. В данной версии операционной системы Linux очень значительное внимание уделяется локализации на немецкий, французский, итальянский и испанский языки.

Mandrakelinux

Созданный еще в 1998 году Mandrakelinux (сайт http://www.mandrakelinux.com/) поставил себе целью сделать Linux простым в использовании для всех. К тому времени операционная система Linux была уже хорошо известна как мощная и стабильная операционная система, которая требовала, к великому сожалению начинающих пользователей, серьезных технических знаний и активного использования "командной строки". Создатели Mandrake увидели возможность совместить ее с наилучшими графическими окружениями рабочих столов, добавили свои собственные графические утилиты настройки и быстро сделали Mandrakelinux известеным своими стандартами в простоте использования и функциональности. Такой творческий подход позволяет Mandrake предложить всю мощь и стабильность Linux совместно с простотой использования и приятным окружением как для профессионалов так и для индивидуальных пользователей. На практике, конечно? не все так гладко. Поначалу Mandrakelinux очень напоминал RedHat Linux, но постепенно все больше и больше удалялся от такого наследия. Изначально акцент был сделан на простоту использования и аппаратную оптимизацию под процессоры Intel Pentium и совместимые с ними. Тем самым

пользователи более слабых компьютеров не могут использовать в своей работе эту операционную систему. Ориентация на потребности рядовых пользователей положительно сказалась на популяризации данного дистрибутива, который нашел множество почитателей во всем мире. В 2005 году компанией Mandrakesoft была куплена известная бразильская компания Conectiva, создатель одноименного дистрибутива Linux (сайт http://www.conectiva.com/) популярного в Латинской Америке. Будем надеяться, что от такого слияния выиграет все сообщество пользователей Linux.

AltLinux

Компания ООО "Альт Линукс" это еще один российский разработчик одноименной операционной системы. Дистрибутивы AltLinux исторически базируются на Mandrake Linux, но компания привнесла в них много нового, особенно большая работа проделана по локализации (русификации) программного обеспечения. На момент написания данной книги последней версией данного дистрибутива был ALT Linux 2.4 Master - универсальный дистрибутив GNU/Linux, включающий в себя множество подготовленных к эксплуатации программных решений для серверов и рабочих станций. К особенностям данного дистрибутива можно отнести повышенные методы обеспечения безопасности, которые обеспечиваются авторизацией пользователей по системе TCB (альтернативной схеме управления теневыми паролями), а также выполнением большинства серверов-демонов в виртуальной среде (chrooted environments). ALT Linux 2.4 Master поддерживает несколько вариантов ядер версий 2.4.x и 2.6.x: для серверов, рабочих станций и т.д. Несмотря на то что это дистрибутив от российского производителя, он имеет широкую аудиторию пользователей на территории России и стран СНГ.

Gentoo

Данный дистрибутив Linux очень сильно отличается от своих собратьев по лагерю Open Source. Gentoo Linux (сайт http://www.gentoo.org/) - представляет собой систему, в которой, с одной стороны, упрощена процедура настройки, установки и обновления программных пакетов, а с другой стороны, все они непосредственно компилируются на компьютере пользователя. Благодаря такой

оптимизации пользователю очень легко собрать не перегруженную излишними сервисами операционную систему при этом максимально оптимизированную под конкретное аппаратное обеспечение. Для тех, кто не хочет компилировать программы, можно получить их уже в собранном виде, но, как мне кажется, это искажает философию Gentoo Linux. Все программное обеспечение данной операционной системы хранится в Portage – системе управления пакетами программ для Gentoo. Имея доступ в Интернет, вы всегда можете обновить локальное дерево Portage пакетами самых последних версии. На сегодняшний день репозитарий Portage насчитывает свыше 8 тысяч пакетов, которые постоянно обновляются. Пользователи любят Gentoo Linux за то, что это пожалуй самый быстрый дистрибутив данной операционной системы. Правда, к его недостаткам следует причислить недостаточную стабильность, для того чтобы порекомендовать его для использования в качестве операционной системы Linux для сервера.

Slackware

Пожалуй, Slackware – это самый юниксоподобный дистрибутив операционной системы Linux. В нем используется система инициализации BSD, благодаря которой проще происходит запуск сервисов при включении компьютера. С другой стороны, пользователи лишены возможности настройки системы в различных режимах, например, однопользовательском, многопользовательском с поддержкой работы в сети, графическом режиме и т.д. Несмотря на наличие утилиты для управления пакетами, дистрибутив Slackware до сих пор работает с программами, которые распространяются в обычных сжатых тарболах (*.tgz). С технической точки зрения Slackware всегда был продвинутым дистрибутивом и сохраняя преданность идеалам UNIX, отличается простотой в использовании и надежностью. Его вполне успешно можно использовать в качестве X-терминал сервера, правда, начинающим пользователям его рекомендовать нельзя.

Linux XP

Авторы дистрибутива операционной системы Linux XP Professional Edition (сайт http://www.linux-xp.com/) заверяют, что это универсальная и безопасная

операционная система для русскоязычных домашних и бизнес-пользователей, основанная на открытых программных решениях мирового лидера в области Linux - корпорации Red Hat, Inc. На самом деле это еще один клон популярного проекта Fedora Core. Созвучное название с операционной системой от Microsoft конечно добавляет популярности данной сборке, особенно у начинающих пользователей, но также отталкивает профессионалов Linux. Пользователям операционных систем семейства Linux, я думаю, незачем копировать методы работы других операционных систем, так как именно в многообразии сила этого свободно распространяемого программного продукта. С другой стороны, доступность программного обеспечения сообщества Open Source позволяет любому человеку создавать свой собственный дистрибутив и вносить в него уникальную функциональность, что еще раз продемонстрировали создатели Linux XP.

Linspire

Последняя, на момент написания книги, версия данного дистрибутива операционной системы Linux называется Linspire 4.5 (сайт http://www.linspire.com/). Этот дистрибутив ранее известный как Lindows, рассчитан на неопытных или начинающих пользователей системы Linux. Лично я не рекомендовал бы его использовать в качестве основы для X-терминал сервера, хотя его создатели уверяют, что данный программный продукт является полнофункциональной операционной системой такой как Microsoft Windows XP или Apple Mac OS X. Linspire имеет встроенную поддержку упрощенного метода установки программного обеспечения, называемую CNR Service, за которую необходимо платить 49,95 $ в год. Данный метод установки и обновления программного обеспечения больше подходит для домашних пользователей, а для сервера, наверное, лучше использовать другие технологии.

myLinux

Для жителей Украины, интерес может представлять проект украиноязычного дистрибутива операционной системы Linux от компании Майлинукс – myLinux (сайт http://mylinux.com.ua/). Так, доступная на момент написания книги версия myLinux 3.1 включала в себя: ядро операционной

системы версии 2.6.9, графическую подсистему X.org X11 6.8.1, графическую среду Gnome 2.8.0, подсистему печати CUPS 1.1.22, офисный пакет OpenOffice.org 1.1.2, браузеры Mozilla 1.7.3, Firefox 1.0, почтовые клиенты Mozilla Mail 1.7.3, Evolution 2.0.3 и т.д.

Как видите, многообразие выборы операционных систем Linux (дистрибутивов) действительно есть и в зависимости от ваших потребностей вы вольны использовать ту систему, которая вам лучше подходит. Главное при этом не забывать о 3-х обязательных составляющих серверного дистрибутива операционной системы Linux для X-терминалов: надежность, безопасность и управляемость. Конечно, можно не придерживаться этих рекомендаций и настроить X-терминал сервер на любом дистрибутиве, но всегда нужно помнить, что главное условие успешности сервера не простота его настройки, а простота и надежность в его работе и обслуживании.

Планирование эксплуатационного обслуживания сервера

Если говорить начистоту, то в обслуживании работающего X-терминал сервера нет ничего особенного. Более того, если исключить проблемы, связанные с выходом из строя оборудования и сетевой безопасности, то эксплуатационное обслуживание вообще выполнять проще простого. К сожалению, в реальных системах это далеко не так, и планирование, а также выполнение резервного копирования – это первостепенная задача системного администратора, особенно в X-терминальной среде. Помню, в одной книге мне встретилось определение служебных обязанностей системного администратора: поддержание информационной системы в работоспособном состоянии. Так вот, для X-терминальной среды этот постулат необходимо несколько подкорректировать. Системный администратор сервера X-терминалов должен поддерживать систему в работоспособном состоянии и выполнять регулярное резервное копирование.

Важно отличить резервное копирование, например, от работы дискового массива RAID в режиме зеркалирования. Точная копия жесткого диска не заменит резервной копии, так как из архива мы можем восстановить даже сознательно удаленные или перезаписанные файлы.

Какие данные архивировать, а какие нет, и как часто выполнять операции по резервному копированию? На все эти вопросы вы должны самостоятельно найти ответы, исходя из потребностей именно вашей X-терминальной системы. Целесообразно архивировать конфигурационные файлы (каталог /etc), файлы журналов, а также данные запущенных серверов-демонов (подкаталоги /var), домашние каталоги пользователей (подкаталоги /home) и директории для совместного доступа к файлам. Очень редко нуждаются в архивировании подкаталоги /usr, так как хранящиеся там файлы практически не подвержены изменениям.

Критериями частоты выполнения резервного копирования должны быть подверженность файлов изменениям и важность содержащейся в них информации. Так как чем важнее файл с данными, тем чаще нужно сохранять его копию в архиве.

Чтобы осознать всю важность выполнения резервного копирования, а также более отчетливо разработать план своих действий, вы должны смириться с мыслью о выходе из строя оборудования вашего X-терминал сервера. Представьте себе, что в один прекрасный момент вы обнаружите, что ваш сервер перестал функционировать, пользователи остались без автоматизированных рабочих мест и вся работа в организации парализована. Какие действия предпринять, чтобы восстановить работоспособность? Хорошо, если это просто замена одного из комплектующих (центральный процессор, материнская плата, сетевой адаптер и т.д.). А если из строя выйдет винчестер? Работающий RAID массив решит эту проблему, но помните, что даже RAID массивы не гарантируют 100% сохранности ваших данных. Единственным решением в подобной ситуации может служить установка операционной системы на новый жесткий диск с восстановлением пользовательской информации из резервных копий. Чем более новой окажется резервная копия, тем лучше для вас и ваших пользователей.

Не следует забывать также о таком немаловажном факторе, как время. От скорости восстановления работоспособности информационной структуры организации зависит в конечном итоге минимизация потерь от форс-мажорных

обстоятельств выхода из строя оборудования, а также судьба дальнейшего трудоустройства системного администратора, который будет выполнять восстановительные работы. Так как если затянуть этот процесс, то можно незаметно для себя оказаться в числе потенциальных кандидатов биржи труда.

Целостности данных на X-терминал сервере угрожают также вредоносные программы, такие как вирусы. Благо, под операционную систему Linux их совсем не много и их появление не носит такого эпидемиологического характера, как в случае с операционными системами от компании Microsoft. Однако это не дает системному администратору никакого права расслабляться, и прекращать наблюдение за работой локальной сети. В некоторых случаях потери от злонамеренного поведения пользователей могут оказаться даже выше, чем физическая поломка оборудования, поэтому такой вариант развития также следует предусматривать.

Для операционной системы Linux написано достаточно много универсальных средств, выполняющих резервное копирование, как свободно доступных так и коммерческих (платных). В принципе, это и не удивительно, так как эта серверная операционная система часто работает в круглосуточном режиме. Под управлениям Linux выполняются задачи, содержащие очень важные данные (веб-сервера, файловые сервера, почтовые базы и т.д.), потерю которых следует предотвратить любой ценой. К сожалению, детальное описание принципов работы, а также примеры использования программного обеспечения для резервного копирования под Linux выходят за рамки данной книги. Чтобы больше узнать о программных продуктах, которые используются для резервного копирования, рекомендую вам посетить сайт http://www.backupcentral.com/, на котором собрано очень много информации по данному вопросу. Там вы найдете описания как коммерческих так и свободно распространяемых программных продуктов.

Информационная отрасль развивается быстрыми темпами, и операционная система Linux не исключение. Установив на сервер X-терминалов операционную систему, нужно быть готовым к выполнению периодического обновления входящих в ее состав пакетов. Практически все дистрибутивы

операционной системы Linux предлагают специальные системные утилиты для выполнения работ по установке и обновлению программного обеспечения. Наверное, дальше всего в этом плане пошли создатели Gentoo Linux. Так, пользователям этой операционной системы для того чтобы обновить все программное обеспечение на компьютере до самой новой версии, достаточно ввести одну команду:

emerge -u world

Конечно, необходимо убедится при этом, что компьютер, на котором выполняется обновление, подключен к глобальной сети Интернет. Правда, и другие менеджеры пакетов не отстают в этом плане от Gentoo. Например, для популярного в дистрибутивах основанных на Red Hat пакетного менеджера yum, подобная команда будет такой:

yum update

Как это ни странно, но главные отличия между дистрибутивами операционной системы Linux скрываются именно в формате используемых ею пакетов. В мире UNIX программное обеспечение традиционно устанавливается из тарболов (архивных файлов, содержащих исходный код или предварительно скомпилированных), имеющих расширения *.tar.gz, *.tgz, *.tar.bz2 и *.tbz. Но использование такого метода установки затрудняет контроль за программным обеспечением, и особенно усложняет его обновление. Поэтому создатели дистрибутивов Linux решили распространять программы в пакетах - специальных стандартизированных форматах. При установке программного обеспечения из тарболов, как правило, выполняются такие действия: содержимое извлекается из архива, пользователь знакомится с инструкциями файла Readme, а затем выполняет установочный сценарий или компиляцию программного продукта. В отличие от тарболов, файлам пакетов присущи более широкие свойства. Например, к характеристикам пакетов относятся зависимости (какие дополнительные пакеты нужны для успешной установки данного пакета), сценарии, выполняющиеся при инициализации пакета, обновлении его до новой версии и удаления с компьютера, стандартизированные файлы описания содержимого пакета и т.д. Для работы с

пакетами в дистрибутивах Linux предусмотрены специальные системные утилиты. Так, для получивших наибольшее распространение, благодаря все той же Red Hat, пакетов в формате *.rpm предназначена одноименная утилита rpm (Red Hat Packages Manager). Пакеты для Debian имеют расширение *.deb и управляются при помощи утилиты dbpkg. Не все дистрибутивы операционной системы Linux перешли на использование пакетов. Некоторые (Rocklinux, Gentoo, CRUX, Archlinux, Sorcerer, SourceMage, Lunar и т.д) до сих пор продолжают использовать тарболы, при этом, как правило, каждый дистрибутив использует свои правила управления программным обеспечением и методы описания и контроля зависимостей.

Хотя использовать утилиты для управления пакетами такие как rpm или dbpkg совсем не сложно, они не совсем подходят для регулярного обновления программного обеспечения. Особенно сложно их использовать при установке пакетов с большой и запутанной системой зависимостей. Для облегчения жизни системных администраторов были придуманы надстройки над системами управления пакетами, своего рода мета-менеджеры пакетов. Из программ подобного типа наибольшей популярности заслуживает, конечно, утилита apt, которая, несмотря на свое "бразильское" происхождение, стала известна широкой общественности линуксоидов из-за использования в дистрибутиве Debian. В дистрибутивах, основанных на RedHat Linux, для подобных целей используется утилита yum, хотя есть примеры успешного использования в них и утилиты apt.

При обновлении пакетов особенно осторожно следует обновлять прикладное программное обеспечение. Именно прикладное, так как его в своей работе используют рядовые пользователи, причем часто даже непредвиденным для создателей образом. Согласитесь, что процесс разработки программного обеспечения идет по довольно простой схеме. Разработчик вносит в свой программный продукт новые функции, тестирует их и предлагает аудитории своих пользователей. Потом учитываются желания пользователей, исправляются ошибки и выходит так называемая стабильная версия программы. Но в мире Linux нет четкого и единого понятия, такого как

операционная система Linux, поэтому составители дистрибутивов адаптируют программный продукт под стандарты своей сборки. В итоге часто можем наблюдать картину, что в новой версии программного продукта нет функций, которые были доступны в старой, зато появились новые ошибки, попросту незамеченные авторами программного пакета. Авторов дистрибутивов операционной системы Linux можно понять – они специалисты узкого профиля и им сложно вникнуть во все детали использования прикладного программного обеспечения. Поэтому во избежание подобных проблем, я рекомендую системному администратору устанавливать новое программное обеспечение сначала на тестовый компьютер, где сравнивать его функциональные возможности и стабильность в работе с предыдущей версией, и если новый релиз успешно пройдет проверку, то только тогда устанавливать его на X-терминал сервер.

При обновлении программного обеспечения важно помнить, что вы работаете в многозадачной среде и нельзя начинать процедуру обновления когда вам вздумается. Если другие пользователи X-терминалов сейчас используют обновляемый программный продукт, то действия системного администратора могут не только привести к потери не сохраненных документов, но и к искажению их содержимого. Поэтому обновление программного обеспечения следует выполнять либо в нерабочее время, либо в то время, когда, каждый пользователь осведомлен о предстоящих работах на сервере.

Вообще, при администрировании работы сети X-терминалов никогда не следует забывать о ее многопользовательской природе. Часто нерадивое отношение одного пользователя может пагубно сказаться на работе всех его коллег. Например, одному пользователю вздумалось поиграть в компьютерную игру, которая значительно нагружает локальную сеть и загрузка центрального процессора держится на отметке в 100%. Понятно, что от такого использования X-терминала не по назначению пострадают все остальные пользователи. Для успешного противостояния такому поведению нужно разработать систему реагирования, которая должна содержать в себе еще и административные

санкции. Конечно, системный администратор может прекратить выполнение любого процесса в системе, и даже заблокировать работу любого пользователя, но без поддержки руководящего звена предприятия все его усилия будут безрезультатны.

При администрировании сети X-терминалов системный администратор в основном выполняет привычные функции по добавлению/удалению пользователей, распределению их полномочий и прав, а также по первичной настройке используемого ими программного обеспечения. Если текучесть кадров не велика, то рутинной работы совсем не много. Более того, сервер работает в графическом режиме, так что ничто не мешает использовать привычные средства с графическим интерфейсом для этих целей, например, такие как redhat-config-users (утилита для управления учетными записями пользователей и групп).

Планирование эксплуатационного обслуживания сервера (продолжение)

Для того чтобы эффективно работать с пользователями, системный администратор должен знать, какими прикладными программами пользуются в своей повседневной работе пользователи X-терминалов. Для автоматического составления такого списка можно воспользоваться нехитрым сценарием на языке Perl, который необходимо запустить на выполнение в фоновом режиме:

```
#!/usr/bin/perl -w

my $log_dir = 'udata/';
my %user_tasks = ();
$|=1;

print "Logging all user tasks\n";
while(1) {
    # Получаем список уникальных процессов
    my @procs = `ps axh -o "user comm" | sort | uniq`;
    for (@procs) {
  my ($user, $ps_name) = split /\s+/;
```

```
  ${ $user_tasks{$user} }{$ps_name} = 1;
   }

   # Цикл по процессам пользователей
   for (keys %user_tasks) {
  my %task_by_user = %{ $user_tasks{$_} };
  my $fn_save = $log_dir.$_;
  open (FUTASK, ">$fn_save") or die "Can't save file $!\n";
  print FUTASK join ("\n", sort keys %task_by_user);
  close(FUTASK);
   }
   print '.';
   sleep 30;
}
```

Его запуск приводит к сбору статистики о процессах, выполняемых от имени всех пользователей системы, при этом статус суперпользователя вовсе не требуется. Как видно из текста сценария, он с интервалом в 30 секунд записывает в директорию, указанную в переменной $log_dir (в сценарии это каталог udata) файлы, совпадающие по именам с пользователями, работающими в системе. Содержимое этих файлов – список процессов, которые запускал пользователь, например:

```
$ cat udata/nata
ROX-Filer
icewm
icewm-session
icewmbg
icewmtray
kdeinit
sim
soffice.bin
```

Для получения более детальной информации о том какие программы, когда и как часто запускают пользователи, необходимо создать более совершенные сценарии. Благо, в распоряжении системного администратора для этих целей в операционной системе Linux есть все средства. Например, если вас интересует информация о том сколько по времени был запущен тот или иной пользовательский процесс, то для ответа на этот вопрос можно использовать предыдущий сценарий, но немного в модифицированной форме:

```
#!/usr/bin/perl -w

my $log_dir = 'udata/';
my %user_tasks = ();
$|=1;

print "Logging all user tasks and count\n";
while(1) {
   # Получаем список уникальных процессов
   my @procs = `ps axh -o "user comm" | sort | uniq`;
   for (@procs) {
  my ($user, $ps_name) = split /\s+/;
  ${ $user_tasks{$user} }{$ps_name}++;
   }

   # Цикл по процессам пользователей
   for (keys %user_tasks) {
  my %task_by_user = %{ $user_tasks{$_} };
  my $fn_save = $log_dir.$_;
  my @result_array = map { $_ . "\t" . $task_by_user{$_}  } sort keys %task_by_user;
  open (FUTASK, ">$fn_save") or die "Can't save file $!\n";
  print FUTASK join ("\n", @result_array);
  close(FUTASK);
```

```
    }
    print '.';
    sleep 30;
}
```

Теперь если взглянуть на содержимое файлов директории udata, можно заметить, что имени процесса соответствует числовое значение частоты присутствия его в виртуальной памяти сервера X-терминалов. Например, чтобы просмотреть любимые процессы пользователя с именем ula, отсортированные по частоте использования, можно применить такую простую команду (первыми идут те процессы, которые пользователь запускает чаще):

```
$ sort -k 2 -n -r udata/ula
wombat  246
ssh-agent       246
oafd    246
icewmtray       246
icewm-session   246
icewmbg 246
icewm   246
evolution-alarm 246
bonobo-moniker- 246
ROX-Filer       178
soffice.bin     165
klines  131
kdeinit 131
```

Разделив число напротив имени процесса на 2 вы получите примерное время работы его на сервере в минутах. Понятно, что процессы завершающиеся за очень короткое время скорее всего не попадут в этот список. К еще одному недостатку данного сценария можно отнести невозможность определения того сколько процессорного времени тратит тот или иной процесс, то есть реально ли работает с этой программой пользователь или просто она запущена в

фоновом режиме. Для ответа на этот и другие более сложные вопросы можно обратится к статистическим данным предоставляемым системой /proc.

Если рассмотреть процесс администрирования сервера X-терминалов и обычного однопользовательского компьютера, то сразу заметным станет главное отличие: работы по администрированию X-терминал сервера следует выполнять в определенное запланированное время, а для обычного компьютера такого жесткого планирования может и не быть. Чтобы эта идея стала более понятной, приведу реальный случай. Решил я как-то сделать резервную копию пользовательских данных, общих папок и каталога конфигурации системы. Написал для этих целей простенький сценарий и запустил его от имени суперпользователя. Дело происходило в рабочее время и я стал свидетелем того, как работающие в системе пользователи ощутили на себе все "прелести" 20-ти минутного фонового процесса, нагружающего на 100% центральный процессор (программа gzip) и дисковую подсистему (программа tar). Не следует забывать также, что задача работала от имени суперпользователя, а стало быть, имела больший приоритет. Этого горького опыта мне с лихвой хватило, чтобы понять как важно правильно выбирать время для выполнения задач администрирования. За основу принятия решения о необходимости выполнения действий по администрированию, я рекомендую использовать такой алгоритм:

Определяем проблему;

Вырабатываем метод решения проблемы;

Если решение проблемы - процесс не длительный и он не отнимает много системных ресурсов, то применяем его сразу;

В противном случае планируем время запуска административного сценариям;

Для выполнения задач регулярного администрирования используем утилиту crontab, а для разовых заданий - утилиту at.

Как уже было замечено ранее, для выполнения задач требующих значительных вычислительных ресурсов, лучше подходит ночное время. Чтобы не нагружать сервер X-терминалов понапрасну, рекомендуется запускать

задания в автоматическом режиме в разное время, например, с интервалом в несколько минут.

Кроме администрирования, X-терминал сервер требует и обыкновенного профилактического обслуживания. Чтобы с аппаратным обеспечением, а как следствие, и программным все было в порядке, необходимо обеспечить приемлемые условия для работы сервера X-терминалов. Во-первых, сервер должен быть размещен на устойчивой, неподвижной поверхности. Лучше, если это будет специальный шкаф для коммуникационного оборудования. Во-вторых, во избежание выхода техники из строя необходимо обеспечить нормальный температурный режим того помещения, где будет расположен сервер. Никаких фантастических характеристик по температуре и влажности соблюдать не нужно, хватит вполне нормальных условий от 18 до 24 (??? градусов Цельсия). В-третьих, важно бороться с пылью, которая может попасть внутрь корпуса X-терминал сервера. Сама по себе пыль опасности не представляет, но то, что она проводит электрический ток, а также забивает вентиляторы и мешает циркуляции воздуха внутри корпуса, представляет значительную опасность. Если помещение достаточно запыленное, то хотя бы раз в три месяца рекомендуется производить чистку системного блока X-терминал сервера. С примером неправильного размещения сервера X-терминалов вы можете ознакомится на рис. 5.2.

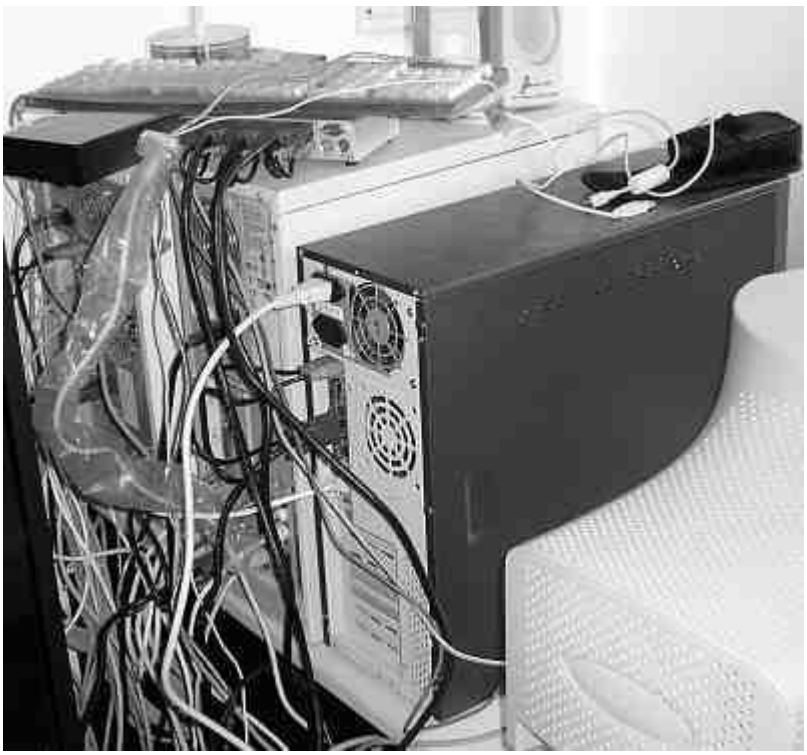

Рис. 5.2. Неправильное расположение компьютеров (слева, сервер X-терминалов)

Неплохо также иметь средства мониторинга температуры внутри корпуса сервера. Так как перегрев - это причина не только выхода из строя аппаратного обеспечения, а еще и составляющая сложно уловимых ошибок в работе оборудования. Любой дисковый накопитель обязан иметь дополнительное охлаждение. Вентилятор для "винчестера" стоит не дорого и позволяет значительно повысить надежность хранения информации на вашем сервере.

Хорошего администратора от плохого отличает то, что пользователи практически не замечают работы первого. Умение предвидеть проблему до ее появления – это настоящее искусство, которому довольно сложно научиться. Своеобразный парадокс системного администрирования связан с тем, что чем больше времени потратить на настройку системы, тем меньше потом она требует обслуживания. Работа с X-терминал сервером только подтверждает это правило. Если сервер настроен правильно, то кроме работы с пользователями он практически не требует вмешательства в свои дела со стороны человека. В идеальном варианте вы должны получить самодостаточную систему, администрировать которую одно удовольствие.

По своему опыту знаю, что работа администратора операционной системы Linux содержит меньше рутинных операций, чем, например, администрирование операционных систем win32. Более интенсивное использование командной оболочки позволяет выполнять те же действия, но за более короткое время. Хотите верьте, хотите нет, но после перехода к использованию операционной системы Linux на рабочем месте у меня перестало болеть запястье правой руки (сказывается уменьшение времени использования манипулятора "мышь"). Я снова стал пользоваться клавиатурой! Когда пользователи начали переходить от традиционных персональных компьютеров с Linux к X-терминалам, то работа по системному администрированию стала еще проще.

Несмотря на то, что администрирование X-терминал сервера связано с определенными тонкостями и нюансами, неоспоримым остается факт, что

обслуживание N пользователей X-терминалов значительно проще, чем обслуживание N пользователей традиционных персональных компьютеров.

Перенос операционной системы на новый сервер

Связывая свою дальнейшую деятельность по эксплуатации компьютерной техники с X-терминалами, следует быть готовым к переносу информации на новый сервер. Вызвано это, в первую очередь, необратимым поступательным развитием информационной технологии. Как бы ни хорош был сервер, купленный 3 года назад, он вряд ли справится со всеми требованиями, которые выдвигает современное программное обеспечение. Модернизация сервера возможна, но ограничена, как правило, материнской платой, так как с заменой этого компонента сопряжено довольно много рабочих параметров операционной системы. Нарастить оперативную память, увеличить объем дисковой подсистемы, поставить более производительный процессор не является проблемой, но в определенный момент возможности модернизации иссякают. Именно в этот момент встает вопрос о покупке нового сервера.

За перенос операционной системы на новый, более производительный сервер выступает и тот факт, что за время роботы программного обеспечения файловая система обрастает избыточным количеством временных файлов, которые не всегда хранятся в каталоге /tmp, а следовательно найти и удалить их совсем не просто. С этой точки зрения перенос операционной системы на новый сервер подобен переезду в новый офис после ремонта: вначале несколько непривычно и придется попотеть (перенести пользовательские каталоги, настройки демонов и т.д.), но зато со временем работать станет значительно лучше. К тому же в новой (чистой) операционной системе можно реализовать функции, которые были, к сожалению, изначально не применены в старой. Например, вам всегда хотелось настроить систему учета сетевого трафика в разрезе пользователей X-терминалов, а сделать это в период эксплуатации старого сервера было сложно ввиду его постоянной загруженности. Еще одним хорошим примером использования нового X-терминал сервера является построение системы RAID в режиме зеркалирования, для повышения надежности хранения информации на жестких

дисках. В общем, аргументов довольно много, и рано или поздно любая система X-терминалов столкнется с необходимостью переноса информации на новый сервер, а чтобы осуществить это с наименьшими негативными последствиями для конечных пользователей, следует заранее продумать, какие мероприятия должны быть для этого выполнены.

Разумно составить план действий по переносу системы на новый сервер. Чем более детальным будет этот план, тем меньше потом будет нареканий на работу в новых условиях. Итак, переносу подлежат:

информация о пользователях и группах (в операционных системах Linux это соответственно файлы: /etc/passwd, /etc/group, а также и их защищенные аналоги /etc/shadow и /etc/gshadow);

данные о доменных именах узлов локальной сети, в частности файл /etc/hosts;

каталоги пользователей, которые хранятся в директории /home;

настройки программного комплекса по управлению X-терминалами (при переходе на новую версию файлы конфигурации потребует изменений);

настройки локальной сети и сетевых интерфейсов сервера (IP-адреса, шлюз по умолчанию, сервер имен и т.д.);

если на X-терминал сервере использовался брандмауэр (firewall), то его конфигурацию также необходимо перенести на новый сервер;

согласно использующемуся программному обеспечению, переносу на новый сервер подлежат также каталоги с совместным доступом. Например, в случае использования X-терминал сервера как контроллера домена для сети Microsoft, переносятся храненимые на нем общие сетевые каталоги, управления на которыми осуществляется при помощи Samba-сервера;

отдельного внимания заслуживает перенос подсистемы печати операционной системы Linux на новый сервер. Сложность заключается в том, что все принтеры, которые были подключены к X-терминалам настраиваются непосредственно на сервере, а следовательно их настройку необходимо осуществлять именно на нем;

в низкобюджетных системах на X-терминал сервер возлагают также функции Интернет-сервера, а как результат, на нем часто работает прокси-сервер, почтовый сервер и даже веб-сервер. Если ваша система относится именно к таким, то необходимо также осуществить перенос на новый сервер и этих служб;

отдельно по каждому использующемуся сервису необходимо составить план переноса его настроек на новый сервер.

Как видно, это только краткий список действий, которые необходимо запланировать. Возможно, в вашей конкретной системе он будет значительно обширней.

Откровенно говоря, перенос операционной системы на новую аппаратную платформу это довольно серьезный шаг. Перед тем как переносить данные со старого сервера, необходимо, чтобы на новом X-терминал сервере операционная система работала как часы. Рекомендуется сначала настроить сервер, тщательно протестировать его работоспособность, а уже после этого приступить к процессу копирования информации. При этом очень важно учитывать фактор времени, так как при возникновении непредвиденных трудностей, перенос операционной системы может затянуться, а согласитесь, что это никак не входило в ваши планы, и поверьте мне, особенно в планы ваших пользователей. Чтобы при переносе информации не страдала производительность труда, выполнять эту процедуру придется в выходной день, так что заранее можно обрадовать системного администратора обещанием отгула за работу в неурочное время.

Лично я не сторонник замены старого сервера на новый за один пусть даже выходной день. Работа с таким жестким ограничением по времени чревата потенциальными ошибками, которые потом будет довольно сложно исправить. Если организация столкнулась с необходимостью в наращивании аппаратных возможностей X-терминал сервера, то что мешает настроить новый сервер на параллельную работу со старым, а также постепенно осуществить перенос всех пользовательских учетных записей на новую платформу?

Перенос операционной системы на новый сервер (продолжение)

Давайте рассмотрим конкретный случай модернизации (замены) X-терминал сервера. Чтобы придать этому примеру более реалистичный вид, определим какие функции были возложены на него. Вот краткий перечень главных из них:

работа с удаленными пользователями менеджера дисплея;

управление учетными записями пользователей X-терминалов;

предоставление доступа к общим файлам для клиентов сети Microsoft посредством Samba-сервера.

Схема локальной сети предприятия, состоящая из 5 персональных компьютеров с операционной системой Windows и 10 X-терминалов представлена на рис. 5.3. Обратите внимание, что для доступа в Интернет используется отдельный сервер.

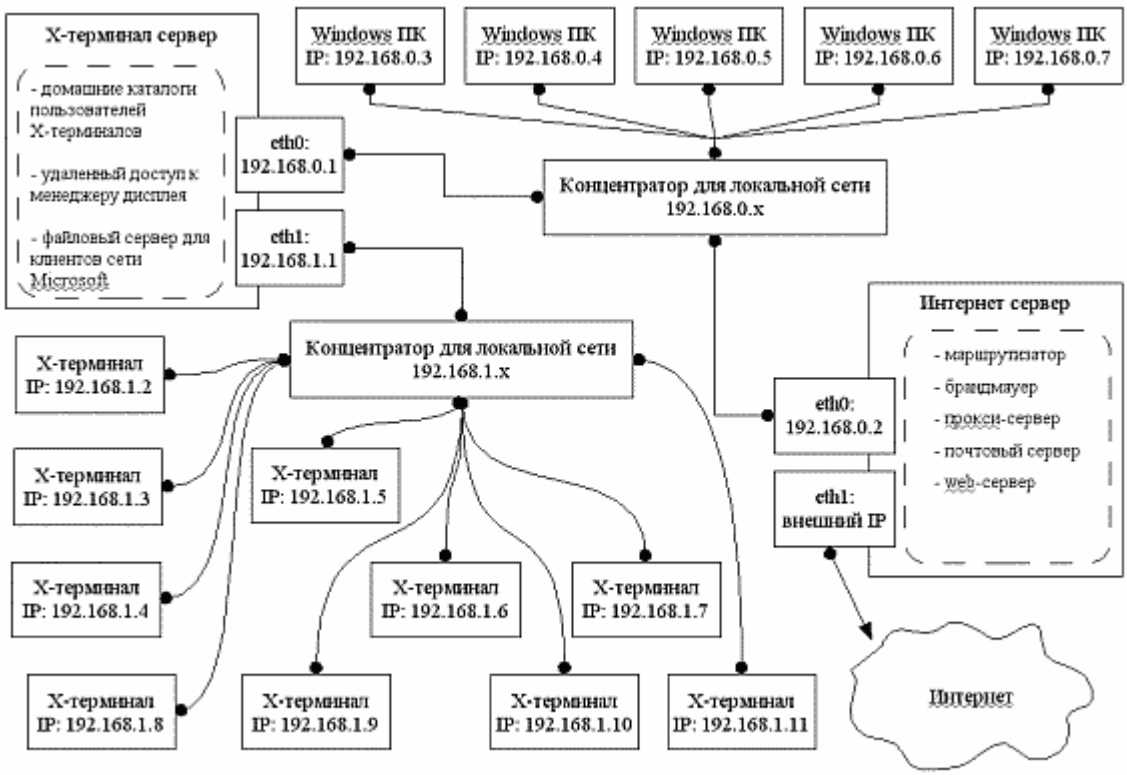

Рис. 5.3. Схема существующей локальной сети предприятия

X-терминал сервер обслуживает сразу две локальные сети: пользователей компьютеров с Windows (192.168.0.x) и клиентов X-терминалов (192.168.1.x). Когда возникает необходимость в использовании нового X-терминал сервера, он подключается к уже существующей локальной сетевой структуре, но с новыми IP-адресами, например, 192.168.0.10, для работы с Windows-клиентами,

и 192.168.2.1 для постепенного перевода пользователей X-терминалов на новый сервер.

Первым делом на новый сервер устанавливается программное обеспечение для работы X-терминалов и настраивается на оптимальную производительность. Затем начинается постепенный перенос пользователей на новый сервер. Например, X-терминал с IP-адресом 192.168.1.2 подключается к новому серверу и теперь у него IP-адрес будет равен 192.168.2.2.

Очень важно, чтобы на старом и новом сервере совпадали идентификаторы пользователей, тогда можно достичь полной преемственности при переносе информации, особенно для каталогов с совместным доступом. Добиться этого можно путем копирования файлов с именами пользователей, паролей и групп, а также путем использования таких информационных служб как NIS или LDAP. Представим, что за терминалом с IP-адресом 192.168.1.2 работал пользователь sasha. Для того, чтобы ему был доступен его домашний каталог, скопируем директорию /home/sasha со всем содержимым со старого сервера на новый. Проблемы это не составляет, так как оба сервера находятся в одной подсети благодаря их сетевым интерфейсам eth0 (192.168.0.1 и 192.168.0.10). Теперь у нас получилось два домашних каталога пользователя sasha: один на старом, а другой на новом сервере. На практике это означает, что сев за X-терминал с адресом 192.168.2.2, он попадет в домашний каталог 192.168.2.1:/home/sasha, а работая на любом другом терминале, он фактически будет обращаться к 192.168.1.1:/home/sasha. Чтобы исключить такую двойственную природу, следует заблокировать пользователя sasha на старом сервере. Сделать это можно такой простой командой:

```
# passwd -l sasha
Locking password for user sasha.
passwd: Success
```

Конечно, при такой схеме мы теряем возможность универсальной работы пользователей на любом X-терминале, зато позволяем осуществить перенос данных на новый сервер в текущем режиме. Теперь пользователь X-терминала

с именем sasha может работать только на одном X-терминале у которого IP-адрес 192.168.2.2 (см. рис. 5.4).

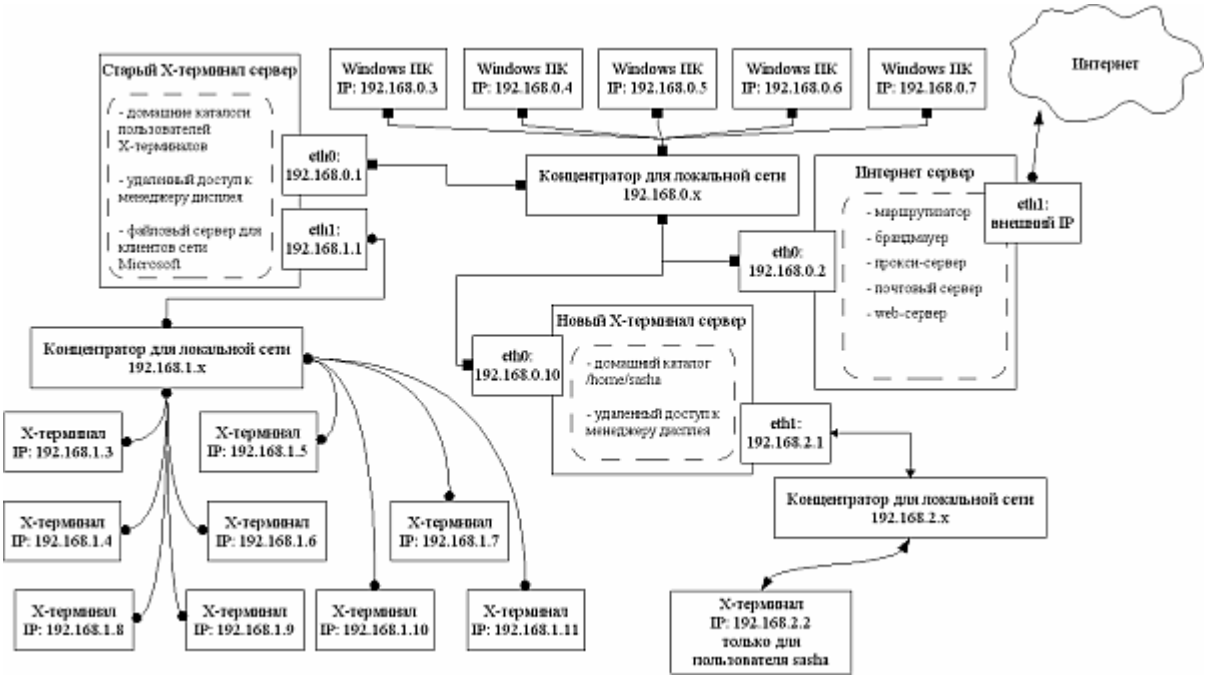

Рис. 5.4. Схема локальной сети предприятия с новым X-терминал сервером

Для доступа пользователей нового X-терминал сервера к ресурсам остальной локальной сети используется, как уже было сказано выше, сетевой интерфейс eth0, и по сути, если смотреть в разрезе IP-адресов подсети 192.168.0.x, то в локальной сети не 18 компьютеров, а только 8 (5 ПК с Windows, 2 X-терминал сервера и 1 Интернет сервер). А находясь в одной локальной сети, нет никаких проблем с доступом к общим файлам и другим сетевым ресурсам, например, к принтерам. Используя подсистему печати CUPS, можно вообще организовать прозрачный доступ к принтерам, которые подключены к X-терминалам или персональным компьютерам с операционной системой Windows. После соответствующей настройки нового X-терминал сервера ему автоматически будет доступен список всех принтеров со старого сервера. Правда, если к X-терминалу с IP-адресом 192.168.2.2 был подключен локальный принтер, то необходимо правильно прописать его в новом сервере, а со старого удалить, так как теперь у этого принтера будет новый IP-адрес.

Пользователи при работе на новом сервере могут столкнуться с проблемой частичной потери их личной информации, и это особенно справедливо при переходе на сервер с новой операционной системой. Например, при использовании новой версии браузера могут быть потеряны закладки

(bookmarks), а при переходе к работе с новым почтовым клиентом – список контактов и даже почтовая база. Чтобы быть готовым к таким трудностям системному администратору следует заранее проверить: как при переходе к новой версии программного продукта осуществляется конвертация данных от предыдущих версий. И в случае, если такой процесс не происходит автоматически, необходимо предусмотреть способы выполнения его вручную. Такая ситуация является еще одним аргументом в пользу постепенного перевода пользователей на новый сервер. По крайней мере вы будете выслушивать претензии от одного неудовлетворенного пользователя X-терминала, а не от всех одновременно.

Когда работа по настройке одного X-терминала будет закончена, следует переходить к копированию информации пользователей других X-терминалов. И постепенно в компетенции старого сервера будут только обязанности по обслуживанию клиентов сети Microsoft. Для переноса конфигурации сервера Samba следует сначала остановить его работу на старом сервере, привести в соответствие с существующими требованиями настройки сервера Samba на новом сервере, переписать каталоги, которые хранят общие сетевые ресурсы и запустить Samba-сервер на новом X-терминал сервере. После проверки работоспособности новой сетевой среды можно смело выключать старый сервер. С новой структурой локальной сети можно ознакомиться на рис. 5.5.

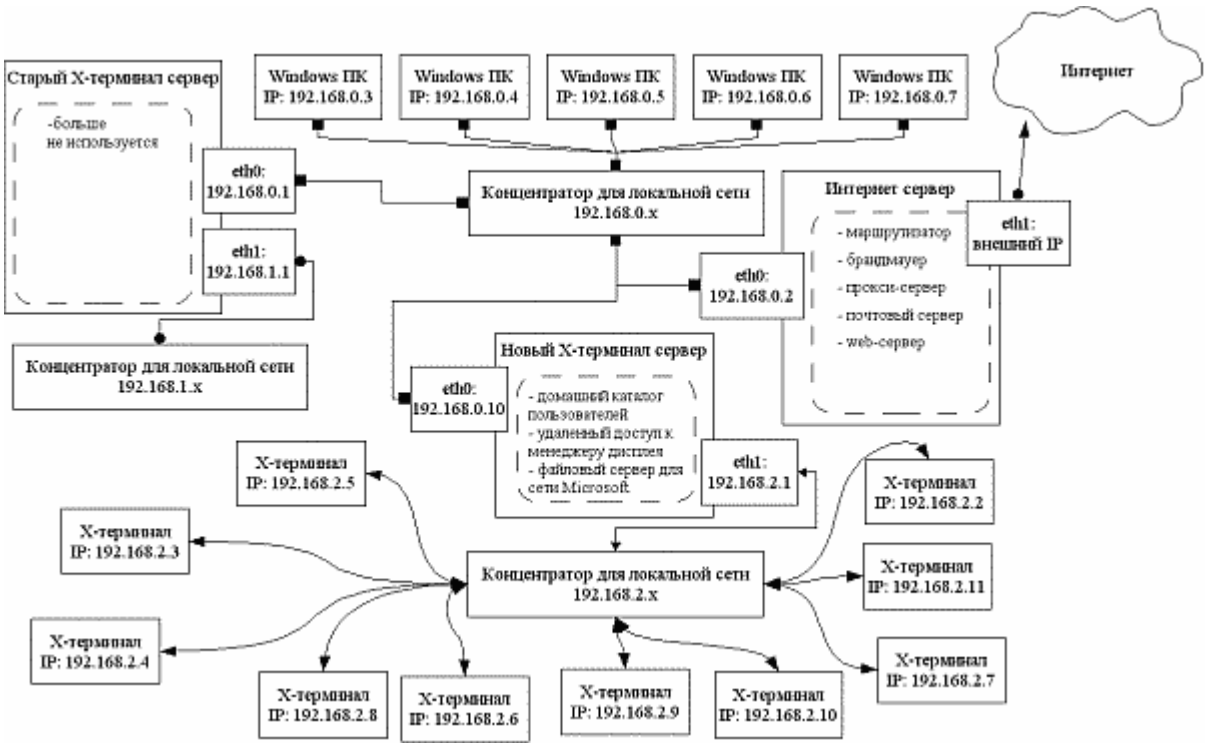

Рис. 5.5. Схема локальной сети предприятия без старого X-терминал сервера

Для чистоты эксперимента нужно отметить, что перенос информации на новый сервер произошел с частичной сменой IP-адресов сетевых интерфейсов. Если это так важно, то можно было сначала изменить IP-адреса работы старого сервера и X-терминалов, а уже затем переносить информацию на новый сервер. Или уже на новом работающем сервере изменить привязку интерфейсов к IP-адресам. Также следует помнить, что для простоты работы с адресами компьютеров была придумана доменная система имен, так почему бы не пользоваться именно ею, вместо использования числовых значений.

Интересно, что в некоторых случаях, при покупке нового высокопроизводительного X-терминал сервера, его используют не вместо, а совместно со старым сервером. Существуют даже проекты объединения серверов в кластер для увеличения производительности системы. Как видите, возможности использования компьютерной техники в терминальных решениях достаточно широки, но главное применять их так, чтобы это, в первую очередь, положительно сказывалось на работе конечных пользователей.

Часть III. Настройка системы (на примере использования LTSP)

Глава 6. Настройка сервера

Установка программного обеспечения (ltsp, nfs, dhcpd, tftp-server ...)

Для установки программного обеспечения, которое требуется для работы сети X-терминалов, следует сначала убедиться, что в наше распоряжение попал хороший и настроенный сервер с операционной системой Linux. Соблюдение этого условия очень важно, так как при настройке X-терминал сервера можно допустить определенные ошибки или неточности, которые воспрепятствуют его правильной работе, и хорошо когда проблемы будут вызваны вашими последними ошибками, а не теми, что вы допустили еще во время установки операционной системы. Если сервер в порядке, то самое время переходить к изучению тонкостей настройки сервера LTSP (Linux Terminal Server Project – проект терминального сервера в Linux).

Официальный сайт проекта LTSP (http://www.ltsp.org) – очень информативный ресурс. Он в частности содержит, кроме программных пакетов, также множество документации, посвященной этому проекту.

Официальный сайт проекта LTSP (http://www.ltsp.org) – очень информативный ресурс. Он в частности содержит, кроме программных пакетов, также множество документации, посвященной этому проекту.

Итак, начнем. Для начала загружаем из сети Интернет пакеты проекта LTSP. В частности, нам понадобятся:

ltsp_core– главный пакет системы LTSP, содержит в себе все необходимое для работы X-терминальной среды;

ltsp_kernel– скомпилированное ядро, которое будет загружаться удаленными X-терминалами;

ltsp_x_core– пакет, содержащий программную среду для работы в X-Windows на X-терминале;

Для дистрибутивов операционной системы Linux, которые базируются на RPM-пакетах это соответственно: ltsp_core-3.0.9-0.i386.rpm, ltsp_kernel-3.0.13-0.i386.rpm и ltsp_x_core-3.0.4-0.i386.rpm. Устанавливаем эти пакеты командами:

```
# rpm -ihv ltsp_core-3.0.9-0.i386.rpm
# rpm -ihv ltsp_kernel-3.0.13-0.i386.rpm
# rpm -ihv ltsp_x_core-3.0.4-0.i386.rpm
```

По умолчанию все содержимое этих пакетов будет установлено в каталог /opt/ltsp.

Есть еще один пакет, который входит в состав установки проекта LTSP: ltsp_x_fonts, но его устанавливать нужно лишь в том случае, если вы решили отказаться от использования общего сервера шрифтов xfs на сервере X-терминалов. А так как все терминалы должны представлять из себя однообразную рабочую среду, то более целесообразно использовать один сервер шрифтов для всей локальной сети. Придерживаясь такой идеологии, мы с минимальными усилиями получаем единый набор шрифтов на всех X-терминалах локальной сети.

Если ваш любимый дистрибутив не поддерживает формат установочных пакетов RPM, или вам просто нравится устанавливать программы без посредников, то вы можете воспользоваться пакетами в формате *.tgz: ltsp_core-3.0.9-i386.tgz, ltsp_kernel-3.0.13-i386.tgz и ltsp_x_core-3.0.4-i386.tgz. Устанавливать их на сервер X-терминалов нужно при помощи таких команд:

```
# tar xzvf ltsp_core-3.0.9-i386.tgz
# cd ltsp_core
# ./install.sh
# cd ..
# tar xzvf ltsp_kernel-3.0.13-i386.tgz
# cd ltsp_kernel
# ./install.sh
# cd ..
# tar xzvf ltsp_x_core-3.0.4-i386.tgz
# cd ltsp_x_core
# ./install.sh
# cd ..
```

Как и для пакетов в формате RPM они по умолчанию будут установлены в каталог /opt/ltsp/i386.Для пользователей дистрибутива Debian операционной системы Linux на сайте www.ltsp.org можно взять пакеты с расширением *.deb.

Установка программного обеспечения (продолжение)

Следующим этапом необходимо правильно сконфигурировать серверную часть установленных пакетов, но для этого нам могут понадобиться дополнительные сетевые службы.

Во-первых, необходимо удостовериться, что в вашей системе присутствуют сетевые службы, которые необходимы для работы проекта LTSP и вообще X-терминалов. К этим службам относятся:

DHCP (Dynamic Host Configuration Protocol – протокол динамического конфигурирования хостов). Если сказать другими словами, то DHCP – это специальный протокол предназначенный для определения сетевой конфигурации удаленного компьютера. Он, как правило, используется для

раздачи динамических IP-адресов компьютерам локальной сети (см. RFC 2131 и RFC 2132);

TFTP (Trivial File Transfer Protocol – тривиальный протокол передачи файлов) – специальный протокол для загрузки операционной системы на бездисковые станции. Из-за простоты реализации его код умещается даже в крохотные микросхемы BootROM сетевых адаптеров;

NFS (Network File System – сетевая файловая система). NFS, как сетевая файловая система, предоставляет пользователям локальной сети общий доступ к файлам, которые хранятся на NFS-сервере. Фактически удаленные пользователи могут обращаться к ним как к сетевым дискам. Разработана NFS еще в начале 80-х годов, поэтому проверена временем и довольно стабильна. Для ее работы на компьютере должна быть запущена служба RPC (Remote Procedure Call – служба вызова удаленных процедур), которая в операционной системе Linux называется portmap;

xinetd (extended Internet services daemon – демон расширенной службы Интернет) предназначен для обработки входящих обращений к серверу. xinetd является более защищенной версией своего предка inetd.

Если вы не имели опыта работы с этими программными продуктами, то в этом нет большой проблемы, так как настройка и использование этих программ не составит большого труда. Итак, приступим к установке требуемых программных пакетов.

Для начала определяем, какие из пакетов уже установлены в системе. В командной строке вводим команду:

```
$ rpm -qa | grep dhcp
dhcp-3.0pl1-23
```

Ее вывод показывает, что в системе уже установлен DHCP. Если установленного DHCP не обнаружено, то устанавливаем его:

```
# rpm -ihv dhcp-3.0pl1-23.i386.rpm
```

Аналогично проверяем, установлен ли TFTP-сервер:

```
$ rpm -qa | grep tftp
tftp-server-0.33-1asp
```

Если же и его нет в списке установленных пакетов, то запускаем инсталляцию этого пакета:

rpm -ihv tftp-server-0.33-1asp.i386.rpm

Поддержка сетевой файловой системы скорее всего уже есть в вашей операционной системе, но проверить не мешает:

$ rpm -qa | grep nfs

nfs-utils-1.0.1-2.9

Если же вы не нашли установленным пакет nfs-utils, то сейчас самое время его установить на сервере X-терминалов:

rpm -ihv nfs-utils-1.0.1-2.9.i386.rpm

Демон xinetd, как правило, всегда установлен, но если нет, то установить его также не сложно:

$ rpm -qa | grep xinetd

xinetd-2.3.10-6

В данном примере демон xinetd присутствует в нашей системе. В противном случае устанавливаем его командой:

rpm -ihv xinetd-2.3.10-6.i386.rpm

Установка программного обеспечения (продолжение)

Когда все нужные пакеты установлены, можно приступать к настройке сервера X-терминалов. Для этого переходим в каталог /opt/ltsp/templates и от имени суперпользователя root запускаем сценарий ltsp_initialize, который выполняет инициализацию проекта LTSP:

$ su -l

Password:

cd /opt/ltsp/templates

./ltsp_initialize

Этот сценарий в диалоговой форме показывает: какие из системных служб будут сконфигурированы, а также требует подтверждения выполненным изменениям в настройке системы. К сожалению, версия LTSP 3.0 имеет ограничение по количеству поддерживаемых дистрибутивов операционной системы Linux, в частности этой версией официально поддерживаются:

Caldera OpenLinux версий 2.3, 2.4;

Conectiva Linux;

Debian GNU/Linux версий 2.0, 2.1, 2.2, 3.0, Hamm, Potato, Slink, Testing, Unstable, Woody;

Mandrake версий 7.2, 8.0, 8.1, 8.2, 9.0, 9.1;

Redhat версий 6.0, 6.1, 6.2, 7.0, 7.1, 7.2, 7.3, 8.0, 9.0;

Suse Linux версий 6.2, 6.3, 6.4, 7.0, 7.0, 7.1, 7.2, 7.3, 8.0, 8.1, 8.2.

Как видите, список достаточно обширен, но далеко не полон. Если вы предпочитаете использовать в качестве серверной операционной системы дистрибутив, который не встречается в приведенном выше списке, то вам придется выполнить дополнительные настройки вручную. Ничего страшного в этом нет, и далее из этой книги вы узнаете как это сделать правильно.

Если же вы счастливый обладатель дистрибутива, что поддерживается проектом LTSP версии 3.0, то после запуска сценария инициализации вы увидите на экране нечто подобное (см. рис. 6.1).

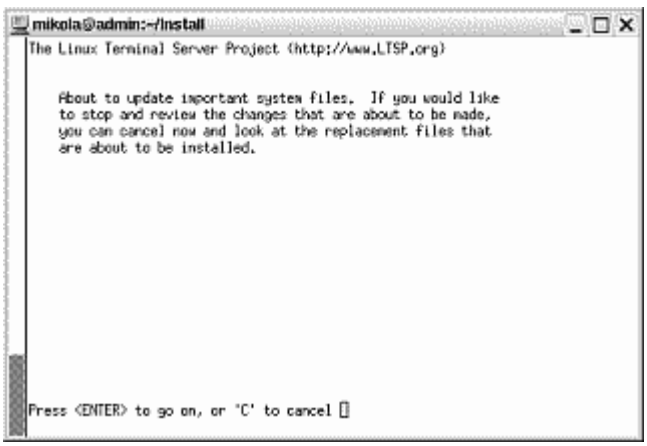

Рис. 6.1. Начало инициализации сервера LTSP

После утвердительного ответа на запрос продолжения установки, на экране появится список файлов, которые будут созданы или изменены в процессе инициализации сервера X-терминалов (см. рис. 6.2).

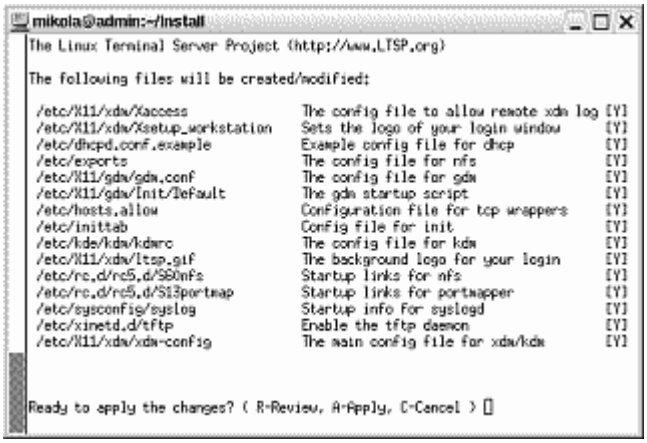

Рис. 6.2. Список файлов, которые будут созданы или изменены при установке LTSP

Для того, чтобы выразить свое согласия с этим списком, и внести необходимые изменения в файлы конфигурации нажмите клавишу “A”, а затем “Enter” для завершения установки проекта LTSP на сервере. Все файлы конфигурации, которые были изменены, будут сохранены на жестком диске сервера под теми же именами и в тех же каталогах, но с расширением *.1 (например, /etc/sysconfig/syslog будет сохранен как /etc/sysconfig/syslog.1).

Если в результате выполнения сценария инициализации сервера LTSP не было замечено ошибок, то после выполнения небольших изменений в файлах конфигурации можно считать настройку сервера законченной. Сейчас самое время перегрузить сервер, дабы удостоверится, что все службы работают нормально и внесенные изменения не отразились на его работоспособности:

/sbin/reboot

Установка программного обеспечения (продолжение)

После выполнения перезагрузки следует также убедиться, что нужные демоны были запущены автоматически после старта системы. Для этого можно, например, воспользоваться утилитой ntsysv:

/usr/sbin/ntsysv

На рис. 6.3 вы можете увидеть экран компьютера с запущенной утилитой ntsysv (в окне Xterm). Пользоваться ей очень просто. Напротив каждого демона, запускающегося автоматически на данном уровне выполнения, стоит звездочка "*", которую можно снять или поставить нажатием на клавишу "Пробел". Так,

из рисунка 6.3 видно, что автоматически будут запускаться демоны: network, nfs, nfslock, portmap и random.

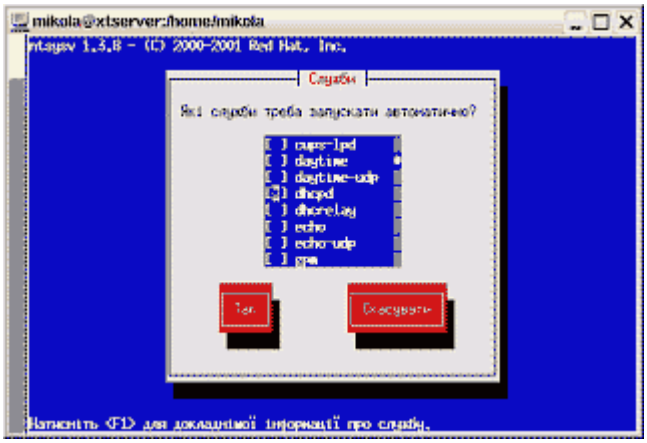

Рис. 6.3. Пример использования утилиты ntsysv

Проверить следует работу демонов: dhcpd, nfs, portmap, syslog, xfs, tftp и xinetd. Если какие-то из них не запущены, то, вполне вероятно, это может быть причиной неработоспособности сети X-терминалов.

Для сторонников использования настоящих консольных утилит с пользовательским интерфейсом командной строки, следует запустить такие команды:

Проверка DHCPD

$ /sbin/service dhcpd status

dhcpd (pid 19402) running...

Проверка NFS и portmap

$ /sbin/service nfs status

rpc.mountd (pid 28633) running...

nfsd (pid 28625 28624 28623 28622 28621 28620 28619 28618) running...

rpc.rquotad (pid 28614) running...

Проверка Syslog

$ /sbin/service syslog status

syslogd (pid 624) running...

klogd (pid 628) running...

Проверка xfs

$ /sbin/service xfs status

xfs (pid 854) running...

Проверка xinetd и tftpd

$ ls -l /etc/xinetd.d/tftp

-rw-r--r-- 1 root root 507 Nov 8 09:04 /etc/xinetd.d/tftp

$ /sbin/chkconfig --list tftp

tftp on

$ ps aux | grep xinetd

root 22530 0.0 0.0 2132 76 ? S Feb01 0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid

mikola 5708 0.0 0.0 3768 600 pts/0 S 10:46 0:00 grep xinetd

$ /sbin/chkconfig --list xinetd

xinetd 0:off 1:off 2:off 3:on 4:on 5:on 6:off

Теперь настало время менять конфигурацию сервера X-терминалов (далее просто сервера) в соответствии с нашими потребностями. Но перед этим следует ввести ряд определений для того, чтобы упростить дальнейшую настройку. Итак, выясним сетевые параметры работы сервера LTSP (см. Таблица 6.1):

Таблица 6.1. Сетевые параметры сервера X-терминалов

Название параметра	Значение параметра
Внешний IP-адрес сервера	192.168.0.26
IP-адрес сервера для работы с X-терминалами	192.168.1.1
IP-адреса локальной сети X-терминалов	192.168.1.x
Домен локальной сети	xterms.net
Доменное имя сервера	xtserver.xterms.net
Адрес шлюза по умолчанию	192.168.0.22
Адрес сервера DNS	192.168.0.22

Вы можете использовать любые значения IP-адресов для настройки своей системы, а те, что предложены в таблице 6.1, взяты в качестве учебного примера.

Проект LTSP был разработан для упрощения настройки и использования X-терминальных решений, и если ваша система была настроена с его помощью без каких-либо проблем, то смело переходите к подразделу этой главы “Главный файл конфигурации сервера LTSP”. Если же при настройки возникли определенные трудности, или вы хотите более подробно разобраться в работе X-терминалов, то следующие страницы главы именно для вас.

Общая настройка системы (inittab, syslog)

Для того, чтобы X-терминал сервер заработал, необходимо изменить параметры сценария инициализации inittab и системного журнала syslog.

Настройка inittab

Ручную настройку сервера целесообразно начать с изучения файла /etc/inittab. Глобально в нем менять ничего не нужно, достаточно лишь удостовериться, что система после старта переходит в графический режим. Это условие крайне важно, так как именно к запущенному X-серверу будут подключаться клиенты X-терминалов. У операционных систем, исторически основанных на RedHat Linux, таким режимом является runlevel=5. О чем в файле /etc/inittab должна свидетельствовать запись “id:5:initdefault:”.

Также именно на 5-м уровне должен запускаться X-сервер строкой, похожей на:

x:5:respawn:/etc/X11/prefdm -nodaemon

После корректировки файла /etc/inittab, следует перейти к настройки системы протоколирования сообщений syslog и ее главному конфигурационному файлу /etc/sysconfig/syslog.

Настройка syslog

Системой LTSP предлагается такое содержание файла /etc/sysconfig/syslog:

```
# Options to syslogd
# -m 0 disables 'MARK' messages.
# -r enables logging from remote machines
# -x disables DNS lookups on messages recieved with -r
# See syslogd(8) for more details
SYSLOGD_OPTIONS="-m 0 -r"
```

Options to klogd

-2 prints all kernel oops messages twice; once for klogd to decode, and

once for processing with 'ksymoops'

-x disables all klogd processing of oops messages entirely

See klogd(8) for more details

KLOGD_OPTIONS="-x"

Как видно из приведенного листинга, от стандартных установок этот файл отличается только тем, что разрешает протоколирование сообщений от удаленных компьютеров (в нашем случае рабочих станций X-терминалов). Иными словами, достаточно всего лишь добавить опцию “-r” в параметрах этого файла и перезапустить сервер-демон:

/sbin/service syslog restart

Shutting down kernel logger: [OK]

Shutting down system logger: [OK]

Starting system logger: [OK]

Starting kernel logger: [OK]

Теперь системный журнал /var/log/messages будет содержать не только локальные сообщения, но и записи, полученные от подключенных к серверу удаленных X-терминалов, например:

tail /var/log/messages

Mar 10 16:07:28 xtserver gconfd (ira-22461): GConf server is not in use, shutting down.

Mar 10 16:07:28 xtserver gconfd (ira-22461): Exiting

Mar 10 16:08:35 sergx -- MARK --

Mar 10 16:09:43 xtserver samba(pam_unix)[21720]: session closed for user luba

Mar 10 16:14:52 irax -- MARK --

Mar 10 16:16:43 manalx -- MARK --

Mar 10 16:27:23 xtserver gdm(pam_unix)[21428]: session closed for user serg

Mar 10 16:28:54 xtserver gconfd (serg-21494): GConf server is not in use, shutting down.

Mar 10 16:28:54 xtserver gconfd (serg-21494): Exiting

Mar 10 16:32:39 xtserver su(pam_unix)[22976]: session opened for user root by mikola(uid=501)

Конфигурирование сетевых служб (dhcp, tftp, nfs, tcpd, dns)

Рассмотрим более детально сценарий инициализации системы LTSP. Сценарий ltsp_initialize содержит в себе более 14 Кбайт текста, и детально разбирать его структуру не нужно, достаточно обратить внимание на то, какие именно сценарии он вызывает для конфигурации системы LTSP. Так, из сетевых служб он настраивает демоны: dhcpd, nfs, tcpd, xinetd и tftpd.

Настройка DHCP

Для службы динамического конфигурирования хостов (DHCP) при инициализации системы LTSP создается файл-образец с именем /etc/dhcpd.conf.example . Содержимое этого файла следующее:

```
# Sample configuration file for ISCD dhcpd
#
# Make changes to this file and copy it to /etc/dhcpd.conf
#
ddns-update-style         none;

default-lease-time            21600;
max-lease-time                21600;

option subnet-mask            255.255.255.0;
option broadcast-address      192.168.0.255;
option routers                192.168.0.254;
option domain-name-servers    192.168.0.254;
option domain-name            "yourdomain.com";
option root-path              "192.168.0.254:/opt/ltsp/i386";

option option-128 code 128 = string;
option option-129 code 129 = text;
```

```
shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }
}

group   {
   use-host-decl-names       on;
   option log-servers        192.168.0.254;

   host ws001 {
      hardware ethernet     00:E0:06:E8:00:84;
      fixed-address         192.168.0.1;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
##       option option-128     e4:45:74:68:00:00;   #This is NOT a MAC address
##       option option-129     "NIC=ne IO=0x300";
   }
   host ws002 {
      hardware ethernet     00:D0:09:30:6A:1C;
      fixed-address         192.168.0.2;
      filename              "/lts/vmlinuz-2.4.9-ltsp-6";
   }
}
```

Как видно из приведенной выше вставки, в этом файле следует обратить особое внимание на такие строчки:

option subnet-mask– означает маску подсети;

option broadcast-address – определяет широковещательный адрес в локальной сети;

option routers – указывает на шлюз по умолчанию;

option domain-name-servers – содержит адрес DNS-сервера;

option domain-name – это то имя, которое будет использоваться в качестве имени сервера для X-терминалов;

option root-path – корневая директория пакета LTSP (на сервере это, как правило, каталог /opt/ltsp/i386);

subnet 192.168.0.0 netmask 255.255.255.0 – описывает локальную сеть в виде адреса и маски подсети;

option log-servers – определяет сервер, на котором будет происходить регистрация системных событий (ведение журнала);

host – указывает на параметры определенного X-терминала. В частности, здесь можно указать его имя, IP-адрес, MAC-адрес сетевой карты, имя файла ядра операционной системы Linux для загрузки ее по сети и т.д.

Если вы имеете опыт настройки протокола динамического конфигурирования хостов (DHCP), то должны знать, что его параметры могут располагаться как в общей части файла /etc/dhcpd.conf, так и в подгруппах. При этом можно настроить раздачу сетевых параметров сразу через несколько сетевых интерфейсов, и тем самым установка на сервере пакетов LTSP никак не помешает использовать уже имеющуюся систему DHCP. Так, для нашего примера вполне допустимым будет такой файл /etc/dhcpd.conf:

```
# Sample configuration file for ISCD dhcpd
ddns-update-style        none;

default-lease-time          21600;
max-lease-time              21600;

option subnet-mask          255.255.255.0;
option domain-name          "xterms.net";

option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.1.0 netmask 255.255.255.0 {
   }
```

```
}

group   {
   use-host-decl-names          on;
   option log-servers           192.168.1.1;
   option broadcast-address     192.168.1.255;
   option routers               192.168.1.1;
   option domain-name-servers   192.168.1.1;
   option root-path             "192.168.1.1:/opt/ltsp/i386";

   host ws001 {
      hardware ethernet     00:80:48:29:3B:53;
      fixed-address         192.168.1.23;
      filename              "/lts/vmlinuz-2.4.24-ltsp-1";
   }
   host ws002 {
      hardware ethernet     00:80:48:29:67:42;
      fixed-address         192.168.1.15;
      filename              "/lts/vmlinuz-2.4.24-ltsp-1";
   }
}
```

Как видно из приведенного выше файла /etc/dhcpd.conf, практически всю специфичную для настройки X-терминалов информацию, мы поместили в блок group. Тем самым мы разрешили совмещать работу X-терминальной сети и функции динамического конфигурирования хостов на одном сервере.

Когда редактирование файла /etc/dhcpd.conf.example закончено, остается только скопировать его содержимое в файл /etc/dhcpd.conf и запустить демон dhcpd, например, такой командой:

```
# /sbin/services dhcpd start
```

Если запуск демона прошел без ошибок, то самое время сделать его автоматически запускаемым, выполнив от имени суперпользователя такую команду:

/sbin/chkconfig --level 5 dhcpd on

Настройка xinetd

Для правильной работы X-терминалов необходимо, чтобы сервер принимал запросы на удаленную загрузку ядра и запускал по требованию демон tftpd. Для этих целей в проекте LTSP используется суперсервер xinetd. Проверим выполняется ли уже на сервере демон xinetd:

$ ps aux | grep xinetd

root 22530 0.0 0.0 2132 288 ? S Feb01 0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid

mikola 24651 0.0 0.0 3764 596 pts/1 S 14:47 0:00 grep xinetd

Если вам как и мне в приведенном выше примере повезло, то ничего больше настраивать не нужно. В противном случае запускаем суперсервер командой:

/sbin/service xinetd start

Starting xinetd: [OK]

А также делаем так, чтобы он всегда запускался при входе сервера в графический режим (уровень выполнения 5):

/sbin/chkconfig --level 5 xinetd on

Настройка TFTP

Для того, чтобы выполнить настройку тривиального протокола передачи файлов TFTP, практически не нужно ничего делать. Достаточно лишь запустить демон tftp и разрешить ему стартовать при запуске системы.

/sbin/chkconfig tftp on

На самом деле для работы с протоколом TFTP нужен демон xinetd, в конфигурационном каталоге которого /etc/xinetd.d/ должен быть файл tftp примерно такого содержания:

default: off

description: The tftp server serves files using the trivial file transfer \

```
#protocol.  The tftp protocol is often used to boot diskless \
#workstations, download configuration files to network-aware printers, \
#and to start the installation process for some operating systems.
service tftp
{
  disable        = no
  socket_type        = dgram
  protocol           = udp
  wait               = yes
  user               = root
  server             = /usr/sbin/in.tftpd
  server_args        = -s /tftpboot
  per_source         = 11
  cps                = 100 2
  flags              = IPv4
}
```

Теперь по запросу удаленного пользователя будет инициироваться сеанс загрузки ядра операционной системы Linux через локальную сеть и протокол TFTP.

Настройка NFS

Для настройки NFS нужно изменить параметры сразу нескольких файлов. Это, во-первых, файл /etc/exports, который определяет для NFS-сервера какие каталоги нужно экспортировать. Так, по умолчанию, проект LTSP предлагает экспортировать папки /opt/ltsp/i386 (только чтение) и /var/opt/ltsp/swapfiles (чтение/запись для использования файлов подкачки через локальную сеть и NFS). К содержимому файла /etc/exports дописывается следующая информация:

```
## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Пнд Лис  1 13:32:48 EET 2004 by the ltsp installation script.
```

For more information, visit the ltsp homepage
at http://www.ltsp.org
#

/opt/ltsp/i386 192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles
192.168.0.0/255.255.255.0(rw,no_root_squash,async)

#
The following entries need to be uncommented if you want
Local App support in ltsp
#
#/home 192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

LTS-end

Как видно из примера, экспортирование каталога /home не происходит, и может понадобиться только для поддержки и использования локальных приложений на X-терминалах. Если вы не будете использовать файлы подкачки через локальную сеть, то строку, содержащую /var/opt/ltsp/swapfiles тоже можно закомментировать. Так как мы используем подсеть с адресацией 192.1.1.0/255.255.255.0, то фрагмент файла /etc/exportsв нашей редакции будет выглядеть следующим образом:

LTS-begin

/opt/ltsp/i386 192.168.1.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles
192.168.1.0/255.255.255.0(rw,no_root_squash,async)
#/home 192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

LTS-end

После внесения изменений в файл /etc/exports необходимо, чтобы сервер NFS перечитал свою конфигурацию, для этого достаточно послать его демонам сигнал HUP:

```
# killall -HUP rpc.nfsd > /dev/null 2>&1
# killall -HUP rpc.mountd > /dev/null 2>&1
```

Или более радикально перезапустить демон:

```
# /sbin/service nfs restart
```

Также при настройке NFS сервера проверяется будут ли стартовать демоны nfs и portmap при переходе системы в графический режим (уровень выполнния 5). Для этого используются простые команды:

```
# /sbin/chkconfig --level 5 portmap on
# /sbin/chkconfig --level 5 nfs on
```

Настройка переключателей TCP

Если вы на сервере используете переключатели TCP (tcp wrappers), то для использования того же сервера, в качестве сервера X-терминалов необходимо внести изменения в файл /etc/hosts.allow, а именно:

```
#
# hosts.allow  This file describes the names of the hosts which are
#          allowed to use the local INET services, as decided
#          by the '/usr/sbin/tcpd' server.
#

## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Пнд Лис  1 13:32:48 EET 2004 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#
```

bootpd: 0.0.0.0

in.tftpd: 192.168.0.

portmap: 192.168.0.

LTS-end

Хотя в нашем конкретном случае эти изменения файла /etc/hosts.allow будут такими:

LTS-begin

bootpd: 0.0.0.0

in.tftpd: 192.168.1.

portmap: 192.168.1.

LTS-end

Настройка DNS

Для правильной работы X-терминалов необходимо, чтобы их доменные имена соответствовали реальным IP-адресам. Для этого проще всего перечислить их в файле /etc/hosts сервера. Например, он может быть такого содержания:

127.0.0.1 localhost.localdomain localhost

192.168.1.1 xtserver.xterms.net xtserver

192.168.0.26 xtserver.xterms.net xtserver

192.168.1.2 ws001.xterms.net ws001

192.168.1.3 ws002.xterms.net ws002

192.168.1.4 ws003.xterms.net ws003

В данном примере файла /etc/hosts, кроме сетевых интерфейсов сервера описаны также три X-терминала с доменными именами ws001, ws002 и ws003 и соответствующими им IP-адресами 192.168.1.2, 192.168.1.3 и 192.168.1.4.

Если этого не сделать, то при запуске X-терминалов могут возникнуть неожиданные ошибки в работе служб NFS и даже удаленного подключения к менеджеру дисплея.

Настройка X-Windows (Xaccess, Xsetup_workstation, xdm, kdm, gdm, xfs)

Иногда кажется, что графическая система X-Windows излишне запутанная и сложная в понимании и настройке. Чтобы развеять это поверхностное впечатление и показать гибкость этой системы, рассмотрим поэтапно процесс настройки XFree86 для работы в качестве сервера X-терминалов. X-терминалы могут работать с любым менеджером дисплея (xdm, kdm, gdm) и любым оконным менеджером (KDE, Gnome, IceWM, twm, fvwm, AfterStep, WindowMaker и т.п.), их выбор – дело вкуса и привычки системного администратора.

Настройка удаленного доступа к X-серверу (Xaccess)

По умолчанию предоставлять доступ к X-серверу извне не требуется. Но в случае построения сети X-терминалов нам нужно именно это. К счастью, проектом X-Windows разработан специальный протокол для удаленного доступа графических клиентов, называется он XDHCP (X Display Manager Control Protocol – протокол управления графическим менеджером дисплея). Проектом LTSP предлагается такое содержимое файла /etc/X11/xdm/Xaccess:

```
# $XConsortium: Xaccess,v 1.5 91/08/26 11:52:51 rws Exp $
#
# Access control file for XDMCP connections
#
# To control Direct and Broadcast access:
#
#pattern
#
# To control Indirect queries:
#
#       pattern                 list of hostnames and/or macros ...
#
# To use the chooser:
#
#pattern                 CHOOSER BROADCAST
```

```
#
# or
#
#pattern                CHOOSER list of hostnames and/or macros ...
#
# To define macros:
#
#       %name           list of hosts ...
#
# The first form tells xdm which displays to respond to itself.
# The second form tells xdm to forward indirect queries from hosts matching
# the specified pattern to the indicated list of hosts.
# The third form tells xdm to handle indirect queries using the chooser;
# the chooser is directed to send its own queries out via the broadcast
# address and display the results on the terminal.
# The fourth form is similar to the third, except instead of using the
# broadcast address, it sends DirectQuerys to each of the hosts in the list
#
# In all cases, xdm uses the first entry which matches the terminal;
# for IndirectQuery messages only entries with right hand sides can
# match, for Direct and Broadcast Query messages, only entries without
# right hand sides can match.
#

*                               #any host can get a login window

#
# To hardwire a specific terminal to a specific host, you can
# leave the terminal sending indirect queries to this host, and
# use an entry of the form:
#
```

```
#terminal-a    host-a


#
# The nicest way to run the chooser is to just ask it to broadcast
# requests to the network - that way new hosts show up automatically.
# Sometimes, however, the chooser can't figure out how to broadcast,
# so this may not work in all environments.
#

*        CHOOSER BROADCAST      #any indirect host can get a chooser

#
# If you'd prefer to configure the set of hosts each terminal sees,
# then just uncomment these lines (and comment the CHOOSER line above)
# and edit the %hostlist line as appropriate
#

#%hostlist     host-a host-b

#*               CHOOSER %hostlist     #
```

От общепринятой конфигурации файл /etc/X11/xdm/Xaccess отличается лишь незакомментированной строчкой, которая разрешает любому узлу подключаться к X-серверу и получать в ответ графический экран ввода имени и пароля пользователя. Следует помнить, что файл /etc/X11/xdm/Xaccess используется только менеджером дисплея XDM, если последний является менеджером дисплея по умолчанию.

Настройка экрана ввода имени пользователя и пароля (Xsetup_workstation)

Для того, чтобы пользователи X-терминалов знали, что они работают с продуктом проекта LTSP, графический экран ввода имени пользователя и

пароля имеет фон с логотипом LTSP (файл /etc/X11/xdm/ltsp.gif). Пример удаленного экрана входа в систему от проекта LTSP можно увидеть на рис. 6.4.

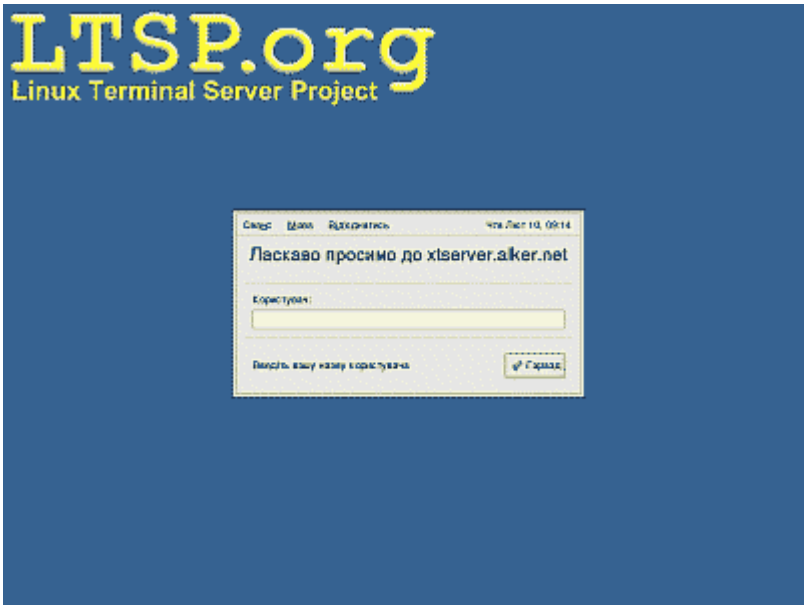

Рис. 6.4. Экран входа в систему LTSP

Сценарий для настройке экрана входа в систему содержится в файле /etc/X11/xdm/Xsetup_workstation и имеет такое содержание:

```
#!/bin/sh
## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Пнд Лис  1 13:32:48 EET 2004 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/usr/X11R6/bin/xsetroot -solid "#356390"
if [ -x /usr/bin/xsri ]; then
    /usr/bin/xsri -geometry +5+5 -avoid 300x250 -keep-aspect \
        /etc/X11/xdm/ltsp.gif
fi

## LTS-end ##
```

В принципе, если у вас есть фантазия и желание изменить внешний вид экран входа в систему, то можете использовать для этих целей любые цвета и графические изображения. Важно помнить, что в этом файле устанавливаются параметры стандартного сервера, а такие менеджеры дисплея как gdm и kdm используют параллельно свои собственные программы отображения экрана входа в систему.

Также не забудьте проверить права доступа к этому файлу, он должен быть исполняемым:

chmod 755 /etc/X11/xdm/Xsetup_workstation

Настройка диспетчера экрана XDM

Диспетчер экрана XMD – это старейший из себе подобных в рамках проекта X-Windows. Он существовал еще до появления сред Gnome и KDE, которые пришли со своими аналогичными разработками, поэтому начнем описание конфигурации именно с него.

Для настройки менеджера дисплея xdm проектом LTSP используется файл /etc/X11/xdm/xdm-config, содержимое которого представлено ниже:

```
! $XConsortium: xdm-conf.cpp /main/3 1996/01/15 15:17:26 gildea $
! $XFree86: xc/programs/xdm/config/xdm-conf.cpp,v 1.6 2000/01/31 19:33:43 dawes Exp $
DisplayManager.errorLogFile:    /var/log/xdm-errors
DisplayManager.pidFile:         /var/run/xdm-pid
DisplayManager.keyFile:         /etc/X11/xdm/xdm-keys
DisplayManager.servers:         /etc/X11/xdm/Xservers
DisplayManager.accessFile:      /etc/X11/xdm/Xaccess
DisplayManager.willing:         su nobody -c /etc/X11/xdm/Xwilling
! All displays should use authorization, but we cannot be sure
! X terminals will be configured that way, so by default
! use authorization only for local displays :0, :1, etc.
DisplayManager._0.authorize:    true
DisplayManager._1.authorize:    true
! The following three resources set up display :0 as the console.
```

```
DisplayManager._0.setup: /etc/X11/xdm/Xsetup_0
DisplayManager._0.startup:      /etc/X11/xdm/GiveConsole
DisplayManager._0.reset:  /etc/X11/xdm/TakeConsole
!
DisplayManager*resources:       /etc/X11/xdm/Xresources
DisplayManager*session:         /etc/X11/xdm/Xsession
DisplayManager*authComplain: false

! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
# DisplayManager.requestPort:   0
## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Пнд Лис  1 13:32:48 EET 2004 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

DisplayManager.*.setup:     /etc/X11/xdm/Xsetup_workstation

## LTS-end ##
```

Так как в файле Xaccess мы разрешили удаленное подключение всем рабочим станциям, то в файле xdm-config достаточно выбрать сценарий настройки экрана ввода имени пользователя и пароля для удаленных компьютеров, им будет описанный выше Xsetup_workstation.

Настройка менеджера дисплея KDM

Среда KDE использует свой менеджер дисплея kdm, который хоть и базируется на исходном коде xdm, все же имеет часть значительных отличий. Для того, чтобы использовать в среде X-терминалов менеджер дисплея kdm,

необходимо внести изменения в файл /etc/kde/kdm/kdmrc (на самом деле это символическая ссылка на файл /etc/X11/xdm/kdmrc). Так как размер этого файла довольно большой, то приведу лишь важный фрагмент:

```
...
[Xdmcp]
# Whether KDM should listen to XDMCP requests. Default is true.
Enable=true
# The UDP port KDM should listen on for XDMCP requests. Don't change the 177.
#Port=177
# File with the private keys of X-terminals. Required for XDM authentication.
# Default is /usr/share/config/kdm/kdmkeys
#KeyFile=
# XDMCP access control file in the usual XDM-Xaccess format.
# Default is /usr/share/config/kdm/Xaccess
# XXX i'm planning to absorb this file into kdmrc, but i'm not sure how to
# do this best.
Xaccess=/usr/share/config/kdm/Xaccess
# Number of seconds to wait for display to respond after the user has
# selected a host from the chooser. Default is 15.
#ChoiceTimeout=10
# Strip domain name from remote display names if it is equal to the local
# domain. Default is true
#RemoveDomainname=false
# Use the numeric IP address of the incoming connection instead of the
# host name. Use this on multihomed hosts. Default is false
#SourceAddress=true
# The program which is invoked to dynamically generate replies to XDMCP
# BroadcastQuery requests.
# By default no program is invoked and "Willing to manage" is sent.
Willing=/usr/share/config/kdm/Xwilling
```

...

Этот, как уже было сказано, довольно большой конфигурационный файл содержит важный раздел [Xdmcp], который отвечает за то, будет ли слушать и принимать менеджер дисплея kdm XDMCP-запросы от удаленных рабочих станций. В оригинальном файле /etc/kde/kdm/kdmrc достаточно поменять значение “Enable=false” на “Enable=true”, чтобы все заработало. В этом же файле для управления доступом к X-серверу используется файл Xaccess=/usr/share/config/kdm/Xaccess. На первый взгляд может показаться, что это новый файл, но на самом деле это символическая ссылка на уже известный нам /etc/X11/xdm/Xaccess, хотя проверить лишний раз не помешает:

$ ls -l /usr/share/config/kdm/Xaccess

lrwxrwxrwx 1 root root 21 Feb 15 17:28 /usr/share/config/kdm/Xaccess -> ../../X11/xdm/Xaccess

Настройка менеджера дисплея GDM

Менеджер дисплея gdm разработан в рамках проекта Gnome. Он не базируется на исходном коде xdm, и написан практически с нуля. Для того, чтобы настроить этот менеджер дисплея, вам нужно отредактировать файл /etc/X11/gdm/gdm.conf. Так, при настройке сервера X-терминалов в рамках проекта LTSP 3.0 предлагается такая его редакция (фрагмент файла):

```
...
# XDMCP is the protocol that allows remote login.  If you want to log into
# gdm remotely (I'd never turn this on on open network, use ssh for such
# remote usage that).  You can then run X with -query  to log in,
# or -indirect  to run a chooser.  Look for the 'Terminal' server
# type at the bottom of this config file.
[xdmcp]
# Distributions: Ship with this off.  It is never a safe thing to leave
# out on the net.  Alternatively you can set up /etc/hosts.allow and
# /etc/hosts.deny to only allow say local access.
Enable=true
# Honour indirect queries, we run a chooser for these, and then redirect
```

```
# the user to the chosen host.  Otherwise we just log the user in locally.
HonorIndirect=true
# Maximum pending requests
MaxPending=4
MaxPendingIndirect=4
# Maximum open XDMCP sessions at any point in time
MaxSessions=16
# Maximum wait times
MaxWait=15
MaxWaitIndirect=15
# How many times can a person log in from a single host.  Usually better to
# keep at 1 to fend off DoS attacks by running many logins from a single
# host
DisplaysPerHost=1
# The port.  177 is the standard port so better keep it that way
Port=177
# Willing script, none is shipped and by default we'll send
# hostname system id.  But if you supply something here, the
# output of this script will be sent as status of this host so that
# the chooser can display it.  You could for example send load,
# or mail details for some user, or some such.
Willing=/etc/X11/gdm/Xwilling
...
```

Как и в случае с менеджером дисплея kdm, в файле /etc/X11/gdm/gdm.conf нас в первую очередь интересует раздел [xdmcp]. Самое главное в этом разделе – разрешить обработку XDMCP-запросов (Enable=true). Также следует обратить внимание на еще два важных параметра этого раздела:

MaxSessions=16– определяет максимальное количество пользователей, которые могут подключаться к X-серверу через XDMCP-запросы. Этот параметр необходимо будет изменить при увеличении количества X-

терминалов в вашей локальной сети (в данном примере, таких X-терминалов не больше 16);

DisplaysPerHost=1– этот параметр определяет, сколько раз к менеджеру дисплея gdm можно подключаться с одного хоста (узла сети) одновременно. Сейчас он установлен в 1, из соображений безопасности и для борьбы с DoS атаками на ваш сервер, но не всегда это значение целесообразно. Например, при повторном подключении одного и того же X-терминала могут возникнуть проблемы, так как система посчитает, что пользователь уже подсоединен. Причиной такого поведения системы могут служит программы демоны, которые продолжают работать на сервере от имени пользователя даже после того, как последний давно выключил свой X-терминал. Лично я рекомендовал бы установить значение DisplaysPerHost равным 2, хотя можно работать и со значением по умолчанию.

Кроме ручного редактирования файла /etc/X11/gdm/gdm.conf можно воспользоваться также программой gdmsetup, которая в диалоговой форме позволяет настраивать основные параметры работы этого менеджера дисплея. Запускается эта программа так:

/usr/sbin/gdmsetup

Для того, чтобы при запуске менеджера дисплея gdm появлялся фон с логотипом LTSP, нужно заменить файл /etc/X11/gdm/Init/Default символической ссылкой на /etc/X11/ xdm/Xsetup_workstation. Делается это так:

ln -s ../../xdm/Xsetup_workstation /etc/X11/gdm/Init/Default

В принципе, в этот же файл можно поместить и другие команды, которые потребуются для инициализации менеджера дисплея gdm.

На рис. 6.5 вы можете увидеть графическую заставку экрана ввода имени пользователя и пароля, которую предоставляет менеджер дисплея gdm.

Рис. 6.5. Графический экран входа в систему менеджера дисплея GDM

Выбор менеджера дисплея по умолчанию

Для работы X-терминалов подойдет любой из описанных выше менеджеров дисплея (xdm, kdm или gdm), поэтому можете смело отдать предпочтение любому из них. Для того, чтобы правильно выбрать менеджер дисплея по умолчанию, следует посмотреть как устроен сценарий его запуска, который находится в файле /etc/X11/prefdm. Нас будут интересовать строки, которые отвечают за выбор менеджера дисплея:

```
preferred=
if [ -f /etc/sysconfig/desktop ]; then
  . /etc/sysconfig/desktop
  if [ "$DISPLAYMANAGER" = GNOME ]; then
          preferred=gdm
  elif [ "$DISPLAYMANAGER" = KDE ]; then
          preferred=kdm
  elif [ "$DISPLAYMANAGER" = XDM ]; then
         preferred=xdm
  fi
fi
```

Как видно из фрагмента этого файла, сценарий проверяет наличие на диске файла /etc/sysconfig/desktop и если такой файл существует, то он включается в текст данного сценария. Далее проверяется значение переменной

DISPLAYMANAGER и сравнивается с одним из значений: GNOME, KDE и XDM. Отсюда следует, что для установления менеджера дисплея по умолчанию достаточно создать файл /etc/sysconfig/desktop, который будет содержать одну единственную строку, например, DISPLAYMANAGER=KDE. Создать этот файл можно простой командой:

```
# echo  'DISPLAYMANAGER=KDE' > /etc/sysconfig/desktop
```

Настройка сервера шрифтов (xfs)

В начале этой главы мы оговорились, что лучше использовать для всех X-терминалов один сервер шрифтов. Тем самым мы достигнем единообразного представления документов на всех X-терминалах нашей локальной сети. Но для осуществления наших планов, не плохо было бы убедиться, что наш сервер шрифтов готов принимать запросы от удаленных клиентов. Проверяется это командой:

```
$ netstat -a | grep xfs
tcp        0      0 *:xfs                 *:*                     LISTEN
```

Если, как в нашем примере, сервер шрифтов слушает удаленные подключения, то значит все нормально и больше настраивать ничего не нужно. Если же вы получили отрицательный ответ, то смотрим файл конфигурации сервера шрифтов /etc/X11/fs/config:

```
#
# Default font server configuration file for Red Hat Linux
#

# allow a max of 10 clients to connect to this font server
client-limit = 10

# when a font server reaches its limit, start up a new one
clone-self = on

# alternate font servers for clients to use
#alternate-servers = foo:7101,bar:7102
```

```
# where to look for fonts
#
catalogue = /usr/share/fonts/CP1251/misc:unscaled,
  /usr/share/fonts/CP1251/misc,
  /usr/share/fonts/CP1251/75dpi:unscaled,
  /usr/share/fonts/CP1251/75dpi,
  /usr/share/fonts/KOI8-R/misc:unscaled,
  /usr/share/fonts/KOI8-R/misc,
  /usr/X11R6/lib/X11/fonts/misc:unscaled,
  /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
  /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
  /usr/X11R6/lib/X11/fonts/misc,
  /usr/X11R6/lib/X11/fonts/Type1,
  /usr/X11R6/lib/X11/fonts/Speedo,
  /usr/X11R6/lib/X11/fonts/cyrillic,
  /usr/X11R6/lib/X11/fonts/TTF,
  /usr/share/fonts/default/Type1,

# in 12 points, decipoints
default-point-size = 120

# 100 x 100 and 75 x 75
default-resolutions = 75,75,100,100

# use lazy loading on 16 bit (usually Asian) fonts
deferglyphs = 16

# how to log errors
use-syslog = on
```

```
# don't listen to TCP ports by default for security reasons
no-listen = tcp
```

Из-за соображений безопасности, по умолчанию сервер шрифтов не обслуживает удаленных клиентов. Для разрешения удаленных подключений достаточно закомментировать строчку “no-listen = tcp” и перезапустить сервер:

```
# /sbin/service xfs restart
```

В сервере шрифтов также существуют ограничения на количество одновременно обслуживаемых клиентов. Так, в приведенном выше листинге это ограничение равно 10 подключениям (параметр "client-limit = 10"). При необходимости можно увеличить или уменьшить значение параметра client-limit. Для одного сервера шрифтов не рекомендуется устанавливать значение client-limit больше 40.

Важно также помнить, что для правильной работы системы X-Windows со шрифтами типа TrueType необходимо загрузить специальный модуль (freetype) для сервера XFree версии 4.x. Ниже приведен соответствующий фрагмент файла /etc/X11/XF86Config:

```
Section "Module"
        Load  "GLcore"
        Load  "dbe"
        Load  "dri"
        Load  "extmod"
        Load  "fbdevhw"
        Load  "glx"
#        Load  "pex5"
        Load  "record"
#        Load  "xie"
        Load  "v4l"
        Load  "type1"
        Load  "freetype"
EndSection
```

Модуль "type1" также используется для работы X-сервера со шрифтами, но не TrueType, а Adobe PostScript Type 1.

Главный файл конфигурации сервера LTSP

Создатели проекта Linux Terminal Server Project довольно неплохо постарались и настройка работы всех X-терминалов практически сосредоточена в одном файле. Расположен этот файл в папке /opt/ltsp/i386/etc и называется lts.conf. Ниже представлена базовая структура этого файла сразу после установки пакетов LTSP.

```
#
# Config file for the Linux Terminal Server Project (www.ltsp.org)
#

[Default]
    SERVER          = 192.168.0.254
    XSERVER         = auto
  X_MOUSE_PROTOCOL   = "PS/2"
  X_MOUSE_DEVICE     = "/dev/psaux"
  X_MOUSE_RESOLUTION = 400
  X_MOUSE_BUTTONS    = 3
  USE_XFS          = N
  LOCAL_APPS       = N
    RUNLEVEL        = 5

#-------------------------------------------------------------------------------
#
# Example of specifying X settings for a workstation
#
[ws001]
      XSERVER          = auto
      LOCAL_APPS       = N
      USE_NFS_SWAP     = N
```

```
        SWAPFILE_SIZE      = 48m
        RUNLEVEL           = 5

[ws002]
        XSERVER            = XF86_SVGA
        LOCAL_APPS         = N
        USE_NFS_SWAP       = N
        SWAPFILE_SIZE      = 64m
        RUNLEVEL           = 3

#
# ws004 is my virtual workstation running in a VMware session
#
[ws004]
        DNS_SERVER         = 192.168.0.254
        XSERVER            = auto
        X4_BUSID           = "PCI:0:15:0"
        X_MODE_0           = 800x600
        LOCAL_APPS         = N
        USE_NFS_SWAP       = N
        SWAPFILE_SIZE      = 64m
        RUNLEVEL           = 5

#-------------------------------------------------------------------------------
#
# Example of a workstation configured to load some modules
#
#[ws001]
#    MODULE_01       = agpgart.o                # This is for i810 video
#    MODULE_02       = uart401.o
#    MODULE_03       = sb.o io=0x220 irq=5 dma=1
```

```
#    MODULE_04       = opl3.o

#-----------------------------------------------------------------------------
#
# Example of ws001 configured for local apps
#
#[ws001]
#    LOCAL_APPS      = Y
#    LOCAL_WM        = Y
#    NIS_DOMAIN      = ltsp
#    NIS_SERVER      = 192.168.0.254

#-----------------------------------------------------------------------------
#
# Example of a serial printer attached to /dev/ttyS1 on workstation ws001
#
#[ws001]
#    PRINTER_0_DEVICE   = /dev/ttyS1
#    PRINTER_0_TYPE     = S            # P-Parallel, S-Serial
#    PRINTER_0_PORT     = 9100         # tcp/ip port: defaults to 9100
#    PRINTER_0_SPEED    = 9600         # baud rate: defaults to 9600
#    PRINTER_0_FLOWCTRL = S            # Flow control: S-Software (XON/XOFF),
#                                      # H-Hardware (CTS/RTS)
#    PRINTER_0_PARITY   = N            # Parity: N-None, E-Even, O-Odd
#                                      # (defaults to 'N')
#    PRINTER_0_DATABITS = 8            # Databits: 5,6,7,8 (defaults to 8)
```

Большая часть параметров этого файла закомментирована и предназначена в первую очередь для ознакомления пользователей с возможными конфигурациями X-терминалов. Структура файла /opt/ltsp/i386/etc/ltsp.conf достаточно простая, каждая секция начинается со строки [<Название секции>]

и заканчивается началом новой секции или концом файла. В этом файле обязательно должна быть секция с именем Default (регистр значения не имеет) – это своего рода главная секция, параметры которой становятся базовыми для всех остальных подразделов.

Все остальные секции описывают отдельно взятые X-терминалы, причем название секции может быть записано одним из 3-х способов:

Доменное имя X-терминала, например [ws000]

IP-адрес рабочей станции, например [192.168.1.5]

MAC-адрес сетевого адаптера X-терминала, например [00:02:56:46:7F:AA]

Вы вольны использовать любой тип именования секций, но наиболее целесообразным является именование по значению HOSTNAME (доменное имя X-терминала), так как в этом случае вам придется вносить минимум изменений в файлы конфигурации при изменении топологии сети или замене сетевого адаптера на X-терминале. Не следует также забывать, что доменное имя компьютера человеку запомнить намного проще, чем, например, IP-адрес или особенно MAC-адрес сетевого адаптера (хотя системные администраторы люди со странностями и я ничуть не удивлюсь, если работать с IP-адресами вам проще чем с доменными эквивалентами).

При настройке серверной части нет необходимости разбираться в параметрах секций, посвященным X-терминалам, достаточно отредактировать только секцию по умолчанию (Default). Вот пример такой редакции:

```
[Default]
        SERVER            = 192.168.1.1
        XSERVER           = auto
        X_MOUSE_PROTOCOL   = "PS/2"
        X_MOUSE_DEVICE     = "/dev/psaux"
        X_MOUSE_RESOLUTION = 400
        X_MOUSE_BUTTONS    = 3
        XkbLayout         = "ru_UA"
        X_COLOR_DEPTH      = 16
        X_MODE_0          = 800x600
```

USE_XFS = Y

LOCAL_APPS = N

RUNLEVEL = 5

Небольшие пояснения к секции [Default]:

SERVER = 192.168.1.1 – указывает сервер по умолчанию;

XSERVER = auto – означает, что программные компоненты проекта LTSP сами будут выбирать какой из доступных X-серверов использовать;

X_MOUSE_PROTOCOL = "PS/2" – будем считать, что большинство X-терминалов имеют манипуляторы “мышь” с интерфейсом PS/2 без скроллера;

X_MOUSE_DEVICE = "/dev/psaux" – то же, но указывает на символьное устройство “мышь” с интерфейсом PS/2;

X_MOUSE_RESOLUTION = 400 – чувствительность “мышки” (чем больше значение, тем выше скорость перемещения курсора “мыши” по экрану);

X_MOUSE_BUTTONS = 3 - “мышка” имеет три кнопки;

XkbLayout = "ru_UA" – раскладка клавиатуры для X-терминалов (очень полезна для жителей Украины). Граждане России могут здесь поставить просто “ru”;

X_COLOR_DEPTH = 16 – глубина цвета экрана X-терминала по умолчанию равна 16 бит или HiColor с 65536 возможными цветами;

X_MODE_0 = 800x600 – разрешение экрана X-терминалов по умолчанию 800x600;

USE_XFS = Y – принудительное использование одного сервера шрифтов для всех X-терминалов;

LOCAL_APPS = N – не использовать локальные приложения для запуска их на X-терминалах;

RUNLEVEL = 5 – после старта, X-терминал автоматически переходит в графический режим для ввода имени пользователя и пароля.

Изменения вступают в силу сразу после сохранения файла /opt/ltsp/i386/etc/ltsp.conf (без перезапуска демонов или других действий).

Реконфигурация сервера при добавлении новых терминалов

Любой администратор сервера X-терминалов рано или поздно столкнется с проблемой изменения файлов конфигурации при добавлении новых X-терминалов или изменении настроек уже существующих. Как правило, все изменения должны вноситься в файлы конфигурации так, чтобы не останавливать работу сервера и не мешать работе других пользователей.

Рассмотрим простой случай добавления нового X-терминала. Предположим, что у нас имеется сеть из 7-ми X-терминалов, с именами ws001, ws002, ws003, ws004, ws005, ws006, ws007 и соответственно IP-адресами с 192.168.1.2 по 192.168.1.8 соответственно. Нужно добавить в сеть X-терминал с доменным именем ws008 и IP-адресом 192.168.1.9. Для этого редактируем файл /opt/ltsp/i386/etc/lts.conf и добавляем в него секцию вида:

```
[ws008]
    XSERVER          = auto
    LOCAL_APPS       = N
    USE_NFS_SWAP     = Y
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
```

Конечно, реальные значения вашего X-терминала могут значительно отличаться от приведенных выше.

После этого переходим к редактированию файла /etc/dhcpd.conf, в который добавляем раздел host, соответствующий параметрам сетевого адаптера X-терминала:

```
host ws008 {
    hardware ethernet    00:02:18:F9:CB:50;
    fixed-address        192.168.1.9;
    filename             "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

MAC-адрес сетевого адаптера X-терминала (00:02:18:F9:CB:50) взят для учебного примера, а в реальной жизни сюда придется подставить настоящее значение. После сохранения файла /etc/dhcpd.conf необходимо перезапустить демон dhcpd:

```
# /sbin/service dhcpd restart
Shutting down dhcpd:                    [  OK  ]
Starting dhcpd:                         [  OK  ]
```

Не забывайте также добавить новую запись в файл /etc/hosts, иначе могут возникнуть проблемы в определении соответствия доменного имени нового X-терминала и его IP-адреса. Ниже представлена новая редакция файла /etc/hosts:

```
127.0.0.1     localhost.localdomain localhost
192.168.1.1   xtserver.xterms.net   xtserver
192.168.0.26  xtserver.xterms.net   xtserver
192.168.0.22  gateway.xterms.net    gateway
192.168.1.2   ws001.xterms.net      ws001
192.168.1.3   ws002.xterms.net      ws002
192.168.1.4   ws003.xterms.net      ws003
192.168.1.5   ws004.xterms.net      ws004
192.168.1.6   ws005.xterms.net      ws005
192.168.1.7   ws006.xterms.net      ws006
192.168.1.8   ws007.xterms.net      ws007
192.168.1.9   ws008.xterms.net      ws008
```

Все. Новый X-терминал готов к работе, достаточно лишь подключить его к локальной сети и нажать на его системном блоке кнопку “Power”.

При добавлении нового X-терминала, когда нужно менять топологию сети (например, разбивать сеть X-терминалов на две подсети) следует в основном изменять настройки сервера DHCP и физически подключать X-терминалы к разным сегментам локальной сети. К примеру, к существующей сети X-терминалов 192.168.1.x добавится сеть 192.168.2.x. Если до этого в сети было 15 X-терминалов на одном сегменте, а вы решили добавить 16-й и разбить сеть на два сегмента, то правильнее будет распределить X-терминалы равномерно (по 8 штук) на каждую из подсетей.

Если на X-терминале вы столкнетесь с проблемой замены комплектующих, то в случае замены сетевой карты необходимо будет прописать новый MAC-адрес в файле /etc/dhcpd.conf и перезапустить демон dhcpd, а в случае замены

любой другой составляющей аппаратной части X-терминала (монитора, видеоадаптера, манипулятора “мышь”, клавиатуры, звуковой карты, принтера и т.п.) достаточно будет лишь внести соответствующие изменения в файл /opt/ltsp/i386/etc/lts.conf.

Проект LTSP был разработан для того, чтобы управление сервером X-терминалов было простым и наглядным. Поэтому, используя его на практике, вы должны по достоинству оценить эти преимущества, особенно при добавлении новых и изменении конфигурации существующих X-терминалов.

Не смотря на то, что авторы данного проекта постарались упростить процесс настройки сервера и сделать его почти автоматическим, вам как системному администратору ничего не мешает внести соответствующие изменения в работу любой из служб сервера. На настройку сервера LTSP необходимо смотреть как на шаблон базового сервера X-терминалов, который предлагается вам в качестве исходного примера. Понятно, что не существует одинаковых систем, и что хорошо для одной, может оказаться неприменимым в других условиях. Почему бы нам не взять от сервера LTSP все лучшее, а остальное переписать под свои потребности. Как знать, может у вас это получиться даже лучше. В любом случае стоит попробовать себя в роли системного администратора сети бездисковых рабочих станций.

Глава 7. Настройка терминалов

Будем считать, что к этому времени мы имеем уже в своих руках работоспособный сервер X-терминалов и сейчас самое время заняться настройкой самих бездисковых рабочих станций. Этот процесс достаточно прост, но только если вы знакомы со всеми его тонкостями. Для новичков в этом деле он сначала покажется достаточно сложным и запутанным, но при помощи небольшой инструкций к действию вы самостоятельно, этап за этапом, настроите ваш первый X-терминал.

Начать нужно с того, чтобы усвоить порядок загрузки X-терминала. После включения питания, бездисковый компьютер выполняет такие действия:

по завершению работы POST-теста начинает работать сетевой загрузчик;

загрузочный образ рассылает широковещательные DHCP-запросы в надежде получить ответ от DHCP-сервера;

когда ответ получен, сетевому адаптеру X-терминала присваивается IP-адрес и по протоколу TFTP с сервера выполняется загрузка ядра Linux;

ядро инициализирует систему, выделяет область оперативной памяти под виртуальный диск, монтирует по сети необходимые каталоги и создает, если нужно, файл подкачки;

после успешного запуска ядра начинают выполняться скрипты инициализации, результатом которых является перевод системы в графический режим (запуск X-сервера на X-терминале);

X-сервер с помощью протокола XDMCP подключается к серверу X-терминалов (точнее к менеджеру дисплея) и пользователь видит на экране приглашение для ввода имени пользователя и пароля.

Рассмотрение настройки X-терминала начнем с самого первого пункта – создания загрузочного образа для осуществления загрузки операционной системы по сети.

Выбор метода сетевой загрузки

X-терминал может загружаться с разных носителей информации (дискета, жесткий диск, CD-ROM, USB-flash накопитель, и т.д). Самый правильный способ загрузки – использование встроенных функций сетевых, карт таких как BootROM (другими словами ПЗУ удаленной загрузки) или PXE, но не всегда этот вариант является оптимальным. Для начала вполне подойдет загрузка X-терминала при помощи обычной дискеты, так как это самый простой способ создания загрузочного образа, так называемый быстрый старт в мир бездисковой сетевой загрузки. Приступим к практике.

Создание дискеты с загрузочным образом

Для того, чтобы правильно выбрать загрузочный образ, необходимо знать точное название чипа сетевой карты. Большой проблемы это не составляет, так как это название практически всегда видно на внешней стороне главной микросхемы сетевого адаптера, но иногда приходится срывать наклейки производителей или применять более изощренные способы. Например, у вас

нет возможности увидеть название чипа сетевой карты, а настроить X-терминал нужно в любом случае. На помощь вам придет система уникальной маркировки всех аппаратных компонентов персонального компьютера. Перегружаем компьютер и после появления информации об установленных IDE приводах нажимаем кнопку Pause (Break). На экране монитора вы должны увидеть нечто подобное:

PCI device listing ...

Bus No.	Device No.	Func No.	Vendor ID	Device ID	Device Class	IRQ
0	1	0	8086	7121	Display Controller	11
0	31	1	8086	2411	IDE Controller	14
0	31	2	8086	2412	Serial Bus Controller	9
0	31	5	8086	2415	Multimedia Device	5
1	3	0	10EC	8029	Network Controller	11
					ACPI Controller	9

Ищем строку “Network Controller” и определяем уникальный номер производителя данной сетевой карты Vendor ID = 10EC, а также уникальный номер самого устройства Device ID = 8029. Вооружившись этой информацией, идем в Интернет и ищем информацию о чипсете, который используется в этой сетевой карте. Прекрасно для этих целей подойдет сайт http://www.pcidatabase.com, где можно осуществить поиск как по коду производителя так и по уникальному номеру устройства. Поиск по производителю дал отличные результаты, теперь мы знаем, что сетевой адаптер

с идентификационным номером 10EC:8029 имеет чипсет от Realtek Semiconductor, который называется RTL8029 (см. рис. 7.1).

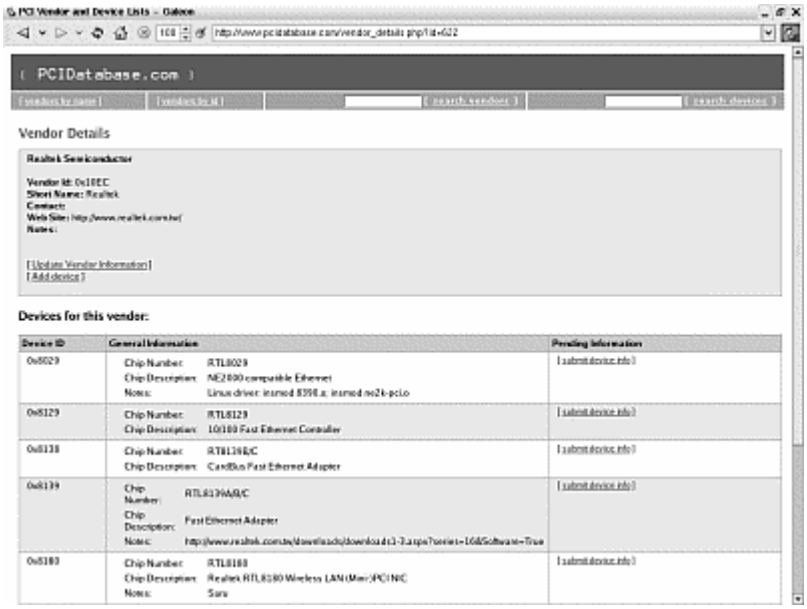

Рис. 7.1. Результат поиска чипсета по производителю

Когда номер чипа известен, нужно создать для него загрузочный образ, который потом записать на дискету. Существует специальный проект по созданию загрузочных образов для Intel x86 совместимых компьютеров – EtherBoot (официальный сайт http://www.etherboot.org/). В рамках этого проекта разрабатываются загрузочные образы практически для всех известных сетевых адаптеров. Ограниченность применения данного дистрибутива только компьютерами с набором команд x86 объясняется требованием к размеру загрузочного образа – в среднем около 16 Кбайт. Столь малый размер BootROM вынуждает авторов EtherBoot использовать при компиляции образов ассемблер, а как известно, этот язык непосредственно привязан к архитектуре процессора. В результате теряем переносимость, но выигрываем в размере загрузочного образа. Правда, слишком переживать по этому поводу не стоит, так как компьютеры с архитектурой x86 получили в мире наибольшее распространение, и, скорее всего, именно их вы будете использовать в качестве X-терминалов.

После установки пакета Etherboot вы будете иметь возможность самостоятельно создавать загрузочные образы для сетевых адаптеров с помощью нехитрой команды типа:

```
# make bin/rtl8029.zdsk
```

Но устанавливать, настраивать и компилировать пакет Etherboot не всегда удобно, особенно если речь идет об одном загрузочном образе для выбранной сетевой карты. К счастью, существует он-лайн, ресурс на котором вы можете получить уже скомпилированные готовые загрузочные образы. Называется этот сайт http://www.rom-o-matic.net (см. рис. 7.2).

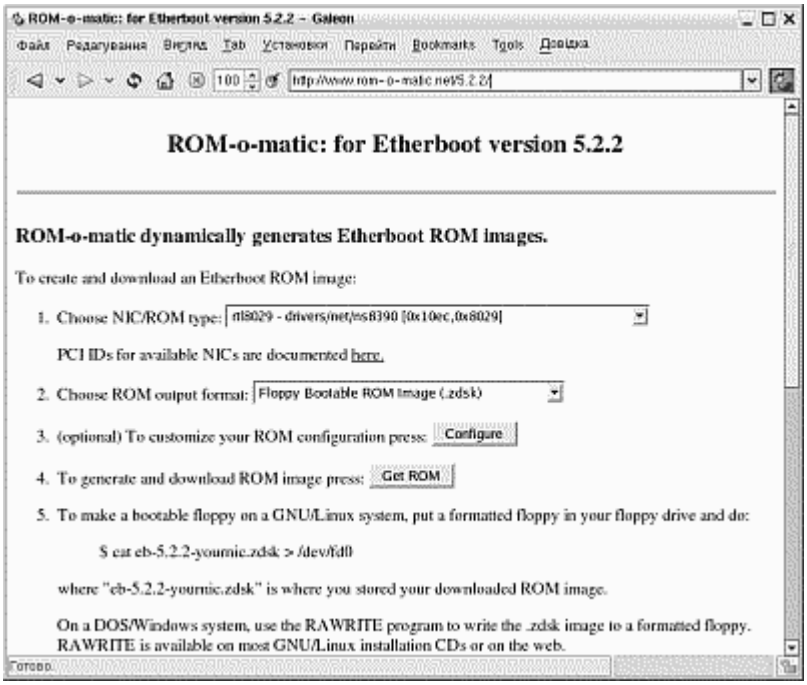

Рис. 7.2. Пример заполнения формы сайта www.rom-o-matic.net для получения загрузочного образа сетевого адаптера rtl8029

На сайте www.rom-o-matic.net собраны уже скомпилированные загрузочные образы для всех доступный сетевых карт. Следует отметить, что этот сайт довольно неплохо обновляется и с выходом новой версии пакеты Etherboot на нем появляется соответствующий раздел. Так, на момент написания книги наиболее новой стабильной версией была 5.2.5, а тестируемым пакетом был 5.3.9. Для того чтобы скачать загрузочный образ для дискеты, необходимо перейти по ссылке самой новой стабильной версии и из списка “Choose NIC/ROM type” выбрать соответствующий тип сетевой карты (для версии 5.2.5 и сетевой карты rtl8029 – это ns8390:rtl8029 – [0x10EC,0x8029]). Далее выбираем “Choose ROM output format” и устанавливаем “Floppy bootable ROM image (.zdsk). Теперь для получения образа достаточно нажать на кнопку “Get ROM”.

При желании вы можете прямо на сайте выбрать специфические параметры созданного образа сетевой загрузки, для этого существует кнопка

“Configure”, но если вы точно не уверены, что это необходимо, то лучше не менять параметры по умолчанию.

В результате правильного выполнения описанной выше последовательности действий вы должны были получить образ с именем eb-5.2.5-ns8390.zdsk и размером 16 Кбайт.

Чтобы переписать его на дискету, нужно взять чистую отформатированную дискету, поместить ее в дисковод вашего компьютера и выполнить команду от имени супер-пользователя root:

cat eb-5.2.5-ns8390.zdsk > /dev/fd0

Через несколько секунд загрузочная дискета будет создана.

Если вам по каким-либо причинам недоступен компьютер с Linux, то записать образ можно и в операционных системах типа DOS/Windows, но для этого нужно использовать программу RAWRITE, которую легко найти в Интернет.

Создание дискеты с загрузочным образом (продолжение)

Теперь необходимо сделать так, чтобы X-терминал загружался с созданной дискеты. Для этого при перезагрузке X-терминала заходим в программу конфигурирования BIOS и устанавливаем порядок загрузки (boot order) таким образом, чтобы на первом месте находился флоппи дисковод. Сохраняем изменения, вставляем дискету в дисковод X-терминала и снова перезагружаем бездисковый компьютер. После выполнения POST-теста на экране вы должны увидеть нечто подобное:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Похожая на выше приведенную запись свидетельствует о правильной загрузке образа. Смело нажимайте на клавишу N или Enter для начала загрузки по сети, хотя спустя 3 секунды загрузка должна начаться автоматически.

Так как мы еще не выбрали правильные параметры для DHCP-сервера, то на экране своего монитора вы должны увидеть нечто похожее на:

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:44:2B:02:4E

Searching for server (DHCP) ...

....

Появление похожей записи означает, что образ для сетевого адаптера выбран правильный и мы теперь даже знаем его MAC-адрес (00:02:44:2B:02:4E), но ответ от DHCP-сервера не получен и идет его активный поиск. Сейчас самое время настроить DHCP-сервер, а точнее добавить запись о тестируемом X-терминале. Для этого в файле /etc/dhcpd.conf необходимо вставить такие строчки:

```
host yanax {
    hardware ethernet     00:02:44:2B:02:4E;
    fixed-address         192.168.1.6;
    filename              "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

В описанном выше примере X-терминал называется yanax, при загрузке ему будет присвоен IP-адрес 192.168.1.6, и загружать по сети он будет ядро операционной системы Linux, что расположено в файле /lts/vmlinuz-2.4.24-ltsp-1 (как вы уже знаете, этот путь не абсолютный, а относительный согласно настроек демона TFTP).

После внесения изменений в файл /etc/dhcpd.conf не забудьте обязательно перезапустить демон DHCP:

/sbin/service dhcpd restart

Теперь включайте X-терминал и следите за проявляющимися сообщениями на экране. После приглашения выбрать загрузку по сети “Boot from (N)etwork or (Q)uit?” на мониторе должы появиться надписи:

..Me: 192.168.1.6, Server: 192.168.1.1, Gateway 192.168.1.1

Loading 192.168.1.1:/lts/vmlinuz-2.4.24-ltsp-1 ... (NBI) .....

.....

Если вы увидели на своем экране нечто подобное, это означает что вы стали свидетелями удаленной загрузки через локальную сеть.

Но не все так безоблачно. Иногда при загрузке с дискеты возникают и неожиданные проблемы. Например, если записать на дискету образ, который не соответствует сетевому адаптеру, то на экране вы увидите нечто похожее на:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.5. (GPL) http://etherboot.org Tagged ELF for [W89C840F]

Relocation _text from [00013e20, 00024230) to [01eefb70, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

Probing isa nic ...

...

В приведенном выше примере, для X-терминала с сетевой картой rtl8029 мы использовали образ загрузки от сетевого адаптера с чипом Winbond (w89c840f). Образ загрузки успешно был прочитан с дискеты, но вот инициализировать сетевую карту он не смог, так как не соответствует ей. Про неверно выбранный образ для загрузки свидетельствуют строки:

Probing pci nic ...

Probing isa nic ...

...

В загрузке с дискеты нет ничего предосудительного. Конечно, она выполняется на несколько секунд медленнее, требует наличия в X-терминале флоппи-дисковода, и иногда дискеты портятся от времени, так что образ быть может придется записывать заново. Зато у этого способа есть и ряд плюсов: дискеты стоят дешево, легко доступны и для их записи не требуется ничего кроме дисковода (для прошивки микросхем BootROM требуется программатор), при замене сетевой карты ничего не стоит переписать на

загрузочную дискету новый соответствующий образ для сетевой удаленной загрузки. В общем, рекомендуется использовать загрузку с дискеты в тестовых целях и при начальной настройке работы X-терминалов, а уже при последующей эксплуатации перейти к более удобному способу, такому как загрузка через BootROM сетевого адаптера или спецификацию PXE.

Определения MAC-адреса сетевой карты

MAC-адрес или физический адрес сетевого адаптера однозначно идентифицирует сетевое устройство. В мире нет двух устройств с одинаковым MAC-адресом (так категорично утверждать нельзя, скажем, не должно быть двух устройств с одним и тем же физическим адресом). MAC-адрес состоит из 6-байт, каждый из которых принято записывать в шестнадцатеричной форме через двоеточие или дефис. Если вы никогда раньше не сталкивались с физическими адресами, то в этом нет ничего страшного. Единственное, что вам следует запомнить, это то, что MAC-адрес – это уникальное число, которое у каждой сетевой карты разное и зависит от производителя.

Когда вы подберете правильный образ для записи на дискету, при инициализации его после загрузки на экране вы увидите правильный MAC-адрес сетевого адаптера (00:02:44:2B:02:4E):

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:44:2B:02:4E

Searching for server (DHCP) ...

Узнать этот адрес другим путем (без использования специальных утилит у вас вряд ли получится). Как правило, сетевые адаптеры комплектуются системными программами, которые позволяют не только узнать MAC-адрес сетевой карты, но и установить другие параметры. Минусом этих утилит

является то, что для своей работы они требуют установленную операционную систему (как правило, MS DOS или Windows), а для X-терминала это не приемлемо.

Если сетевая карта была установлена в компьютер с системой win32, то проще всего узнать ее MAC-адрес можно, если выполнить команду:

> ipconfig /all

Настройка протокола IP для Windows

Имя компьютера . . . . . : over-book
Основной DNS-суффикс . . . . . :
Тип узла. . . . . : неизвестный
IP-маршрутизация включена . . . . : нет
WINS-прокси включен . . . . . : нет

Подключение по локальной сети - Ethernet адаптер:

DNS-суффикс этого подключения . . :
Описание . . . . . : Realtek RTL8139 Family PCI Fast Ethernet NIC
Физический адрес. . . . . : 00-02-3F-78-B3-F8
Dhcp включен. . . . . : нет
IP-адрес . . . . . : 192.168.0.6
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : 192.168.0.22
DNS-серверы . . . . . : 194.44.39.130
194.44.39.140

Для персонального компьютера с операционной системой Linux это также просто, нужно лишь использовать команду:

$ /sbin/ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:00:21:2B:0C:F5

inet addr:192.168.0.14 Bcast:192.168.0.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:209353 errors:0 dropped:0 overruns:0 frame:0

TX packets:158989 errors:0 dropped:0 overruns:0 carrier:0

collisions:18112 txqueuelen:100

RX bytes:140380907 (133.8 Mb) TX bytes:23047256 (21.9 Mb)

Interrupt:11 Base address:0xc000

Или другую команду:

$ dmesg | grep eth

divert: allocating divert_blk for eth0

eth0: RealTek RTL-8029 found at 0xc000, IRQ 11, 00:00:21:2B:0C:F5.

В любом случае, для того чтобы узнать MAC-адрес сетевой карты, необходимо сначала ее инициализировать при помощи драйвера, специальной утилиты или загрузочного образа проекта Etherboot. Для сетевых карт совместимых со спецификацией PXE достаточно лишь активировать этот метод загрузки и удостоверится, что компьютер подключен к локальной сети (требуется наличие сетевого кабеля в гнезде сетевого адаптера).

Загрузка X-терминала с других носителей информации(lilo, cd-rom, msdos)

Иногда нужно выполнить загрузку не с дискеты, а скажем с жесткого диска или CD-ROM. В таких случаях описанная выше методика использования дискет не подойдет. Для этих целей рассмотрим ряд специфических приемов, которые помогут вам запустить удаленную загрузку с компьютера, на котором установлена операционная система Linux, с загрузочного компакт-диска и из сеанса MS DOS.

Загрузка при помощи(LInux LOader)

Начнем объяснение издалека, с ответа на вопрос - “Зачем это нужно?”. Например, я на работе использую персональный компьютер с операционной системой Linux, но иногда для администрирования и выполнения других производственных функций я пользуюсь им как X-терминалом. Для того, чтобы совместить эти два компьютера в одном, я при загрузке могу выбирать: загружать мне компьютер как X-терминал или как обычную традиционную

рабочую станцию Linux. Чтобы сделать это возможным, следует немного реконфигурировать загрузчик LILO и добавить к нему специфический загрузочный образ.

Оригинальный файл настройки LILO /etc/lilo.conf на моем компьютере выглядит так:

```
$ cat /etc/lilo.conf
disk=/dev/hda
bios=128
disk=/dev/hdd
bios=129

boot=/dev/hda
prompt
timeout=300
lba32
default=linux-2.4.20

# ASPLinux
image=/boot/vmlinuz-2.4.20-9asp
initrd=/boot/initrd.2.4.20-9asp.img
label=linux-2.4.20
root=/dev/hda5
read-only
append="hdc=ide-scsi"


# MS DOS
other=/dev/hdd1
label=dos1
```

Как видно из приведенной распечатки, сейчас у загрузчика LILO две загрузочных записи. Необходимо добавить третью с образом сетевой загрузки.

Для ее создания необходимо иметь специальный загрузочный образ, предназначенный для загрузчика LILO. На сайте http://www.rom-o-matic.net (см. рис. 7.3) в поле “Choose ROM output format:” следует выбрать “LILO/GRUB/SYSLINUX loadable kernel format (.zlilo)”.

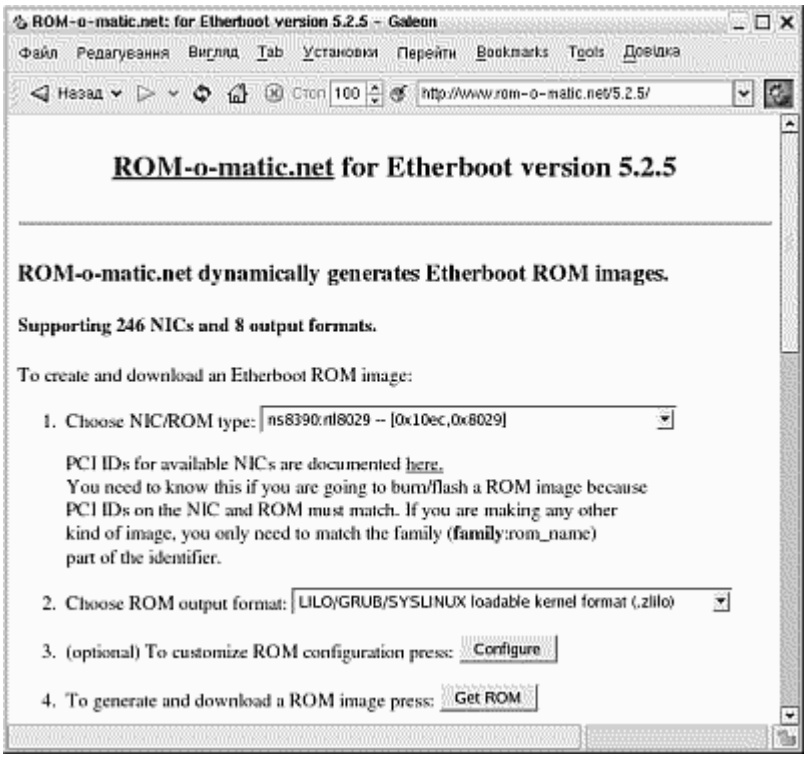

Рис.7.3. Выбор образа для загрузчика LILO сетевого адаптера rtl8029

Полученный файл eb-5.2.5-ns8390.zlilo сохраняем в каталоге /boot под любым именем, например, eb8029.zlilo:

```
# cp eb-5.2.5-ns8390.zlilo /boot/eb8029.zlilo
```

Добавляем в /etc/lilo.conf соответствующий раздел:

```
# Etherboot image
image=/boot/eb8029.zlilo
label=etherboot
```

И обновляем MBR жесткого диска командой:

```
# /sbin/lilo
Added linux-2.4.20 *
Added dos1
Added etherboot
```

Теперь перегружаем компьютер и в приглашении LILO выбираем третий пункт “etherboot”. Если загрузочный образ выбран правильно и он

соответствует сетевому адаптеру, мы получаем загрузку ПК в режиме X-терминала.

Если в режиме X-терминала вы работаете чаще чем в режиме обычного компьютера, то можно сделать этот образ загружаемым по умолчанию. Для этого достаточно исправить “default=linux-2.4.20” на “default=etherboot” и еще раз обновить Master Boot Record (MBR) системного диска командой lilo.

Создание загрузочного компакт диска

В редких случаях может потребоваться загрузка с компакт диска. Так, например, я встречал ноутбуки, в которых не было флоппи-дисковода, а сетевая карта была интегрированная или типа PCMCIA, что не позволяло использовать ее для удаленной загрузки. Единственный способ использовать такой портативный компьютер для работы в качестве X-терминала (зачем, это уже другой вопрос, например, не выбрасывать же старый портативный ноутбук с 468-м процессором или Pentium) – это создать системный компакт диск. Делается это так (пример для сетевой карты с чипсетом rtl8029):

Для начала создаем загрузочный флоппи диск и, если есть возможность, проверяем его на работоспособность (загружаем X-терминал с него):

```
# cat eb-5.2.5-ns8390.zdsk > /dev/fd0
```

Далее создаем папку для хранения образа загрузки и делаем ее активной:

```
$ mkdir ForRecord
$ cd ForRecord/
```

Копируем дискету в образ загрузки:

```
$ dd if=/dev/fd0 of=boot.img
```

Создаем загрузочный ISO-образ для последующей записи на компакт-диск:

```
$ mkisofs -b boot.img -o image.iso .
```

Записываем полученный образ на компакт-диск, который собираемся потом использовать для загрузки (подойдет как CD-R так и CD-RW диск). В данном примере для записи компакт-диска используется утилита cdrecord, но для записи стандартного образа ISO9660 подойдет и любая другая программа. В случае использования программы cdrecord, перед использованием аргумента параметра -dev, необходимо выполнить ее с опцией -scanbus:

```
# cdrecord -scanbus
Cdrecord 2.01a04 (i686-pc-linux-gnu) Copyright (C) 1995-2002 Jцrg Schilling
Linux sg driver version: 3.1.24
Using libscg version 'schily-0.7'
scsibus0:
    0,0,0     0) 'ASUS    ' 'CRW-5232AS      ' '1.0 ' Removable CD-ROM
    0,1,0     1) *
    0,2,0     2) *
    0,3,0     3) *
    0,4,0     4) *
    0,5,0     5) *
    0,6,0     6) *
    0,7,0     7) *
```

Теперь записываем образ на компакт-диск:

```
$ cdrecord -dev=0,0,0 -v -eject image.iso
Cdrecord 2.01a04 (i686-pc-linux-gnu) Copyright (C) 1995-2002 Jцrg Schilling
TOC Type: 1 = CD-ROM
...
Device seems to be: Generic mmc CD-RW.
Using generic SCSI-3/mmc CD-R driver (mmc_cdr).
Driver flags   : MMC-3 SWABAUDIO BURNFREE
Supported modes: TAO PACKET SAO SAO/R96P SAO/R96R RAW/R16 RAW/R96P RAW/R96R
Drive buf size : 1951488 = 1905 KB
FIFO size      : 4194304 = 4096 KB
Track 01: data     1 MB
Total size:        2 MB (00:11.97) = 898 sectors
Lout start:        2 MB (00:13/73) = 898 sectors
Current Secsize: 2048
...
BURN-Free is ON.
```

Turning BURN-Free off

Performing OPC...

Starting new track at sector: 0

Track 01: 1 of 1 MB written (fifo 100%) [buf 76%] 166.5x.

Track 01: Total bytes read/written: 1835008/1835008 (896 sectors).

Writing time: 5.141s

Average write speed 2.3x.

Fixating...

Fixating time: 79.148s

cdrecord: fifo had 29 puts and 29 gets.

cdrecord: fifo was 0 times empty and 0 times full, min fill was 100%

Теперь проверяем, что у нас получилось: перезагружаем X-терминал и не забываем установить в BIOS загрузку с CD-ROM. Если вы сделали все правильно, то на экране должны увидеть нечто похожее:

Verifying DMI Pool Data . . .

Boot from CD:

1. FD 1.44 Mb System Type-(0F)

Loading ROM image .....

В принципе, на загрузочный компакт диск можно дописать много другой полезной информации, так как свободного места там остается предостаточно. Единственная проблема использования такого гибридного компакт-диска заключается в том, что этот диск вам всегда будет нужен для начальной загрузки X-терминала.

Загрузка в режиме MS DOS

Не знаю, когда может понадобиться такая возможность, но существует способ загружать X-терминал из среды MS DOS. Для этого вам понадобится специальный образ формата “DOS Executable ROM Image (.com)” для вашего сетевого адаптера. Выбор нужного образа загрузки для сетевой карты с чипсетом rtl8029 приведен на рисунке 7.4, это файл eb-5.2.5-ns8390.com.

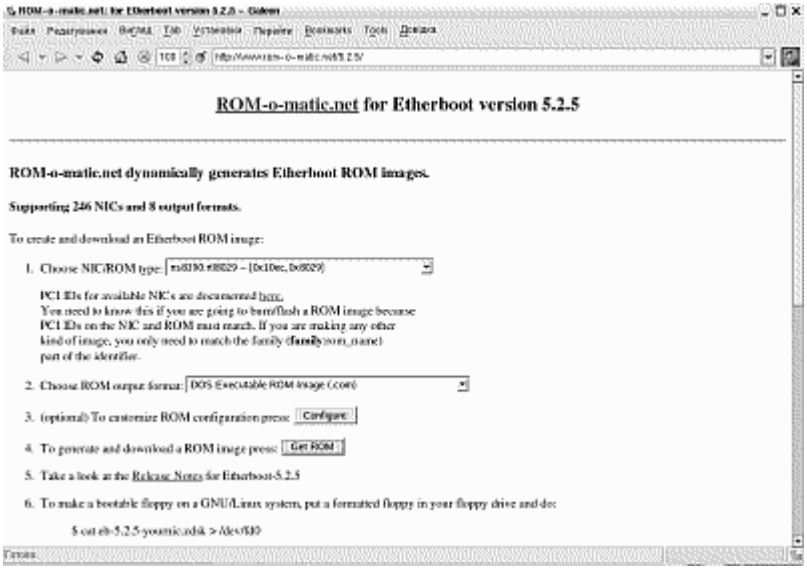

Рис. 7.4. Выбор образа для операционной системы MS DOS и сетевого адаптера rtl8029

Теперь полученный образ переносим на носитель, доступный операционной системе MS DOS и, либо указываем его как автоматически запускаемый при старте, либо будем запускать его вручную. Для проверки вполне подойдет загрузочная дискета с MS DOS или MS Windows.

Использование сетевых карт с BootROM или PXE

BootROM

Самой правильной и удобной является загрузка штатными средствами сетевой карты. Для этого ваш сетевой адаптер должен обладать возможностью загрузки по сети. Наибольшее распространение получили сетевые карты с BootROM – специальным отсеком для установки загрузочной микросхемы. В этой микросхеме типа EPROM или EEPROM записывается уже известный нам код Etherboot, но в формате “Binary ROM Image (.zrom)”, который и осуществляет удаленную загрузку X-терминала (см. рис. 7.5).

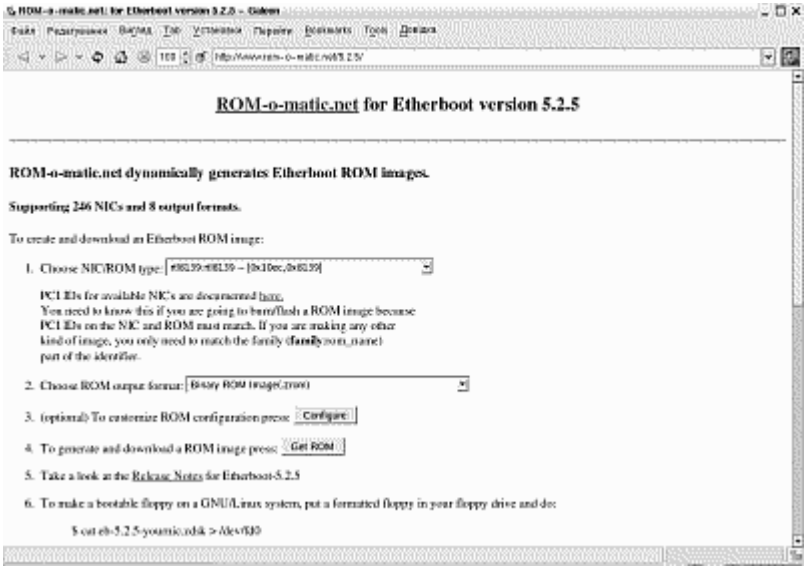

Рис. 7.5. Выбор образа BootROM для сетевого адаптера с чипсетом rtl8139

К сожалению, только некоторые сетевые карты обладают средствами самостоятельной записи кода в микросхему BootROM, да и то только при наличии специальной утилиты от производителя. Все остальные могут только считывать из этой микросхемы данные, а записывать их вам придется в другом месте с помощью специального прибора - программатора. Описание принципа действия программатора и методика работы с ним выходят за рамки данной книги. Важно лишь отметить, что этот прибор есть практически в каждой компьютерной фирме, которая занимается ремонтом и обслуживанием компьютерной техники и за умеренную плату всегда можно договориться о записи нескольких микросхем. Нужно только иметь сами микросхемы и образы для прошивки.

Как правильно выбрать микросхему для установки в гнездо BootROM? Для начала нужно определить размер загрузочного образа. Он, как правило, не превышает 16 кбайт, но бывают исключения. Далее отправляйтесь в ближайший магазин радио-деталей за микросхемой с маркировкой:

27CXXX, где

27– обязательный префикс маркировки микросхемы;

C– указывает на одноразовость микросхемы (можно записывать только один раз), поэтому можно покупать и без буквы “C” ("многоразовая" микросхема будет стоить, как правило, дороже);

XXXX– числовое значение емкости в килобитах, бывает 64 (8 Кбайт), 128 (16 Кбайт), 256 (32 Кбайт), 512 (64 Кбайт) и т.д.

При маркировании микросхем некоторые производители используют свои собственные правила. Так перед числом 27 вполне могут идти малозначащие буквы, и даже буква "C" в названии не всегда указывает на одноразовый характер чипа. Так, в моей практике был случай, когда я купил многоразовую микросхему BootROM с маркировкой 27C512F-12F1. Поэтому прежде чем купить ту или иную микросхему лучше дополнительно проконсультироваться в магазине радиотоваров или в глобальной сети Интернет.

Важно помнить, что емкость EPROM микросхемы может быть больше чем загрузочный образ, но не должна быть меньше. Так, образ eb-5.2.2-rtl8139.zrom размером 16384 байт можно записать на микросхемы типа 27C128, 27C256 и 27512, но нельзя на 27C64.

Пример микросхемы BootROM установленной в сетевой адаптер вы можете увидеть на рис. 7.6:

Рис. 7.6. Установленная микросхема BootROM

После прошивки образа, содержащую его микросхему необходимо поместить в разъем BootROM сетевого адаптера. Для правильной установки на сетевой карте и микросхеме имеется специальная пометка – полукруглая выемка (см. рис. 7.7). Теперь вставляем сетевой адаптер в компьютер и переходим к настройке BIOS.

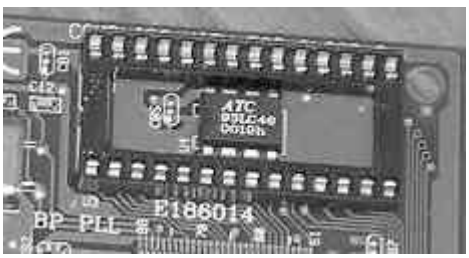

Рис. 7.7. Сетевая карта с пустым гнездом BootROM

В большом количестве компьютеров выполнять дополнительную настройку BIOS нет необходимости, так как их программа инициализации

рассчитана на сетевую загрузку как самою приоритетную, но иногда приходится повозиться.

Так, в конфигурации BIOS X-терминала вам следует искать пункты, отвечающие за порядок (приоритет) загрузки и использование для этого сетевых устройств. В частности, следует обратить внимание на разделы “Boot from LAN first”, “Network boot”, “Boot first = Network” и похожие на них.

Помню, как-то попал мне в руки компьютер с материнской платой от MSI и довольно нестандартной программой конфигурирования BIOS. Так вот, там был раздел, отвечающий за режим работы монитора при выполнении загрузки, ну а я по невнимательности его пропустил. В результате, при настройке X-терминала я после включения компьютера получал абсолютно черный экран на жидкокристаллическом мониторе Samsung 152N (ну не поддерживает этот монитор режим CGA). Тогда я думал, что X-терминал просто виснет при загрузке и стал менять его комплектующие: сетевые карты, видеоадаптер и даже флоппи-дисковод. Как вы сами понимаете ничего не помогало. После двух дней мучений, я наконец поставил в этот X-терминал сетевую карту от уже настроенного X-терминала (то есть ее MAC-адрес уже был правильно занесен в файл /etc/dhcpd.conf). И что вы думаете, после 3-х секунд того же черного экрана я увидел бегущий текст загрузки ядра Linux. Вот тогда я и догадался об истинных причинах проблем с загрузкой. Во всех предыдущий случаях на экране появлялась надпись “Searching for server (DHCP) ....”, но так как монитор не мог работать в выбранном режиме, то естественно, экран оставался черным. При работе же с настроенной сетевой картой, после инициализации ядра, операционная система Linux переключала режим монитора на стандартный.

К сожалению, вставкой в компьютер сетевой карты с правильно записанным BootROM, не всегда заканчивается настройка аппаратной части X-терминала. Так, на практике, мне приходилось встречаться с такими компьютерами, у которых запустить X-терминал можно было только в случае установки сетевой карты в определенный PCI-слот. Бывало доходило до анекдотической ситуации, когда X-терминал на базе старенького Pentium в AT-

корпусе прекрасно с первого раза загружался по сети, а тот же сетевой адаптер в X-терминале с более современным Celeron и на более новой материнской плате в ATX-корпусе напрочь отказывался понимать сетевую загрузку. Но после небольших манипуляций с комплектующими и настройкой конфигурации BIOS вам всегда удастся настроить загрузку X-терминала через локальную сеть. Если же вы столкнетесь с подобными проблемами, то для их решения можно попробовать обнулить CMOS память или сбросить все установки по умолчанию, а уже после этого перейти к настройке сетевого запуска X-терминала. Не стоит также забывать и о альтернативных методах загрузки. Например, если у вас есть древний привод CD-ROM, то его вполне можно использовать в качестве загрузочного устройства, к тому же иногда это единственной способ закрыть 5,25” отверстие в корпусе типа desktop. Если загрузка по сети методом BootROM не работает можно еще попробовать обновить BIOS до самой последней доступной версии, иногда в новой редакции BIOS исправляются подобные ошибки.

Использование сетевых адаптеров PXE

PXE (Pre-boot eXecution Environment – предзагрузочная среда выполнения) - это спецификация фирмы Intel для выполнения удаленной загрузки. Сетевые карты с функцией PXE стоят, как правило, дороже и для их настройки в X-терминалах следует использовать несколько другие приемы. Сетевая загрузка при помощи PXE, кроме довольно дорогих сетевых адаптеров от Intel и 3Com, свойственна также встроенным в материнские платы устройствам. Скорее всего это связано с тем, что в полнофункциональном устройстве (материнской плате) легче реализовать удаленную загрузку через PXE-метод.

Признаюсь честно, мне долго не могла попасться в руки PXE-сетевая карта, но в один прекрасный день у меня на столе оказался системный блок с материнской картой ASUS P4P800-VM, сразу скажу, не очень подходящий выбор для X-терминала, но эта материнская плата имела интегрированную сетевую карту с чипсетом Intel 82562EZ, который поддерживал спецификацию PXE, а следовательно, мог быть использован для проверки этого метода удаленного запуска.

Первое, что необходимо сделать при работе с сетевой картой PXE – это включить функцию удаленной загрузки в BIOS. А также правильно установить приоритет загрузки через сеть. Для материнской платы ASUS P4P800-VM я так и сделал, и после перегрузки компьютера увидел на экране такое сообщение:

Intel ® Boot Agent FE v4.1.10

Copyright © 1997-2003, Intel Corporation

Initializing and establishing link...

PXE-E61: Media test failure, check cable

PXE-M0F: Exiting Intel Boot Agent

Reboot and Select proper Boot device

or Insert Boot Media in selected Boot device and press a key

В процессе самодиагностики было выявлено отсутствие сетевого кабеля, о чем PXE-загрузчик и сообщил на экране. Осознав свою оплошность, я подключил сетевой кабель и нажал как было написано любую клавишу. Теперь на экране появились такие строки:

Intel ® Boot Agent FE v4.1.10

Copyright © 1997-2003, Intel Corporation

CLIENT MAC ADDR: 00 11 2F CC 69 89 GUID: 848FC32 391F 8F07 66E5E96DF98A

DHCP...

После аббревиатуры DHCP стали появляться точки на экране, что явно свидетельствовало о рассылки DHCP-запросов. Через несколько секунд сообщение на экране сменилось таким:

PXE-E51: No DHCP or proxyDHCP offers were received

PXE-M0F: Exiting Intel Boot Agent

Reboot and Select proper Boot device

or Insert Boot Media in selected Boot device and press a key

Понятно, что сетевая карта не получила ответ на свой DHCP-запрос, так как мы не описали ее в конфигурационном файле DHCP-сервера. Зная MAC-адрес (00:11:2F:CC:69:89) PXE-сетевой, карты необходимо добавить информацию о ней в соответствующем разделе файла /etc/dhcpd.conf, например:

```
host pxeadmin {
    hardware ethernet     00:11:2F:CC:69:89;
    fixed-address         192.168.1.14;
    filename              "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

Обратите внимание, что параметр filename я специально оставил прежним, то есть указывающим на ядро операционной системы Linux, которое загружалось сетевыми загрузчиками Etherboot. Сохраняем файл /etc/dhcpd.conf и перезапускаем демон dhcpd:

```
# /sbin/service dhcpd restart
Shutting down dhcpd:                    [  OK  ]
Starting dhcpd:                         [  OK  ]
```

Повторяем загрузку тестового X-терминала с PXE-сетевой картой. Теперь на экране отображаются такие строки:

```
Intel ® Boot Agent FE v4.1.10
Copyright © 1997-2003, Intel Corporation
CLIENT MAC ADDR: 00 11 2F CC 69 89 GUID: 848FC32 391F 8F07 66E5E96DF98A

DHCP.
CLIENT IP: 192.168.1.14 MASK: 255.255.255.0 DHCP IP: 192.168.1.1
GATEWAY IP: 192.168.1.1

PXE-E79: NBP is too big to fit in free base memory
PXE-M0F: Exiting Intel Boot Agent
```

Reboot and Select proper Boot device

or Insert Boot Media in selected Boot device and press a key

Последнее сообщение об ошибке свидетельствует, что попытка загрузки методом PXE ядра операционной системы Linux завершилась неудачей. Причина – слишком большой размер файла ядра.

Спецификация PXE предполагает получение по сети сначала небольшого загрузчика (размером не более 32 Кбайт), который и будет инициировать дальнейшую настройку бездисковой станции. Поэтому для выполнения удаленной загрузки через PXE-сетевую карту, необходимо в качестве параметра filename передать именно такой файл, а не ссылку на ядро Linux.

К счастью, создатели LTSP позаботились о нас и в дистрибутив своего проекта включили все необходимые файлы. Так, в каталоге /tftpboot/lts/2.4.24-ltsp-1/ находится файл pxelinux.0, который необходимо использовать для загрузки X-терминала через PXE-сетевую карту. А следовательно, правильная конфигурация сервера DHCP для загрузки методом PXE должна выглядеть так:

```
host pxeadmin {
    hardware ethernet     00:11:2F:CC:69:89;
    fixed-address         192.168.1.14;
    filename              "/lts/2.4.24-ltsp-1/pxelinux.0";
}
```

Еще раз перезапускаем демон dhcpd для того, чтобы он перечитал файлы конфигурации и снова пытаемся запустить X-терминал. Теперь сообщения на экране выглядят намного лучше:

Intel ® Boot Agent FE v4.1.10

Copyright © 1997-2003, Intel Corporation

CLIENT MAC ADDR: 00 11 2F CC 69 89 GUID: 848FC32 391F 8F07 66E5E96DF98A

DHCP.

CLIENT IP: 192.168.1.14 MASK: 255.255.255.0 DHCP IP: 192.168.1.1

GATEWAY IP: 192.168.1.1

PXELINUX 1.66 2002-01-01 Copyright © 1994-2002 H. Peter Anvin

UNDI data segment at: 00094190

UNDI data segment size: 94B0

UNDI code segment at: 0009D640

UNDI code segment size: 2210

PXE entry point found (we hope) at 9D64:0106

My IP address seems to be C0A8010E 192.168.1.14

ip=192.168.1.14 : 192.168.1.1 : 192.168.1.1 : 255.255.255.0

TFTP prefix: /lts/2.4.24-ltsp-1/

Try to load: pxelinux.cfg/C0A8010E

Try to load: pxelinux.cfg/C0A8010

Try to load: pxelinux.cfg/C0A801

Try to load: pxelinux.cfg/C0A80

Try to load: pxelinux.cfg/C0A8

Try to load: pxelinux.cfg/C0A

Try to load: pxelinux.cfg/C0

Try to load: pxelinux.cfg/C

Try to load: pxelinux.cfg/default

Loading bzImage-2.4.24-ltsp-1 .....

X-терминал успешно загрузился, но в приведенном выше фрагменте загрузочных сообщений я рекомендую обратить ваше внимание на значение C0A8010E. Это записанный в шестнадцатеричной форме IP-адрес нашего X-терминала, то есть 192.168.1.14. Перед выполнением загрузки ядра операционной системы Linux через локальную сеть, PXE-загрузчик пытается открыть файл конфигурации, который соответствует полученному от сервера IP-адресу. После неудачной попытки (если такого файла на диске не оказалось) он продолжает открывать файлы, названия которых хотя бы частично соответствуют IP-адресу удаленного компьютера. В последнюю очередь считывается файл /tftpboot/lts/2.4.24-ltsp-1/pxelinux.cfg/default, в котором

должна быть указана конфигурация для всех X-терминалов локальной сети вне зависимости от их IP-адреса.

Файл конфигурации для загрузчика PXE очень напоминает файл загрузчика LILO и в основном содержит параметры, которые передаются ядру (опция append). Вот пример этого файла /tftpboot/lts/2.4.24-ltsp-1/pxelinux.cfg/default:

```
prompt=0
label linux
  kernel bzImage-2.4.24-ltsp-1
  append init=/linuxrc rw root=/dev/ram0 initrd=initrd-2.4.24-ltsp-1.gz
```

Не совсем понятен принцип порядка загрузки файлов конфигурации. Так, если отсутствует файл, названный именем IP-адреса в шестнадцатеричной форме, то PXE-загрузчик пытается найти файлы в такой последовательности:

Имя файла вшестнадцатеричной форме	СоответствующийIP-адрес
C0A8010E	192.168.1.14
C0A8010	192.168.1.0
C0A801	192.168.1.0
C0A80	192.168.0.0
C0A8	192.168.0.0
C0A	192.10.0.0
C0	192.0.0.0
C	12.0.0.0

Если с предложенными C0A8010E (192.168.1.14), C0A801 (192.168.1.0), C0A8 (192.168.0.0) и C0 (192.0.0.0) еще можно согласиться, то использование C0A8010, C0A80, C0A и C вызывает недоумение.

Как только встретится соответствующий (даже частично) IP-адресу X-терминала конфигурационный файл, поиск прекращается и данный файл используется в качестве базисного. Поэтому, если у вас в сети есть X-терминал с PXE-сетевой картой, ядру которого нужно передать специфические

параметры, то наиболее рациональный способ сделать это – скопировать файл pxelinux.cfg/default в тот же каталог, но с именем, которое соответствует IP-адресу в шестнадцатеричной форме. Например, для X-терминала с IP-адресом 192.168.2.25 следует выполнить такие команды:

```
# cd /tftpboot/lts/2.4.24-ltsp-1/pxelinux.cfg/
# cp default C0A80219
```

Далее вновь созданный файл необходимо отредактировать согласно требованиям X-терминала.

Настройка сетевой карты ISA

Настройка работы сетевых адаптеров с интерфейсом ISA заслуживает особого внимания. Дело в том, что ISA-устройства, как правило, не могут себя сами идентифицировать, то есть для их настройки понадобится приложить дополнительные усилия. Конечно, сетевая карта с интерфейсом ISA это далеко не идеальный выбор, но в некоторых случаях, к сожалению, единственный. Так, в старых 486-х ПК вообще может не быть PCI-слотов, но, согласитесь, не выбрасывать же из-за такой мелочи компьютер. Не в пользу ISA-сетевых карт говорит и скорость их работы, так вы вряд ли найдете сетевой адаптер со скоростью более 10 Мбит/с, конечно 10 Мбит/с вполне хватит для офисной работы и даже прослушивания mp3-музыки через локальную сеть, а вот фильм, увы, посмотреть не удастся.

Если вы все же “счастливый” обладатель сетевой карты ISA, то для ее настройки вам необходимо добавить специальный параметр в опции файла /etc/dhcpd.conf. Рассмотрим более детально этот пример:

```
host ws001 {
    hardware ethernet     00:E0:06:E8:00:84;
    fixed-address         192.168.1.7;
    filename              "/lts/vmlinuz-2.4.19-ltsp-1";
    option option-128     e4:45:74:68:00:00;   #This is NOT a MAC address
    option option-129     "NIC=ne IO=0x300";
}
```

Как видно из фрагмента файла конфигурации сервера DHCP, на загрузку через сетевую карту с ISA интерфейсом указывает специальный параметр “option option-128 e4:45:74:68:00:00”, важно оставить его именно в таком виде, так как e4:45:74:68:00:00 – это не MAC-адрес сетевого адаптера. Далее следует выбрать тип сетевого адаптера и перечислить специфические для его модуля ядра параметры. Если вы до этого никогда не настраивали ISA устройств в Linux, то эта процедура может показаться вам достаточно сложной. На самом деле особо сложного в ней ничего нет, просто нужно понимать принцип использования модулей ядра операционной системы Linux и иметь под рукой документацию по нужному вам модулю сетевой карты.

Так, очень часто достаточно знать название модуля для сетевой карты, например, 3c509 для сетевого адаптера 3COM Etherlink III. В настройке сетевой карты вам поможет документация к модулям ядра для сетевых карт, которую вы сможете найти в каталоге /usr/src/linux/Documentation/networking. Так, для настройки упомянутой выше карты производства 3Com достаточно было сделать вот такие изменения:

```
host ws001 {
    hardware ethernet     00:E0:06:E8:00:84;
    fixed-address         192.168.1.7;
    filename              "/lts/vmlinuz-2.4.19-ltsp-1";
    option option-128     e4:45:74:68:00:00;   #This is NOT a MAC address
    option option-129     "NIC=3c509";
}
```

После внесения соответствующих изменений в файл /etc/dhcpd.conf и перезапуска DHCP-сервера можно проверять сетевую загрузку при помощи сетевого адаптера с интерфейсом ISA.

Создание пользовательского образа для загрузки

В некоторых случаях требуется создать загрузочный образ со специфическими параметрами. Например, вас раздражает 3-х секундная пауза после инициализации образа загрузки, которая возникает после вопроса: “Boot from (N)etwork or (Q)uit?”. Поможет вам в решении этой проблемы раздел сайта

http://www.rom-o-matic.net/5.2.5/, который отвечает за выбор дополнительных параметров образов. Для этого достаточно нажать в пункте 3 “(optional) To customize ROM configuration press:” кнопку "Configure".

На странице тонкой настройки загрузочного образа вам доступен выбор из нескольких десятков опций, предназначение которых, несмотря на детальные пояснения, не всегда понятны. В общем, перед вами полная свобода выбора и поле деятельности для экспериментов с образами сетевой загрузки BootROM. Большинству пользователей подойдет уже готовые образы и никаких дополнительных параметров указывать не придется.

Загрузка в текстовом режиме

После выбора образа для сетевой удаленной загрузки необходимо указать текстовый режим работы X-терминала по умолчанию. Именно текстовый, а не графический, так как для настройки он подходит значительно лучше. Например, вы используете X-терминал с такими данными в файле /etc/dhcpd.conf:

```
host yanax {
    hardware ethernet     00:02:44:2B:02:4E;
    fixed-address         192.168.1.6;
    filename              "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

Теперь в файле /opt/ltsp/i386/etc/lts.conf необходимо в параметрах выбранного X-терминала (yanax) указать правильное значение RUNLEVEL (текстовому режиму соответствует число 3):

```
[yanax]
    RUNLEVEL          = 3
```

Никакие демоны перезапускать не нужно, и поэтому сразу перейдем к проверке загрузки нашего X-терминала в текстовом режиме. Если у вас сервер настроен правильно и образ удаленной загрузки соответствует сетевому адаптеру, то на экране появится нечто подобное:

Linux version 2.4.24-ltsp-1 (root@BigDog) (gcc version 2.96 20000731 (Red Hat Linux 7.1 2.96-98)) #1 Wed Feb 4 00:03:37 EST 2004

BIOS-provided physical RAM map:

BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)

BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)

BIOS-e820: 00000000000f2158 - 0000000000100000 (reserved)

BIOS-e820: 0000000000100000 - 0000000002000000 (usable)

BIOS-e820: 00000000ffff2158 - 0000000100000000 (reserved)

32MB LOWMEM available.

On node 0 totalpages: 8192

zone(0): 4096 pages.

zone(1): 4096 pages.

zone(2): 0 pages.

DMI not present.

Kernel command line: rw root=/dev/rd/0 init=/linuxrc rw

Initializing CPU#0

Detected 166.195 MHz processor.

Console: colour VGA+ 80x25

Calibrating delay loop... 331.77 BogoMIPS

Memory: 29088k/32768k available (1203k kernel code, 3292k reserved, 481k data, 88k init, 0k highmem)

Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)

Inode cache hash table entries: 2048 (order: 2, 16384 bytes)

Mount cache hash table entries: 512 (order: 0, 4096 bytes)

Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)

Page-cache hash table entries: 8192 (order: 3, 32768 bytes)

Intel Pentium with F0 0F bug - workaround enabled.

CPU: After generic, caps: 000001bf 00000000 00000000 00000000

CPU: Common caps: 000001bf 00000000 00000000 00000000

CPU: Intel Pentium 75 - 200 stepping 0c

Checking 'hlt' instruction... OK.

Checking for popad bug... OK.

POSIX conformance testing by UNIFIX

PCI: PCI BIOS revision 2.10 entry at 0xf7531, last bus=0

PCI: Using configuration type 1

PCI: Probing PCI hardware

PCI: Probing PCI hardware (bus 00)

Limiting direct PCI/PCI transfers.

Activating ISA DMA hang workarounds.

Linux NET4.0 for Linux 2.4

Based upon Swansea University Computer Society NET3.039

Initializing RT netlink socket

apm: BIOS version 1.1 Flags 0x03 (Driver version 1.16)

Starting kswapd

devfs: v1.12c (20020818) Richard Gooch (rgooch@atnf.csiro.au)

devfs: boot_options: 0x1

intelfb: Framebuffer driver for Intel(R) 830M/845G/852GM/855GM/865G chipsets

intelfb: Version 0.7.7, written by David Dawes

tridentfb: Trident framebuffer 0.7.5 initializing

pty: 256 Unix98 ptys configured

RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize

Uniform Multi-Platform E-IDE driver Revision: 7.00beta4-2.4

ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx

NET4: Linux TCP/IP 1.0 for NET4.0

IP Protocols: ICMP, UDP, TCP

IP: routing cache hash table of 512 buckets, 4Kbytes

TCP: Hash tables configured (established 2048 bind 4096)

NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.

RAMDISK: Compressed image found at block 0

Freeing initrd memory: 928k freed

EXT2-fs warning: checktime reached, running e2fsck is recommended

VFS: Mounted root (ext2 filesystem).

```
Mounted devfs on /dev
ne2k-pci.c:v1.02 10/19/2000 D. Becker/P. Gortmaker
  http://www.scyld.com/network/ne2k-pci.html
eth0: RealTek RTL-8029 found at 0xfcc0, IRQ 10, 00:02:44:2B:02:4E.
VFS: Mounted root (ext2 filesystem).
Trying to move old root to /initrd ... failed
Unmounting old root
Trying to free ramdisk memory ... failed
Mounted devfs on /dev
Freeing unused kernel memory: 88k freed
=====================================================
Running /linuxrc
Mounting /proc
linuxrc: Installing ne2k-pci driver
modprobe ne2k-pci
Running dhclient
Mounting root filesystem: /opt/ltsp/i386 from 192.168.1.1.
Doing the pivot_root
Mounting the devfs filesystem
Running /sbin/init
Started device management daemon for /dev
Mounting /proc filesystem
Creating ramdisk on /tmp
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Current hostname: yanax
Setting up loopback device
Building the XF86Config-4 file
Building the start_ws script
Starting syslogd
bash-2.05# _
```

Теперь мы оказались в командной оболочке (shell) X-терминала. Из приведенных выше сообщений загрузки видно, что X-терминал имеет 32 Мбайт оперативной памяти, процессор Intel Pentium 166 МГц и сетевой адаптер, построенный на чипсете rtl8029.

Загрузка в текстовом режиме (продолжение)

Так как X-терминал все же не полноценный персональный компьютер, то в текстовом режиме его оболочке доступен лишь ограниченный набор команд. Именно теперь мы можем просмотреть как аппаратная часть X-терминала определилась ядром Linux.

Для начала проверим какие модули ядра загружены:

```
# /sbin/lsmod
Module              Size  Used by
ne2k-pci            4288   1
8390                5808   0 [ne2k-pci]
crc32               2848   0 [8390]
```

Совсем не густо, кроме сетевой карты больше ничего нет.

Проверим список запущенных процессов:

```
# ps
  PID  Uid     Stat Command
    1 0         S    init [3]
    2 0         S    [keventd]
    3 0         S    [kapmd]
    4 0         S    [ksoftirqd_CPU0]
    5 0         S    [kswapd]
    6 0         S    [bdflush]
    7 0         S    [kupdated]
   76 0         S    /bin/dhclient -p 68 eth0
   94 0         S    [rpciod]
  149 0         S    /bin/dhclient -p 68 eth0
  186 0         S    /sbin/devfsd /dev
  457 0         S    syslogd -m 60 -R 192.168.1.1
```

```
 496 0        S    /bin/startsess tty1 /bin/bash --login
 497 0        S    /bin/startsess tty2 /bin/bash --login
 498 0        S    sleep 300
 499 0        S    /bin/bash --login
 500 0        S    /bin/bash --login
 501 0        S    sleep 300
 502 0        S    sleep 300
 512 0        R    ps
```

Распределение оперативной памяти X-терминала:

```
# free
             total       used       free     shared    buffers
  Mem:       30104      10088      20016          0         60
 Swap:       65532          0      65532
Total:       95636      10088      85548
```

Данные по активным сетевым интерфейсам:

```
# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:44:2B:02:4E
          inet addr:192.168.1.6  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17079 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14629 errors:0 dropped:0 overruns:0 carrier:0
          collisions:251 txqueuelen:1000
          RX bytes:6356056 (6.0 Mb)  TX bytes:1998899 (1.9 Mb)
          Interrupt:10 Base address:0xfcc0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

Можно также просмотреть информацию подсистемы /proc.

```
# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 5
model           : 2
model name      : Pentium 75 - 200
stepping        : 12
cpu MHz                 : 166.194
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : yes
coma_bug        : no
fpu             : yes
fpu_exception           : yes
cpuid level     : 1
wp              : yes
flags           : fpu vme de pse tsc msr mce cx8
bogomips        : 331.77

# cat /proc/meminfo
        total:    used:    free:  shared: buffers:  cached:
Mem:  30826496 10387456 20439040        0    61440  6934528
Swap: 67104768        0 67104768
MemTotal:        30104 kB
MemFree:         19960 kB
MemShared:           0 kB
Buffers:            60 kB
Cached:           6772 kB
SwapCached:          0 kB
```

```
Active:          2580 kB
Inactive:        5516 kB
HighTotal:          0 kB
HighFree:           0 kB
LowTotal:       30104 kB
LowFree:        19960 kB
SwapTotal:      65532 kB
SwapFree:       65532 kB

# cat /proc/interrupts
           CPU0
  0:      16535          XT-PIC  timer
  1:        448          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
 10:      30483          XT-PIC  eth0
NMI:          0
ERR:          0

# cat /proc/ioports
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
03c0-03df : vga+
0cf8-0cff : PCI conf1
fcb0-fcbf : Intel Corp. 82371SB PIIX3 IDE [Natoma/Triton II]
fcc0-fcdf : Realtek Semiconductor Co., Ltd. RTL-8029(AS)
```

fcc0-fcdf : ne2k-pci

fce0-fcff : Intel Corp. 82371SB PIIX3 USB [Natoma/Triton II]

cat /proc/pci

PCI devices found:

Bus 0, device 0, function 0:

Host bridge: Intel Corp. 430HX - 82439HX TXC [Triton II] (rev 3).

Master Capable. Latency=32.

Bus 0, device 3, function 0:

Class ff00: Siemens Nixdorf AG Tulip controller, power management, switch extender (rev 1).

Bus 0, device 3, function 1:

Class ff00: Siemens Nixdorf AG Tulip controller, power management, switch extender (#2) (rev 1).

Bus 0, device 7, function 0:

ISA bridge: Intel Corp. 82371SB PIIX3 ISA [Natoma/Triton II] (rev 1).

Bus 0, device 7, function 1:

IDE interface: Intel Corp. 82371SB PIIX3 IDE [Natoma/Triton II] (rev 0).

Master Capable. Latency=32.

I/O at 0xfcb0 [0xfcbf].

Bus 0, device 7, function 2:

USB Controller: Intel Corp. 82371SB PIIX3 USB [Natoma/Triton II] (rev 1).

IRQ 11.

Master Capable. Latency=64.

I/O at 0xfce0 [0xfcff].

Bus 0, device 18, function 0:

VGA compatible controller: Matrox Graphics, Inc. MGA 1064SG [Mystique] (rev 2).

IRQ 9.

Master Capable. Latency=64.

Non-prefetchable 32 bit memory at 0xfedfc000 [0xfedfffff].

Prefetchable 32 bit memory at 0xfd800000 [0xfdffffff].

Non-prefetchable 32 bit memory at 0xfe000000 [0xfe7fffff].

Bus 0, device 19, function 0:

Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8029(AS) (rev 0).

IRQ 10.

I/O at 0xfcc0 [0xfcdf].

Имея под рукой такой набор инструментов, самое время переходить к настройке графической части X-терминала, а именно X-серверу. Но это только в том случае, если ваш X-терминал успешно загрузился, иначе в первую очередь нужно будет устранить проблемы сетевых служб, таких как NFS, TFTP, DHCP и т.д.

Из текстового режима X-терминал всегда можно перевести в графический, для этого достаточно выполнить команду:

/tmp/start_ws

Конфигурирование X-клиента

Авторы проекта LTSP не плохо постарались, чтобы настройка графического сервера X-Windows была простой задачей, однако не всегда все происходит так гладко, как нам бы этого хотелось. Так существующая возможность автоматической настройки X-терминала не всегда дает нужные результаты, но именно с нее рекомендуется начинать свое знакомство с настройкой X-сервера X-терминала.

Автоматическое построение файла XF86Config

Для того, чтобы поручить системе LTSP самой формировать файл настройки X-сервера достаточно в параметрах главного конфигурационного файла указать в разделе выбранного X-терминала опции “runlevel = 5” и “xserver = auto”, если вы этого еще не сделали в разделе [Default]:

[yanax]

RUNLEVEL = 5

XSERVER = auto

Runlevel = 5 – означает переход после запуска в графический режим, а Xserver = auto, говорит системе самой выбрать наиболее подходящий сервер X-Windows.

Теперь затаите дыхание, так как приближается торжественный момент: мы включаем X-терминал и смотрим справилась ли система LTSP с самостоятельной конфигурацией X-сервера. В моем случае все прошло относительно удачно, и через несколько секунд после загрузки ядра операционной системы Linux я увидел на экране графическое приглашение для ввода имени пользователя и пароля. Правда были и неточности, в частности, у “мышки” не работал скроллер.

Ниже приведу пример конфигурационного файла, который был автоматически сгенерирован после включения X-терминала.

```
# cat /etc/XF86Config
Section "ServerLayout"
        Identifier "XFree86 Configured"
        Screen      0   "Screen0" 0 0
        InputDevice     "Keyboard0" "CoreKeyboard"
        InputDevice     "Mouse0" "CorePointer"
EndSection
Section "Files"
        FontPath   "tcp/192.168.1.1:7100"
EndSection

Section "Module"
        Load   "extmod"
EndSection

Section "InputDevice"
       Identifier  "Keyboard0"
       Driver      "keyboard"
       Option      "XkbModel"        "pc105"
```

```
    Option     "XkbRules"          "xfree86"
    Option     "XkbLayout" "ru_UA"
    Option     "XkbOptions" "grp:ctrl_shift_toggle,grp_led:scroll"
    Option     "XkbVariant" "winkeys"
EndSection

Section "InputDevice"
    Identifier  "Mouse0"
    Driver      "mouse"
    Option      "Device"          "/dev/psaux"
    Option      "Protocol"        "PS/2"
    Option      "BaudRate"        "1200"
    Option      "Resolution"      "400"
    Option      "Emulate3Buttons" "off"
    Option      "ZAxisMapping"     "4 5"
    Option      "Buttons"         "3"
EndSection

Section "Monitor"
    Identifier      "My Monitor"
    VendorName      "Unknown"
    ModelName       "Unknown"
    HorizSync       31-62
    VertRefresh     55-90
EndSection

Section "Device"
    Identifier "My Video Card"
Driver     "mga"
```

```
EndSection

Section "Screen"
    Identifier "Screen0"
    Device "My Video Card"
    Monitor "My Monitor"
    DefaultDepth  16
    Subsection "Display"
        Depth 16
        Modes "800x600"
    EndSubSection
EndSection

Section "DRI"

EndSection
```

Как видно по формату этого файла используется X-сервер версии 4.x. Давайте кратко остановимся на характерных особенностях этого файла.

Параметр FontPath "tcp/192.168.1.1:7100" указывает на использование внешнего сервера шрифтов. Из дополнительных модулей загружается всего один: Load "extmod". Используются параметры клавиатуры, которые мы указали в разделе [Default] файла /opt/ltsp/i386/etc/lts.conf:

Option "XkbLayout" "ru_UA"– соответствует тройной раскладке клавиатуры (английской, русской и украинской);

Option "XkbOptions" "grp:ctrl_shift_toggle,grp_led:scroll"– назначает сочетание клавиш Ctrl+Shift ответственным за переключение раскладки клавиатуры и используется при этом индикатор клавиатуры “Scroll”. Теперь при выборе русской или украинской раскладки загорается индикатор “Scroll”, а при выборе английской он гаснет;

Option "XkbVariant" "winkeys"– использует такое же расположение клавиш, как это принято в операционных системах семейства Windows.

Для манипулятора “мышь” выбирается символическое устройство /dev/psaux: Option "Device" "/dev/psaux", и стандартный протокол: Option "Protocol" "PS/2".

Для монитора используются довольно типовые параметры, а вот видеоадаптер (Matrox MGA Mystique) система распознала правильно, выбрав для него драйвер mga: Driver "mga".

Графический режим по умолчанию был выбран согласно параметрам из раздела [Default]: глубина цвета HiColor (Depth 16), и разрешающая способность экрана 800x600 (Modes "800x600").

Данный пример вовсе не является показательным при настройке X-терминалов. Очень часто при первом же запуске вы либо получите мигающий черный экран и сообщение о невозможности запустить X-сервер либо серый фон графического режима с курсором в виде диагонального крестика. В любом случае настроить X-терминал у вас получится, но для этого придется затратить больше усилий. Я советую использовать автоматически сгенерированный файл /etc/XF86Config за базис и вносить в него требуемые изменения по мере необходимости.

Оптимизация конфигурации клиента

В предыдущем примере нас устроило почти все в автоматической настройке X-сервера, но хотелось бы также получить и работающее колесо у “мыши”. Для того, чтобы сообщить X-серверу, о том, что наша “мышка” имеет скроллер, добавим такую строку в раздел файла /opt/ltsp/i386/etc/lts.conf, отвечающего за конфигурацию клиента:

```
[yanax]
    RUNLEVEL          = 5
    XSERVER           = auto
    X_MOUSE_PROTOCOL   = "IMPS/2"
```

Для “мышки” с интерфейсом PS/2, протоколом поддерживающим скроллер, является IMPS/2.

Подобных параметров для тонкой настройки X-сервера в проекте LTSP довольно много. Вот их сокращенный список:

XSERVER – это уже известный вам параметр означает, какой X-сервер использовать. В большинстве случаев достаточно установить его равным “auto”, но иногда требуется принудительный выбор. Для XFree версий 4.x используется один X-сервер: XFree86, для которого автоматически выбирается драйвер видеоадаптера. Если требуется принудительно указать тип видеодрайвера, то его значение следует сопоставить параметру XSERVER. Например, для использования драйвера VESA, следует в файле конфигурации записать: XSERVER = vesa. Для версий 3.3.6 следует выбирать специальный сервер, который соответствует видеокарте, например, XF86_SVGA, XF86_S3 или XF86_S3V.

DISABLE_ACCESS_CONTROL – этот параметр указывает, разрешается ли доступ к удаленному дисплею X-терминала остальным пользователям локальной сети. По умолчанию только активному пользователю X-терминала разрешен доступ к ее оконной среде. Установив значение данного параметра “Y” вы снимите это ограничения и тем самым откроете потенциальную брешь в безопасности использования X-терминалов. Как всегда, имеем философскую дилемму: что лучше - повышенная безопасность или упрощение администрирования. Следует также заметить, что разрешить доступ к экрану X-терминала можно также при помощи команды xhost. Если ограничение на доступ к экрану получено, то пользователи имеют возможность не только запускать приложения на удаленных дисплеях, но и просматривать их содержимое. Например, простая команда:

$ import -window root -display irax:0 screen.png

Записывает текущее содержимое экрана удаленного X-терминала с доменным именем irax в графический файл screen.png.

X_MODE_0 – устанавливает разрешение монитора по умолчанию. Кроме этого параметра можно использовать также X_MODE_1 и X_MODE_2. Типичным примером установки разрешения монитора можно считать такое: X_MODE_0 = 1024x768. При желании и знании полной спецификации режима работы монитора согласно требований X-Windows можно задать и более детальное описание, например:

X_MODE_0 = 800x600 40 800 840 968 1056 600 601 605 628 +hsync +vsync.

Данный параметр является не обязательным, и при его отсутствии будет использовано значение по умолчанию.

X_DEVICE_OPTION_01 – устанавливает значение параметров, специфических для видеодрайвера. Используются значения от X_DEVICE_OPTION_01 до X_DEVICE_OPTION_10. Например, параметр X_DEVICE_OPTION_01 = "NoAccel" отключает аппаратную акселерацию у определенных видеоадаптеров. Список доступных параметров можно узнать из документации к X-серверу (см. man XF86Config) и нужному видеодрайверу (см. man mga, man radeon, man nv, man s3virge и т.п.).

X4_MODULE_01 – указывает X-серверу версий 4.x загрузить от 1 до десяти специальных модулей (параметры от X4_MODULE_01 до X4_MODULE_10). Типичным примером может служить команда X4_MODULE_01 = "dri" в паре с X4_MODULE_02 = "glx" (да, на X-терминале возможен даже OpenGL, правда с некоторыми ограничениями). Загружать вручную модуль extmod не требуется, так как его загрузка выполняется по умолчанию.

X4_BUSID – некоторым типам чипсетов нужен этот параметр, чтобы правильно распознать видеоадаптер. На практике же его чаще используют для работы с клиентом VMware, для этого достаточно указать значение параметра равным “PCI:0:15:0”.

X4_DRI – этот параметр указывает модуль, который будет использован для работы с Direct Rendering Interface X-сервера версии 4.x.

X_VIDEORAM – принудительно определяет объем видеопамяти видеоадаптера. Значение необходимо задавать в килобайтах. Например, для видеоадаптера с 4Мб видео-ОЗУ следует задать параметр X_VIDEORAM = 4096. Особенно полезен этот параметр на материнских платах с чипсетом i810 и ограниченным объемом оперативной памяти. Если этот параметр опущен, то система попытается сама определить объем доступной видеопамяти. Бывают довольно забавные случаи, когда требуется использовать это значение. Например, в своей практике я сталкивался с X-терминалом на чипсете i810,

который напрочь отказывался работать с оперативной памятью в 32 Мбайт. На экране появлялось такое сообщение об ошибке:

(EE) I810(0): Less than 6Mb of AGP memoryis available. Cannot proceed.

(EE) Screen(s) found, but none have a usable configuration.

Fatal server error:

no screen found

Смысл его заключается в том, что системе не хватает AGP-памяти для инициализации видео-карты. Правда, если установить неправильное значение X_VIDEORAM, то на экране можно увидеть такое сообщение об ошибке:

(EE) I810(0): Ring buffer allocation failed

X_MOUSE_PROTOCOL – знакомый уже параметр указывает на протокол работы с манипулятором “мышь”. Значение по умолчанию “PS/2”, которое соответствует “мышке” подключенной к порту PS/2. Для “мышек” на COM порту принято использовать протокол “Microsoft”. Если “мышь” имеет скроллер, то для его активации следует использовать протоколы IMPS/2 и IntelliMouse соответственно для PS/2 и COM-порта. Этот параметр непосредственно зависит от X_MOUSE_DEVICE.

X_MOUSE_DEVICE – определяет к какому символьному устройству подключена “мышка”, для PS/2-"мышей" это /dev/psaux (значение по умолчанию), а для COM-портов: /dev/ttyS0 (COM1:) или /dev/ttyS1 (COM2:). Для менее распространенных “грызунов” с интерфейсом USB таким устройством является /dev/input/mice, правда для их активизации требуется загрузить дополнительные модули (как правило, usb-uhci, mousedev и usbmouse).

X_MOUSE_RESOLUTION – этот параметр определяет скорость перемешения курсора “мыши” по экрану X-терминала. Чем больше значение, тем больше скорость. По умолчанию для мышей PS/2 используется значение 400, а для COM “мышек” - 50.

X_MOUSE_BUTTONS – указывает сколько кнопок у вашей “мышки”. По умолчанию считается, что “мышь” 3-х кнопочная, причем если “мышка” со скроллером, то нажатие на скроллер соответствует 3-й кнопке. Данный

параметр связан с параметром X_MOUSE_EMULATE3BTN, который позволяет эмулировать нажатие 3-й кнопки на 2-х кнопочной “мышке”.

X_MOUSE_EMULATE3BTN – если этот параметр равен “Y”, то включается режим эмуляции нажатия 3-й кнопки. Теперь нажатие двух кнопок одновременно (левой и правой) воспринимается X-сервером, как третья кнопка.

X_MOUSE_BAUD – в документации написано, что данный параметр имеет смысл только для “мышей”, подключенных к последовательному порту (хотя, насколько мне известно, все "мыши" работают через последовательный порт PS/2, COM или USB). Он задает скорость обмена информацией в бодах. Значение по умолчанию 1200.

X_COLOR_DEPTH – устанавливает глубину цвета для экрана X-терминала. Доступные значения 8, 15, 16, 24 и 32. К сожалению, не все X-серверы поддерживают данные цветовые режимы. Значение по умолчанию 16 соответствует режиму HiColor и 65536 оттенкам цвета. Числовые значения глубины цвета показывают количество бит, которым описывается каждый пиксель. Так, в режиме X_COLOR_DEPTH = 15, на каждый цветовой компонент RGB отводиться по 5 бит, что в итоге составляет 25 x 25 x 25 = 32 x 32 x 32 = 32768 оттенков цвета. Информацию про поддержку цветовых режимов X-сервером смотрите в документации к нему и драйверу, соответствующему вашей видеокарте.

X_HORZSYNC– устанавливает значение горизонтальной разверстки монитора в кГц. Значение по умолчанию – "31-62". Правильные параметры ищите в документации к вашему монитору.

X_VERTREFRESH– значение вертикальной разверстки монитора в Гц. Значение по умолчанию – "55-90". Этот параметр в паре с предыдущим напрямую влияет на то, с какой частотой будет обновляться картинка на вашем мониторе. Установка неправильных параметров может привести даже к выходу монитора из строя. В своей практике мне еще не удалось таким способом “спалить” ни один монитор, но предупредить о возможной опасности никогда не будет лишним.

Оптимизация конфигурации клиента (продолжение)

XF86CONFIG_FILE– в редких случаях для X-терминала требуется использовать файл конфигурации X-сервера, который кардинально отличается от того, что предлагает проект LTSP. Если вы все же столкнулись с такой ситуацией, то использование такого файла действительно выход. Поместить этот файл нужно в директорию /tftpboot/lts/ltsroot/etc и дать ему соответствующее название, например, XF86Config.Unique. Теперь в параметрах X-терминала достаточно сослаться на уникальный файл конфигурации: XF86CONFIG_FILE = XF86Config.Unique. Полезен этот файл также для экспериментов с конфигурацией X-сервера, так как теперь вы никак не ограничены в возможностях его настройки. Единственное условие – наличие соответствующих знаний по используемым параметрам файла XF86Config.

USE_TOUCH– установив этот параметр в значение равное “Y”, вы тем самым сообщаете X-серверу, что к X-терминалу подключен сенсорный экран, реагирующий на прикосновение. Настройка сенсорного экрана довольно специфическая задача, поэтому она выходит за рамки данной книги (ваш покорный слуга не имел возможности на практике поиграть с настройками этих “чудо” устройств). Тем не менее для настройки работы сенсорного экрана используются следующие опции файла /opt/ltsp/i386/etc/lts.conf:

X_TOUCH_DEVICE

X_TOUCH_MINX

X_TOUCH_MAXX

X_TOUCH_MINY

X_TOUCH_MAXY

X_TOUCH_UNDELAY

X_TOUCH_RPTDELAY

Кстати, сенсорные экраны могут найти прекрасное применение в сочетании с X-терминалами, так как последние могут быть установлены в местах, недоступных обычным компьютерам (запыленные помещения, воздействие вибраций, завышенный или заниженный температурный режим и т.д).

Следующие параметры относятся к настройке работы клавиатуры X-терминала:

XkbTypes – здесь можно указать тип используемой клавиатуры. Значение по умолчанию “default”. Используется только в X-сервере версии 3.3.6.

XkbCompat – определяет совместимость клавиатуры. Значение по умолчанию “default”. Используется только в X-сервере версии 3.3.6.

XkbSymbols – набор символов для данного типа клавиатуры. Значение по умолчанию “us(pc101)”. Используется только в X-сервере версии 3.3.6.

XkbModel – модель клавиатуры, которая используется. Значение по умолчанию “pc101”. Используется только в X-сервере версии 3.3.6.

XkbLayout – раскладка клавиатуры. Очень важный параметр для 2-х и 3-х язычных пользователей. Значение по умолчанию “us” удовлетворит только жителей англоязычных стран и некоторых радикальных сисадминов. Для жителей Украины подойдет такая раскладка: XkbLayout = "ru_UA". Граждане России будут использовать раскладку: XkbLayout = "ru".

К сожалению, в базовой конфигурации X-терминала LTSP отсутствует настройка таких параметров клавиатуры как XkbOptions и XkbVariant.

XkbOptions – задает дополнительные параметры работы клавиатуры. Так, при использовании нескольких раскладок клавиатуры целесообразно использовать значение данного параметра:

Option “XkbOptions” "grp:ctrl_shift_toggle,grp_led:scroll"

При котором индикатор клавиатуры “Scroll” будет показывать какой язык сейчас активный. Например, английский - индикатор выключен, русский – индикатор включен.

XkbVariant– позволяет задать режим использования Windows-совместимой клавиатуры, что, согласитесь, очень даже уместно, так как в наших магазинах вы не часто встретите клавиатуру соответствующую стандартам XFree. Наиболее популярное значение:

Option "XkbVariant" "winkeys"

Не смотря на то, что в автоматически генерируемом файла /etc/XF86Config таких параметров нет, ничто не мешает сделать их используемыми по

умолчанию. Для этого достаточно знать, каким образом формируются файлы XF86Config на X-терминалах. За автоматическое формирование файлов /etc/XF86Config отвечают два скрипта:

/opt/ltsp/i386/etc/rc.setupx – для X-сервера версии 4.x

/opt/ltsp/i386/etc/rc.setupx3 – для X-серверов версии 3.3.6

Открываем эти файлы и ищем раздел, посвященный клавиатуре. В файле /opt/ltsp/i386/etc/rc.setupx он должен выглядеть примерно так:

```
#
# Setup Keyboard and Mouse info
#

XkbLayout=`get_cfg XkbLayout`

cat <<-EOF >>${XF86CONFIG}

Section "InputDevice"
	Identifier  "Keyboard0"
	Driver      "keyboard"
	Option      "XkbLayout" "${XkbLayout:-"us"}"
EndSection
```

Как видите, довольно скудный набор параметров. Просто добавим сюда нужные строчки (для XkbVariant и XkbOptions). Теперь новый раздел настройки клавиатуры выглядит так:

```
#
# Setup Keyboard and Mouse info
#

XkbLayout=`get_cfg XkbLayout`

cat <<-EOF >>${XF86CONFIG}
```

```
Section "InputDevice"
    Identifier  "Keyboard0"
    Driver      "keyboard"
    Option      "XkbLayout" "${XkbLayout:-"us"}"
    Option      "XkbOptions" "grp:ctrl_shift_toggle,grp_led:scroll"
    Option      "XkbVariant" "winkeys"
EndSection
```

Аналогичные изменения вносим и в файл /opt/ltsp/i386/etc/rc.setupx3, если предполагаем использовать клиенты с X-сервером версии 3.3.6. Так, раздел относящийся к настройке клавиатуры, в оригинальном файле выглядел следующим образом:

```
#
# Setup Keyboard and Mouse info
#

XkbTypes=`   get_cfg XkbTypes    "default"`
XkbCompat=`  get_cfg XkbCompat   "default"`
XkbSymbols=` get_cfg XkbSymbols  "us(pc101)"`
XkbModel=`   get_cfg XkbModel    "pc101"`
XkbLayout=`  get_cfg XkbLayout   "us"`

cat <<-EOF >>${XF86CONFIG}
   Section "ServerFlags"
   EndSection

   Section "Keyboard"
      Protocol        "Standard"
      AutoRepeat      500 5
      LeftAlt         Meta
      RightAlt        Meta
      ScrollLock      Compose
```

```
        RightCtl        Control
        XkbKeycodes     "xfree86"
        XkbTypes        "${XkbTypes}"
        XkbCompat       "${XkbCompat}"
        XkbSymbols      "${XkbSymbols}"
        XkbGeometry     "pc"
        XkbRules        "xfree86"
        XkbModel        "${XkbModel}"
        XkbLayout       "${XkbLayout}"
    EndSection
```

После внесения соответствующих изменений посмотрите на его новую редакцию:

```
#
# Setup Keyboard and Mouse info
#

XkbTypes=`   get_cfg XkbTypes    "default"`
XkbCompat=`  get_cfg XkbCompat   "default"`
XkbSymbols=` get_cfg XkbSymbols  "us(pc101)"`
XkbModel=`   get_cfg XkbModel    "pc101"`
XkbLayout=`  get_cfg XkbLayout   "us"`

cat <<-EOF >>${XF86CONFIG}
    Section "ServerFlags"
    EndSection

    Section "Keyboard"
        Protocol        "Standard"
        AutoRepeat      500 5
        LeftAlt         Meta
        RightAlt        Meta
```

```
ScrollLock      Compose
RightCtl        Control
XkbKeycodes     "xfree86"
XkbTypes        "${XkbTypes}"
XkbCompat       "${XkbCompat}"
XkbSymbols      "${XkbSymbols}"
XkbGeometry     "pc"
XkbRules        "xfree86"
XkbModel        "${XkbModel}"
XkbLayout       "${XkbLayout}"
XkbOptions      "grp:ctrl_shift_toggle,grp_led:scroll"
XkbVariant      "winkeys"
EndSection
```

Такие несложные коррективы позволяют значительно улучшить автоматическую настройку X-терминалов. После внесения изменений в файлы, необходимо их сохранить и перегрузить X-терминал, чтобы увидеть результат воочию.

Вообще изучение структуры файла /opt/ltsp/i386/etc/rc.setupx - это прекрасный способ понять, как проект LTSP автоматически генерирует файл /etc/XF86Config, а также лучше разобраться с особенностями настройки X-сервера.

Типовые конфигурации клиентов

Настройка графической подсистемы X-терминала – всегда привязана к аппаратному обеспечению. Следовательно, здесь не столько теории сколько практики. Из всего множества выпущенного в мире компьютерного оборудования у конечного пользователя присутствует, как правило, лишь ничтожная его часть. Так, например, если ваш старый персональный компьютер был куплен между 1996 и 1997 годом, то с большой степенью вероятности можно утверждать, что он оснащен видеоадаптером с интерфейсом PCI и чипсетом от s3. И таких примеров достаточно много. Поэтому целью данного

раздела является описание наиболее типичных конфигураций реальных X-терминалов. Итак, приступим.

Клавиатура

Никаких специальных приемов настройки не требует. Достаточно указать тип раскладки, например, XkbLayout = "ru".

“Мышка” подключенная к порту PS/2 трех кнопочная без скроллера:

X_MOUSE_PROTOCOL = "PS/2"

X_MOUSE_DEVICE = "/dev/psaux"

X_MOUSE_BUTTONS = 3

“Мышка” подключенная к порту PS/2 двух кнопочная без скроллера:

X_MOUSE_PROTOCOL = "PS/2"

X_MOUSE_DEVICE = "/dev/psaux"

X_MOUSE_BUTTONS = 2

X_MOUSE_EMULATE3BTN = Y

“Мышка” подключенная к порту PS/2 двух кнопочная со скроллером:

X_MOUSE_PROTOCOL = "IMPS/2"

X_MOUSE_DEVICE = "/dev/psaux"

X_MOUSE_BUTTONS = 2

“Мышка” подключенная к COM-порту (COM1) без скроллера (3 кнопки):

X_MOUSE_PROTOCOL = "Microsoft"

X_MOUSE_DEVICE = "/dev/ttyS0”

X_MOUSE_BUTTONS = 3

“Мышка” подключенная к COM-порту (COM2) без скроллера (2 кнопки с эмуляцией нажатия третьей):

X_MOUSE_PROTOCOL = "Microsoft"

X_MOUSE_DEVICE = "/dev/ttyS1”

X_MOUSE_BUTTONS = 2

X_MOUSE_EMULATE3BTN = Y

“Мышка” подключенная к COM-порту (COM1) со скроллером (2 кнопки):

X_MOUSE_PROTOCOL = "IntelliMouse"

X_MOUSE_DEVICE = "/dev/ttyS0”

X_MOUSE_BUTTONS = 2

“Мышка” подключенная к USB-порту со скроллером (2 кнопки):

MODULE_01 = usb-uhci

MODULE_02 = mousedev

MODULE_03 = usbmouse

X_MOUSE_PROTOCOL = "IMPS/2"

X_MOUSE_DEVICE = /dev/input/mice

X_MOUSE_BUTTONS = 2

Разрешение для ЭЛТ-монитора с диагональю 15” (800x600):

X_MODE_0 = 800x600

Разрешение для ЭЛТ-монитора с диагональю 17” (1024x768):

X_MODE_0 = 1024x768

Разрешение для ЭЛТ-монитора с диагональю 19” (1280x1024):

X_MODE_0 = 1280x1024

Разрешение для LCD-монитора с диагональю 15” (1024x768):

X_MODE_0 = 1024x768

Разрешение для ЭЛТ-монитора с диагональю 17” (1280x1024):

X_MODE_0 = 1280x1024

Глубина цвета HiColor (16 бит):

X_COLOR_DEPTH = 16

Глубина цвета TrueColor (24 бита):

X_COLOR_DEPTH = 24

Глубина цвета TrueColor (32 бита):

X_COLOR_DEPTH = 32

Горизонтальная и вертикальная частоты разверстки для ЭЛТ-монитора с диагональю 17”:

X_HORZSYNC = "30-96"

X_VERTREFRESH = "50-160"

Запрет аппаратной акселерации (иногда помогает избавиться от артефактов):

X_DEVICE_OPTION_01 = "noaccel"

Работа в режиме VESA:

XSERVER = vesa

X-сервер для большинства видеоадаптером на чипсете s3:

XSERVER = XF86_S3

X-сервер SVGA версии 3.3.6:

XSERVER = XF86_SVGA

Использование принтера подключенного к параллельному порту X-терминала (порт tcp/ip 9100):

PRINTER_0_DEVICE = /dev/lp0

RINTER_0_TYPE = P

Использование принтера подключенного к параллельному порту X-терминала с нестандартным портом tcp/ip, например, 9150:

PRINTER_0_DEVICE = /dev/lp0

RINTER_0_TYPE = P

PRINTER_0_PORT = 9150

Использование принтера подключенного к USB-порту X-терминала (порт tcp/ip 9100):

MODULE_01 = usb-uhci

MODULE_02 = printer

PRINTER_0_DEVICE = /dev/usb/lp0

PRINTER_0_TYPE = S

Автоматическая настройка звуковой карты на X-терминале (демон звука EsounD):

SOUND = Y

SOUND_DAEMON = esd

Ручная настройка звуковой карты на X-терминале (демон звука EsounD, звуковая карта на чипсете OPL3sa2):

SOUND = Y

SOUND_DAEMON = esd

VOLUME = 100

SMODULE_01 = soundcore

SMODULE_02 = sound

SMODULE_03 = ad1848

SMODULE_04 = uart401

SMODULE_05 = opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=1 dma2=0

Как видите, особым разнообразием типовые настройки не отличаются. Правда, иногда приходится сталкиваться и с трудностями. Так, наиболее распространенной проблемой настройки X-терминалов является рассогласованность между параметрами видеоадаптера и монитора, или, другими словами, в тех режимах, что может работать видеоадаптер - не может работать монитор и наоборот. Как правило, все решается путем установки правильных значений X_HORZSYNC и X_VERTREFRESH, но иногда может понадобиться и полная спецификация нужного режима.

На практике, проблемы между видеоадаптером и монитором проявляются так: после запуска X-терминала его экран начинает мигать с интервалом в несколько секунд, а затем переходит в текстовый режим и на экране содержится сообщение об ошибке. В другом случае после аналогичного мигания монитор все же входит в графический режим, но частота регенерации экрана неудовлетворительная, например, 60 Гц.

Для решения подобных проблем следует в первую очередь убедиться, что выбранное разрешение по умолчанию (параметр X_MODE_0) соответствует техническим возможностям и монитора и видеокарты. Так, при установленном X_MODE_0 = 102x768 не каждый 15-ти дюймовый монитор покажет качественную картинку, да и видеоадаптер должен быть оснащен по крайней мере 2 Мбайтами видеопамяти. Затем следует проверить глубину цвета (параметр X_COLOR_DEPTH). Помните, что не каждый X-сервер поддерживает полный набор глубин цвета. Далее устанавливаем стандартные параметры для горизонтальной и вертикальной разверстки. Здесь программа минимум – заставить монитор включить хотя бы VESA-совместимый графический режим в 60 Гц. Если мы своей цели достигли, то проще всего отредактировать значение нижней границы параметра X_VERTREFRESH, и

установить его равным желаемой частоте вертикальной разверстки, например, 85 Гц. То есть из режима:

X_VERTREFRESH = "50-160"

Мы получим:

X_VERTREFRESH = "85-160"

Теперь перегружаем X-терминал и смотрим как он работает при новой частоте регенерации экрана.

Настраивать X-терминал из графического режима неудобно, так как X-сервер в цикле будет пытаться перепробовать разные конфигурации, а пользователь на протяжении нескольких минут вынужден будет любоваться мигающим экраном. Поэтому для настройки X-терминала целесообразно перевести его в текстовый режим, и уже из него запускать X-сессию командой:

/tmp/start_ws

Кстати, содержимое файла этого командного сценария предельно простое (для версий XFree86 4.x):

/usr/X11R6/bin/XFree86 -query 192.168.1.1, где

192.168.1.1 – IP-адрес сервера X-терминалов, а параметр query указывает на необходимость использовать подключение к внешнему X-серверу.

Например, для XFree86 версии 3.3.6 и сервера S3, содержимое файла /tmp/start_ws будет таким:

/usr/X11R6/bin/XF86_S3 -query 192.168.1.1

Влияние видеосистемы на работу X-терминала

Производительность X-терминала в первую очередь зависит от его видеосистемы, то есть связки видеокарта + монитор. Процессор в расчет практически не берется, так как в отрисовывании окошек на X-терминале он принимает очень скромную роль. Объем оперативной памяти также не особо критичен (все, что больше 8 Мбайт, должно подойти).

С монитором ситуация довольно простая. Он должен обеспечивать приличное качество изображения в приемлемой для пользователя разрешающей способности и частоте обновления экрана (последнее важно лишь для мониторов на электронно-лучевой трубке). Конечно, при прочих

равных, чем больше разрешение монитора, тем лучше. Рациональный минимум – 800x600.

С видеокартой все обстоит несколько сложнее. Во-первых, скорость вывода на экран графической информации должна быть высокой, а соответственно про ISA-карты можно забыть сразу, а AGP-картам отдавать преимущество перед картами с интерфейсом PCI. Как уже было упомянуто в этой книге, лучше отдать предпочтение видеоадаптерам, которые поддерживаются проектом XFree86 версий 4.x, чем совместимым с версией 3.3.6. И даже если вам в руки попала AGP-карта поддерживаемая в XFree86 версии 4.x и у неё достаточно видеопамяти, поверьте, радоваться еще рано. Приведу реальный пример из жизни. Настраивал я как-то X-терминал с такой конфигурацией: PII-233/RAM 32Mb/AGP Matrox G100 4Mb/PCI rtl8139 100 Mbit. Вроде бы очень даже достойная для терминала конфигурация, я бы даже сказал излишняя. О видеокарте Matrox на X-терминале вообще можно только мечтать, так как видеоадаптеры производства этой фирмы издавна славятся своим качеством работы в 2D и хорошей поддержкой в XFree. Быстрая графическая шина AGP и достаточный объем видеопамяти (4 Мб) теоретически должны были обеспечить наилучшее быстродействие, а на практике было впечатление, что у данного X-терминала включились тормоза, причем особенно это было заметно при выполнении прокрутки в документах OpenOffice (рабочая область при прорисовке несколько раз мигала). Интересно, что установка этого видеоадаптера в еще более быстрый X-терминал (Celeron 1 GHz), никак не обозначилась на скорости его работы. Я начал разбираться с параметрами, которые были доступны драйверу mga (команда man mga). Справочная информация оказалась очень содержательной и я стал подбирать параметры для тонкой настройки этого драйвера. Так, например, отключение аппаратного ускорения Option "NoAccel" вызвало жуткое замедление работы графической подсистемы. Помог в решении проблемы скорости только один параметр Option "ShadowFB", но неожиданно появилась новая проблема: после щелчка “мышкой” с экрана исчезало изображение курсора. Если “мышкой” подвигать, то курсор снова появлялся. Если быть кратким, то день возни с параметрами

видеокарты так ничего и не дал. Единственное, что меня тогда насторожило, так это то, что иногда X-терминал вообще не запускался. На экране появлялось искаженное изображение и компьютер “зависал”. Как позже выяснилось, данный видеоадаптер был с аппаратным дефектом, что и сказывалось на его работоспособности. У вас может сложиться мнение, что данная история не несет в себе никакой поучительной нагрузки, так как видеокарта в итоге оказалась бракованная. Но не следует забывать, что X-терминалы используют, как правило, устаревшее оборудования, а стало быть нужно всегда ожидать, что рано или поздно вам придется столкнуться с трудноуловимыми аппаратными сбоями.

Если в X-терминале используется сервер XFree версии 3.3.6, то пользователя подстерегает еще одна проблема: отсутствие поддержки сглаженных шрифтов этим сервером. В принципе это не так уж и страшно, но кроме этого я столкнулся также с проблемами запуска приложений KDE, а именно: корректного отображения на экране шрифтов и значительного замедления отображения на экране окон этих программных продуктов. Подозреваю, что данная проблема скрыта за использованием устаревшего X-сервера, но в некоторых случаях это единственный выход заставить X-терминал работать.

В документации, посвященной настройки клиентов LTSP, вы часто можете встретить утверждение, что лучше отказаться от использования серверов версии 3.3.6, а также что практически любую видеокарту можно запустить с драйвером vesa или на худой конец с драйвером vga. Работать в vesa режиме конечно можно, но комфортными такие условия явно не назовешь. Во-первых, частота регенерации экрана в 60 Гц подойдет только пользователям жидкокристаллических мониторов (правда, иногда мне удавалось заставить работать X-терминал в VESA-режиме и при больших частотах), а во-вторых, отсутствие полноценной аппаратной поддержки значительно замедлит процесс отображения графической информации на экране X-терминала. Более того, в процессе формирования изображения клиента большую роль будет играть центральный процессор сервера (он вынужден будет заполнять

дополнительными вычислениями отсутствующую аппаратную акселерацию), что в конечном итоге негативно скажется на работе остальных пользователей.

Шрифты, точнее сказать их поддержка – это пожалуй самое слабое место X-Windows. Даже имеющаяся сейчас полная поддержка шрифтов типа TrueType все равно находится на уровень ниже чем в операционных системах Windows и MacOS. Тут и повышенные требования к самим шрифтам, и стабильность сервера шрифтов (xfs) и даже отображение их на экране, в общем работать можно, но хотелось, чтобы все было лучше. Конечно, не будем спорить о преимуществах и недостатках метода сглаживания экранных шрифтов в операционной системе Linux, это дело привычки, но, на мой взгляд, в Windows они выглядят немного лучше. Проблемы со шрифтами не чужды и X-терминалам. Помню был случай, когда я настраивал X-терминал с видеокартой S3 Trio64 (86C764X) 1Мб PCI. Несмотря на то, что для работы с этой картой в проекте LTSP был рекомендован как сервер XF86_S3 так и XFree86 я решил опробовать ее на новом X-сервере версии 4.2.99. На удивление сервер стартовал без проблем, и на экране я увидел графическую заставку ввода имени пользователя и пароля в режиме монитора 800x600 85 Гц. Я сначала обрадовался, но потом понял что сделал это рано и зря. На экране постоянно появлялись артефакты (искажения изображения), что сводило все мои усилия на нет. Ответным действием было запрещение акселерации (в параметрах файла /opt/ltsp/i386/etc/lts.conf):

X_DEVICE_OPTION_01 = "noaccel"

После перезагрузки X-терминала артефакты исчезли, но значительно замедлилась работа графической системы терминала. Обратившись к файлу справки X-сервера, я узнал, что аппаратная акселерация состоит из многих составляющих, и путем экспериментов было выяснено, что этими компонентами для данной s3-карты являются:

X_DEVICE_OPTION_01 = "XaaNoSolidFillRect"

X_DEVICE_OPTION_02 = "XaaNoScreenToScreenCopy"

X_DEVICE_OPTION_03 = "XaaNoColor8x8PatternFillRect"

К моему величайшему сожалению, применение только этих трех ограничений вызвало практически равносильное падение производительности X-терминала, как в случае с X_DEVICE_OPTION_01 = "noaccel". Интересно другое, что при работе этой видеокарты на обычном компьютере с операционной системой Linux подобных артефактов не наблюдалось. По видимому, ошибка была скрыта в алгоритме работы с удаленным X-сервером.

Благо у меня в запасе была еще одна видеокарта производства все того же фирмы S3 Trio64V+ (86C765) 2Мб PCI. После установки ее в X-терминал без всяких дополнительных опций никаких артефактов замечено не было. И не смотря на это, мне все же пришлось перевести ее в режим работы с сервером XF86_S3, так как при работе с XFree86 наблюдалось нарушение кернинга шрифтов (неправильное отображение межбуквенных расстояний), когда одни буквы "налазили" на соседние, а пробел между словами иногда вообще нельзя было отличить.

Если делать вывод по особенностям работы видеосистемы X-терминалов, то следует отметить, что ее скорость - главная составляющая хорошей работы пользователя X-терминала, поэтому у X-терминала два главных компонента: видеокарта и монитор.

Проблемы, требующие собственного файла XF86Config

Редко, но иногда попадается нестандартная конфигурация аппаратного обеспечения терминала, которая требует создания собственного файла конфигурации для X-сервера. У меня в практике был только один такой случай. Я настраивал свой ПК для альтернативной работы в качестве X-терминала. То есть это был полноценный персональный компьютер, который мог работать в двух режимах: как персоналка и как X-терминал (для этого достаточно было вставить в флоппи дисковод загрузочную дискету с загрузчиком Etherboot). С обычной рабочей станцией Linux проблем не было, она загружалась в графическом режиме нормально. А при работе в режиме X-терминала я получал только черный экран с сообщением об ошибке, что устройство не найдено. Конфигурация этого компьютера была следующая: Intel Celeron Tualatin 1 GHz / Motherboard PC Partner i815 EP / RAM 512Mb / Video AGP SIS

6326 4Mb /HDD WD 40Gb 8 Mb cache 7200 rpm / FDD / CDRW ASUS 5232AS / Ethernet PCI rtl8029 10 Mbit / ATX 250 W. Почему-то автоконфигуратор LTSP определял в этом компьютере видеокарту не SIS 6326, а Intel 810, а указать в параметрах тип видеодрайвера для XFree86 версий 4.x в этой системе не предусмотрено (на самом деле еще как и предусмотрено, но для этого используется тот же параметр XSERVER). Выход из подобной ситуации – создание своего собственного файла конфигурации для X-сервера. Полностью самостоятельно создавать файл настройки XF86Config занятие неблагодарное. Лучше откорректировать уже существующий файл. Для того, чтобы получить содержимое автоматически построенного файла следует перевести терминал в текстовый режим и включить файл подкачки через NFS:

```
[ws001]
    RUNLEVEL        = 3
    USE_NFS_SWAP    = Y
    SWAPFILE_SIZE   = 32m
```

Пример приведен для X-терминала с именем ws001 и размером файла подкачки в 32 Мбайта. Использование файла подкачки вызвано необходимостью получения доступа с возможностью записи в директории на X-терминал сервере (каталог /var/opt/ltsp/swapfiles). Теперь для того, чтобы копировать сгенерированный файл XF86Config на сервер, достаточно выполнить такую команду:

```
# cp /etc/XF86Config /tmp/swapfiles
```

На сервере файл /var/opt/ltsp/swapfiles/XF86Config следует переместить в папку /tftpboot/lts/ltsroot/etc и назвать его, например, XF86Config.ws001 :

```
# mdir /tftpboot/lts/ltsroot
# mdir /tftpboot/lts/ltsroot/etc
# mv /var/opt/ltsp/swapfiles/XF86Config /tftpboot/lts/ltsroot/etc/XF86Config.ws001
```

В файле /opt/ltsp/i386/etc/lts.conf делаем такую запись:

```
[ws001]
    RUNLEVEL        = 5
```

XF86CONFIG_FILE = XF86Config.ws001

Теперь все изменения, которые пользователь будет вносить в файл /tftpboot/lts/ltsroot/etc/XF86Config.ws001, будут отражаться на работе X-терминала с именем ws001.

Еще раз подчеркну, что подобные исключения бывают крайне редко, и как правило, система LTSP обеспечивает нужную функциональность для тонкой настройки X-сервера, но в любом случае полезно знать, что имеется возможность использования специального файла XF86Config.

Выбор оптимального программного обеспечения для работы на X-терминале

Как было уже неоднократно сказано, для работы на X-терминале можно использовать практически любое программное обеспечение, но часть его будет работать лучше, а часть хуже. Это в первую очередь связано с тем, как программное обеспечение использует ресурсы локальной сети. Чем больше нагрузка на сеть, тем менее комфортно будет чувствовать себя пользователь. Помню был в моей практике случай, когда сеть X-терминалов работала через коаксиальный кабель в 10 Мбит/с, так вот запуск любым пользователем почтового клиента Ximian Evolution приводил к 10 секундной паузе в работе всех X-терминалов. В любом случае, при выборе программного обеспечения для X-терминальной сети следует руководствоваться критериями пропускной способности локальной сети, количеством рабочих станций и возможностями сервера.

Так как с серверной операционной системой мы уже определились (надеюсь и вы тоже), то следующим пунктом будет выбор оптимального оконного менеджера. Конечно, если вам очень нравятся KDE или Gnome, то вас сложно будет переубедить в том, что для X-терминала это далеко не самый лучший выбор. Согласен, функциональность этих менеджеров окон находится на достаточно высоком уровне, но и системных ресурсов они потребляют изрядное количество, да и стабильной работой они также не могут похвастаться. Для X-терминалов скорее подойдет менеджер окон попроще, например IceWM, Window Maker, twm, mwm или XFce.

Лично я перепробовал все эти менеджеры окон и должен отметить, что каждый из них имел свои как сильные, так и слабые стороны. Так Window Maker (см. рис. 7.8) мне понравился своим стильным дизайном, возможностью находу добавлять и удалять виртуальные рабочие столы, удачной GUI-программой настройки своего поведения (для каждого пользователя отдельно), а также тем, что весь он управляется одним программным файлом /usr/bin/wmaker, который в памяти занимает менее 8 мегабайт (половина из которых в совместном использовании). Конечно, для пользователей, привыкших к интерфейсу операционных систем win32, его внешний вид покажется несколько революционным. Все элементы управления расположены в одинаковых по размеру квадратных иконках, причем часто без подписей, что вносит определенную путаницу в сознание начинающего пользователя. Главное меню вызывается щелчком правой кнопкой “мыши” по свободному полю рабочего стола, причем если вы щелкните очень близко к краю экрана, то часть меню останется для вас недоступной (хотя если потянуть “мышкой” в направлении скрытой части, то все меню плавно прокрутится для обозрения). Раскрытые на весь экран окна закрывают собой импровизированную панель задач (ряд квадратных иконок в нижнем левом углу экрана), причем на каждом рабочем столе присутствуют иконки всех запущенных программ, хотя запуск некоторых из них не приводит к появлению соответствующей иконки. Если вы вызвали системное меню (щелчок правой кнопкой “мыши”), или список активных окон (щелчок средней кнопкой “мыши”), то чтобы убрать их с экрана, нужно обязательно либо выбрать один из их пунктов, или щелкнуть на свободном месте рабочего стола правой или соответственно средней кнопкой “мыши”. В общем, данный менеджер окон вызвал у меня противоречивые чувства. Так, мне лично он понравился, но я бы не рекомендовал его ставить всем пользователям, особенно тем, которые имеют опыт работы только с системами Microsoft Windows.

Рис. 7.8. Пример экрана менеджера окон Window Maker запущенного на X-терминале

Предложив менеджер окон twm для работы на X-терминале я, конечно, погорячился. На мой взгляд, единственным его достоинством является мгновенный запуск. Честно говоря, когда я первый раз запустил его, то даже подумал, что он “завис”, так как кроме синего фона экрана и курсора мышки из интерфейсных элементов больше ничего не было. После небольшого изучения я обнаружил контекстное меню ассоциированное с левой кнопкой “мыши”. Панель задач для этого менеджера окон заменяет TWM Icon Manager (см. рис. 7.9 вверху слева), а виртуальных рабочих столов, которыми так славится X-Windows нет и в помине. Переход фокуса между окнами очень непривычен. Так, щелчок средней кнопкой по заголовку окна перемещает его или на самый верхний уровень (поверх всех окон) или на самый низ (позади всех открытых окон). При вызове нового окна менеджер просит пользователя самостоятельно расположить его на экране, что, по моему мнению, только мешает работать. На первых минутах использования twm, даже закрыть окно стандартными средствами менеджера окон у меня не получилось. Правда, в главном меню я нашел пункты Kill и Delete, но появившийся после их вызова курсор в виде черепа с костями вызвал у меня ассоциацию больше с аварийным нежели со стандартным способом закрытия окна программы, хотя пункт “Delete” предназначен именно для нормального завершения работы. Выполненный со спартанским аскетизмом этот менеджер окон подойдет только для очень узкого круга пользователей. Читателям я бы порекомендовал один раз запустить этот менеджер окон затем чтобы убедиться в том, какой большой путь прошел

графический интерфейс UNIX систем за последнее время. Для реабилитации потенциальных возможностей оконного менеджера twm следует отметить, что он поддается очень тонкой настройке и конфигурации, так что при изучении документации и правке его конфигурационных файлов от него можно добиться уникальной производительности.

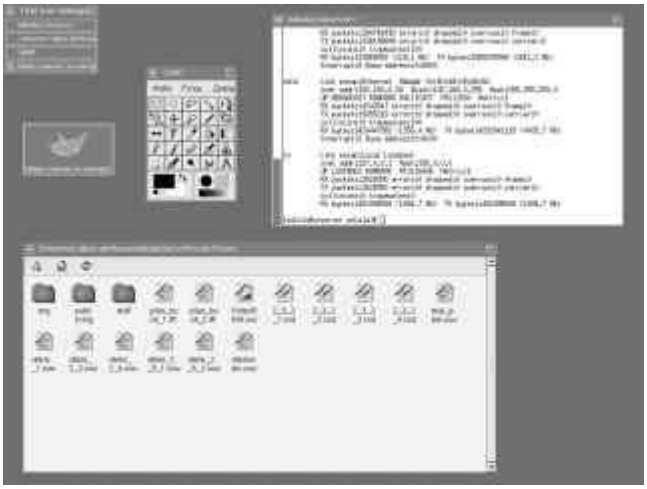

Рис. 7.9. Пример экрана менеджера окон twm запущенного на X-терминале

Менеджер окон mwm (Motif Window Manager) (см. рис. 7.10) – это большой шаг по сравнению с twm, но все же я бы поспешил назвать его полноценным современным оконным менеджером. Меня приятно порадовала поддержка стандартных клавиш управления окнами (Alt+Tab, Alt+F4 и т.д.), правда переключение по нажатию Alt+Tab приводит не к переходу к предыдущему окну, а к прокручиванию списка окон с изменением их порядка. Управление окнами очень напоминает интерфейс Windows 3.1, они точно также сворачиваются в иконки в нижнем левом углу рабочего стола (правда, все они почему-то в квадратных рамках). Очень не хватает этому менеджеру окон панели задач и полноценного меню. Также не помешала бы программа настройки его поведения. Отсутствие виртуальных рабочих столов тоже нельзя отнести к его преимуществам. По требованиям к оперативной памяти mwm нельзя причислить к разряду требовательных – чуть больше 8 Мбайт при 3 Мбайтах в разделяемой памяти. Если обобщить свои субъективные впечатления о этом менеджере окон, то на мой взгляд уж слишком в нем все просто, а необходимой пользователю функциональности как раз и не хватает, хотя может вам он и больше понравится.

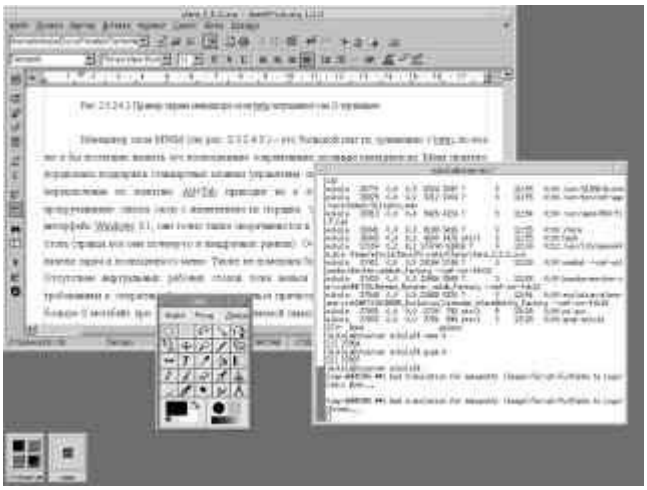

Рис. 7.10. Пример экрана менеджера окон mwm запущенного на X-терминале

Выбор оптимального программного обеспечения для работы на X-терминале (продолжение)

Победителем моего соревнования “облегченных” менеджеров окон стал IceWM (см. рис. 7.11). Конечно, в памяти, он занимает места довольно много (3 работающих программы суммарным размером в 16 Мбайт), зато оцените его преимущества:

Интерфейс очень напоминает Microsoft Windows 95 (там же расположена кнопка “Пуск”, рядом панель задач, справа внизу системный трэй и привычные часы);

Панель задач позволяет быстро переключаться между запущенными программами, и управлять их расположением на экране;

Есть панель кнопок быстрого запуска приложений;

Поддержка виртуальных рабочих столов. Причем можно настраивать их количество и присваивать им имена;

На главной панели есть поле ввода консольных команд;

Простой доступ к меню “Список окон” (специальная кнопка на панели задач или щелчок “мыши” по свободному месту рабочего стола);

Функция “свернуть/восстановить” все окна, которая позволяет мгновенно показать рабочий стол;

Встроенные индикаторы сетевой активности (все интерфейсы настраиваемые, включая даже /dev/ppp0 и т.п.) и загрузки центрального процессора;

Главное меню можно вызвать как с панели задач, так и по щелчку на рабочем столе;

Привычные сочетания клавиш для управления положением окон на экране;

Возможность установки на рабочий стол фонового рисунка;

Поддержка тем рабочего стола, с полной сменой внешнего вида окон (включая даже расположение и количество кнопок управления окном);

Стабильность в работе;

Быстрый запуск менеджера окон;

Тонкая настройка поведения менеджера окон через программу конфигурации или простым редактированием файлов.

Рис.7.11. Пример экрана менеджера окон IceWM запущенного на X-терминале

Когда я стал ближе разбираться с конфигурацией менеджера окон IceWM, то понял, что это очень гибкая система с возможностями тонкой настройки. Причем настройке подлежит практически все, начиная с внешнего вида окон и расположения кнопок на них, а заканчивая поведением окна при изменении его размеров, получении окном фокуса ввода и выборе клавиш быстрого доступа. Так, для X-терминалов я специально запретил прорисовку окон при их перемещении или изменении размеров, делается это путем редактирования строчек из файла /etc/icewm/preferences:

```
#  Opaque window move
OpaqueMove=0 # 0/1

#  Opaque window resize
```

OpaqueResize=0 # 0/1

Применить подобное ограничение я рекомендую всем, так как перемещение или изменение размеров окна приводит к резкому увеличению передачи графической информации через локальную сеть, а следовательно, падению быстродействия.

Конечно, у менеджера окон IceWM есть и недостатки, главным из которых я считаю отсутствие как такового рабочего стола. То есть, фоновую картинку вы конечно поменять сможете, а вот разместить на нем ярлычок нет. Также недостает этому оконному менеджеру и встроенной программы конфигурации, так как править файлы в текстовом редакторе не всегда удобно (особенно рядовому пользователю). Вообще, благодаря своим маленьким размерам (двоичный дистрибутив размером в 2 Мбайта) IceWM практически не имеет встроенных приложений, таких как текстовые редакторы, файловые менеджеры, почтовые клиенты и т.д., а следовательно, вам придется заимствовать их из других проектов или использовать обособленные разработки.

Рис. 7.12. Пример экрана менеджера окон XFce

Позже мне на глаза попал еще один перспективный менеджер окон XFce (см. рис 7.12). Правда назвать его совсем легковесным я бы не отважился, так как суммарно его программные компоненты занимают в памяти более 70 Мбайт, но зато очень большой процент в разделяемой памяти. К его сильным сторонам следует отнести стильный дизайн (чем то отдаленно напоминающий MacOS), удобную панель задач, управление виртуальными рабочими столами, доступную панель управления, встроенный файловый менеджер, хранитель

экрана, хорошую поддержку национального языка и многое другое. Этот файловый менеджер может стать неплохим кандидатом для установки на X-терминалах учебных заведений, так как в его визуальном оформлении чувствуется что-то новое, что должно способствовать эффективному процессу обучения.

Как видите, выбор оконных менеджеров довольно широкий и пользователи различных пристрастий могут выбрать себе тот, что лучше соответствует их понятию о работе с окнами. Значительная степень конфигурирования существующих менеджеров окон, позволяет ориентировать их для нужд конечных пользователей X-терминалов.

Советы по работе и настройке типового софта – OpenOffice, Gimp, Evolution и т.д.

Поскольку X-терминалы работают под управлением операционной системы Linux, то и прикладные программы также должны быть для этой операционной системы. Ошибочно полагать, что операционная система Linux рассчитана только для серверов и хакеров. Поверьте, что и для офисного работника в ней найдется достаточно хороших и надежных прикладных программ.

Файловый менеджер

Начнем наш обзор с выбора файлового менеджера – незаменимого инструмента в руках любого пользователя. С выбором файлового менеджера для менеджеров окон типа Gnome, KDE или XFce проблем не возникает, так как эти среды имеют собственный файловый менеджер, который, кстати, можно использовать и отдельно. Так, у Gnome он называется nautilus, у KDE – konqueror, а у FXce – xffm (см. рис. 7.13).

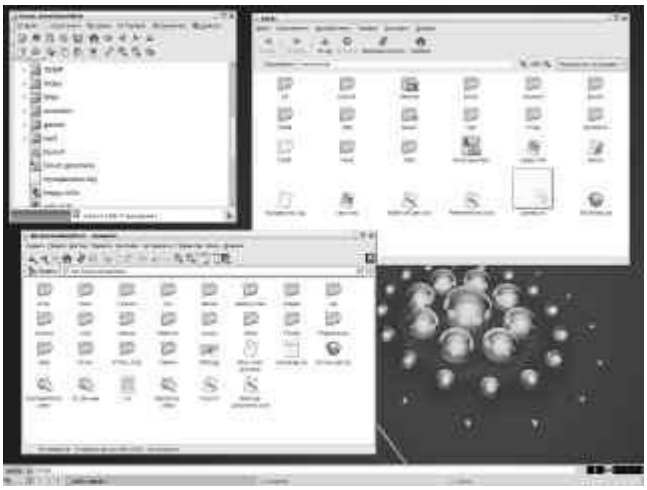

Рис. 7.13. Пример 3-х файловых менеджеров

Как мне кажется, самый продвинутый файловый менеджер – konqueror, но не следует забывать о его сильной интеграции со средой KDE. В памяти же он занимает 27 Мбайт, что, согласитесь, довольно много.

Лично для меня все файловые менеджеры, описанные выше, несколько тяжеловаты. Да, я согласен с их большой функциональностью, но рядовому пользователю нужен файловый менеджер проще, который быстро запускается и имеет простой интерфейс. К главным его функциям должны относиться:

открытие файлов (привязка типов MIME к ассоциированным программным продуктам);

копирование, перемещение, переименование и удаление файлов;

создание каталогов и новых файлов;

работа при помощи “мышки”;

изменение атрибутов файла;

настраиваемость как администратором так и пользователем;

поиск файлов.

После опробования многих файловых менеджеров в операционной системе Linux, я остановил свой выбор на ROX-Filer. Сразу оговорюсь, что это далеко не идеальный выбор, так как он в полной мере удовлетворяет только некоторым главным критериям файлового менеджер, но все же его преимущества были более ощутимые.

Рис. 7.14. Внешний вид файлового менеджера ROX-Filer

Кратко опишу свойства этого файлового менеджера. Во-первых, ROX-filer (см. рис. 7.14) – это очень быстрый и маленький файловый менеджер. В виртуальной памяти его размер около 9 Мбайт, а весь бинарный дистрибутив версии 1.2 всего 827 Кбайт. Внешний вид ROX-Filer очень напоминает “Мой компьютер” операционных систем win32 с отключенным показом веб-содержимого. Все файловые операции выполняются одним нажатием кнопки “мыши”, но предусмотрен также вариант с двойным щелчком. Файловый менеджер умеет копировать и перемещать файлы, поддерживает механизм Drag and Drop, умеет искать файлы по маске имени файла, а также по дополнительным опциям (правда, синтаксис поиска довольно сложный). Данный файловый менеджер полностью настраивается для работы администратором, а конкретный пользователь может использовать свои настройки. Имеется возможность изменить иконки файлов, типы MIME и связанные с данным типом прикладные программы, создать специальные обработчики по типу “Send To...”, а также показывать скрытые файлы и упорядочивать списки файлов по имени, типу, дате и размеру. Конечно, полнофункциональным файловым менеджером ROX-Filer назвать нельзя, так как, например, он не не обладает встроенными средствами просмотра файлов, имеет неудобные диалоговые окна копирования/перемещения файлов и с его помощью сложно управлять атрибутами файлов. К его недостаткам также следует добавить плохую поддержку русского языка (работает только в KOI8-R), отсутствие на панели инструментов кнопок переходов “Вперед” и “Назад”

(есть только “Вверх”) и отсутствие при копировании больших объемов информации (файлов) индикатора выполнения.

Мне понравилось в ROX-Filer то, что он очень тонко настраивается на работу с определенными типами файлов. Конфигурационные файлы сохраняются в 5-ти подкаталогах каталога Choices:

MIME-icons – содержит иконки для всех типов файлов (например, документам Microsoft Word соответствует икона application_msword.xpm);

MIME-info – описывается типы MIME, например:

application/x-compressed-tar

ext: tgz tar.gz

MIME-types – в этой папке каждому типу MIME сопоставляется файл запуска соответствующей программы, например, файл application_msword содержит:

#! /bin/sh

exec ooffice "$1"

ROX-Filer – здесь хранится один единственный файл Options, который содержит настройки ROX-Filer, которые можно выбрать через вызов пункта “Options...” контекстного меню;

SendTo – каждый файл в этом каталоге соответствует сценарию, который будет вызван при выборе пункта контекстного файлового меню “Send To...” .

Если же вы предпочитаете файловые менеджеры, которые работают в стиле легендарного Norton Commander, то возможно вас заинтересуют Midnight Commander, Krusader или Gnome-Commander.

Веб-браузер

Для работы во всемирной паутине X-терминалу необходим браузер. Благо, под операционную систему Linux браузеров написано предостаточно. Самый популярный из всех, пожалуй, Mozilla, хотя для браузера он на мой взгляд немного жаден к системным ресурсам. Так, после старта с пустой страничкой на экране он занимает в памяти 87 Мбайт. Конечно, есть и его облегченная версия FireFox, которая в последнее время имеет очень много сторонников. Не стоит сбрасывать со счетов и браузер среды KDE – konqueror.

Лично мне нравится браузер среды Gnome – Galeon, хотя он несколько тяжеловат (64 Мбайта в оперативной памяти – сказывается родственность с Mozilla), но все же проще в использовании чем Mozilla, хотя еще раз подчеркну, что выбор браузера – это дело вкуса и привычек.

Отдельное внимание следует уделить безопасности пользователей при работе в World Wide Web. В данном вопросе браузеры Linux лишены таких потенциально опасных компонентов как VBScript и ActiveX, очень многие из них умеют блокировать всплывающие окна PopUp, а также поддерживают апплеты-Java и сценарии JavaScript. По поводу использования Flash-анимации в X-терминальных решениях есть много “за” и много “против”. Возможно, суть проблемы скрыта в реализации Flash-плагинов для Linux-браузеров, но как факт остается то, что воспроизведение флэш роликов требует значительных затрат процессорного времени. Хорошо, если пользователь открыл только одну страничку с Flash-анимацией, но как правило, в закладках браузера пользователь обычно открывает до десятка страниц, каждая из которых может содержать flash-баннер или анимационную вставку. В итоге получаем довольно серьезную загрузку центрального процессора сервера X-терминалов, и, как следствие, падение производительности всех X-терминалов локальной сети. В общем, включать или не включать flash-анимацию зависит только от тех потребностей, которые преследуют пользователи, работая в Web и возможности вашего сервера. Лично я считаю, что Flash не заслуживает того процессорного времени, которое она потребляет, так как по большому счету это мигающие баннеры, вращающиеся логотипы или красивые кнопочки, без которых пользователь прекрасно сможет обойтись. К тому же, отсутствие поддержки Flash позволит уменьшить интернет трафик в локальной сети, так как его объекты не будут загружаться без активации соответствующего встраиваемого модуля. Будем надеяться, что в будущем ситуация с работой Flash-анимации улучшится.

Обсуждая вопрос выбора браузера для X-терминалов, я чуть не забыл о браузере Opera. Этот кроссплатформенный браузер поддерживает и Linux, так

что у его сторонников всегда будет возможность иметь под рукой привычный инструмент для путешествий в сети.

Работая на X-терминале с различными браузерами, я не заметил какого-либо замедления в работе или падения их производительности вызванного особенностями удаленной работы, поэтому можно утверждать, что для X-терминала подойдет практически любой браузер. Главное условие, чтобы браузер был удобен пользователю, ну а тут у браузеров в Linux есть все шансы.

Работая долгое время в одной операционной системе и с одним набором прикладных программ, любой пользователь постепенно привыкает к ним. Поэтому переход в Linux большинством пользователей воспринимается враждебно. Главные их аргументы - это фразы, похожие на: “Тут все так неудобно!”, “Как здесь вообще можно работать?” или “Почему я должен всякими глупостями забивать голову (осваивать новую операционную систему), когда мне нужно работать?”. Но, поработав в операционной системе Linux около месяца, они уже по-другому начинают воспринимать эту систему, и даже видеть в ней положительные аспекты, например, браузеры. Так, некоторые пользователи, которые на работе работают в Mozilla, а дома в Internet Explorer, стали и дома пользоваться браузером Mozilla для Windows.

Почтовый клиент

Как и с браузерами, в семействе почтовых клиентов для операционной системы Linux все в порядке. Их много и их возможности очень разнообразны. Очень неплохая почтовая программа входит в состав браузера Mozilla, особенно пользователи хвалят ее анти-спам фильтр. Также неплохими возможностями обладает почтовый клиент среды KDE – kmail. Для юниксоидов всегда найдется консольная программа mutt. Мне нравится работать в почтовой программе Ximian Evolution (см. рис. 7.15). Не могу сказать, что это супер почтовый клиент и меня все в нем устраивает, но его дружественный пользовательский интерфейс и широкие возможности помогают в полной мере использовать потенциал электронной почты. К недостаткам я бы отнес хранение почтовых данных в папке evolution домашнего каталога пользователя, выбор для этих целей скрытой папки был бы более оптимальным.

Единственной проблемой, которая была замечена при эксплуатации этого почтового клиента на X-терминалах, это завышенные требования к пропускной способности локальной сети при его старте, что может быть заметно при работе по низкоскоростной локальной сети.

Рис. 7.15. Рабочее окно почтового клиента Ximian Evolution

Клиент ICQ

Служба мгновенных сообщений ICQ имеет массу поклонников по всему миру. Несмотря на это, на ее официальном сайте http://www.icq.com недоступна версия для операционной системы Linux (есть только варианты для Microsoft Windows, Palm OS и MacOS). Конечно, пользователям Linux отчаиваться не стоит, так как в любом случае можно воспользоваться on-line версией ICQ-клиента, доступной на сайте http://go.icq.com, которая называется ICQ2GO или использовать в своей работе программное обеспечение сторонних разработчиков.

До недавнего времени самым популярным ICQ-клиентом для Linux была программа licq. Но в последнее время все больше пользователей отдают предпочтение другому программному продукту с загадочным названием sim, которое однако очень банально расшифровывается, как простой клиент службы мгновенных сообщений (Simple Instant Messenger).

Пользоваться программой sim до того просто, что это получится у любого пользователя без подготовки. Внешний вид окна контактов и сообщений

программы sim вы можете увидеть на рис. 7.16. К числу преимуществ клиента мгновенных сообщений sim следует отнести:

гибкую систему настройки под особенности работы пользователя;

работу программы в системном трее;

всплывающие сообщения на Рабочем столе о получении новых сообщений или появлении участников диалога в сети;

подключение к сети ICQ как напрямую, так и через прокси-сервер;

очень простая процедура автоматической регистрации нового пользователя.

Рис. 7.16. Клиент ICQ - SIM

Конечно, выбор клиента для отправки мгновенных сообщений зависит от многих факторов, на первом месте из которых удобство в работе пользователя. Рекомендую обратить также ваше внимание на еще одного представителя программных продуктов данного класса: клиента служб мгновенных сообщений Gaim. Так, в число его достоинств входит поддержка таких протоколов, как AIM (Oscar и TOC), ICQ, IRC, Yahoo!, MSN Messenger, Jabber, Gadu-Gadu, Napster, и Zephyr. Кроме того, Gaim имеет модульную структуру, которая позволяет значительно расширить его возможности. Например, он позволяет запускать сценарии написанные на языке Perl.

Пакет офисных приложений

Без текстового редактора и табличного процессора вряд ли можно представить работу в современном офисе. В мире win32 в этом плане все довольно спокойно, так как подавляющее большинство пользователей отдают предпочтение продукту от Microsoft – Microsoft Office, с его всемирно

известными Microsoft Word, Microsoft Excel, Microsoft Access, Microsoft Outlook и Microsoft PowerPoint. Лично я далек от абсолютизирования возможностей данного программного пакета, так как всем известна печальная история распространения макро-вирусов, которые и появились на свет благодаря Microsoft Office. Но следует признать, что это очень хороший пакет офисных приложений и конкурировать с ним на равных очень сложно. Подтверждает этот факт и то, что форматы файлов Microsoft Word (*.DOC) и Microsoft Excel (*.XLS) стали практически стандартами де-факто для текстовых и табличных документов.

CrossOver Office

Корпорация Microsoft не выпускает свой флагманский продукт для операционной системы Linux. Но тем не менее существует коммерческое решение, которое позволяет запускать популярный офисный пакет от Microsoft на Linux компьютере, и называется оно – CrossOver Office (сайт http://www.codeweavers.com/site/products/cxoffice/). Правда слово "коммерческое" подразумевает платное использование данного продукта, а именно на 14.12.2004 года ограниченное использование этого пакета одним пользователем стоило 39,95 $, а многопользовательская лицензия 74,95 $, да и покупку лицензии на сам Microsoft Office никто не отменял. К сожалению, у меня не было возможности проверить работу CrossOver Office в среде X-терминалов и оценить возможные перспективы ее использования. Следует добавить, что проект CrossOver Office позволяет также запускать в Linux многие программы для операционной системы Windows, в том числе Microsoft Project, Adobe Photoshop, Lotus Notes и т.д. Необходимо также помнить, что использование этого пакета подразумевает запуск программ win32 в режиме эмуляции, что в любом случае скажется на их производительности, по сравнению с родной операционной средой.

StarOffice

Корпорация Sun Microsystems предлагает пользователям операционной системы Linux альтернативный коммерческий офисный пакет StarOffice (сайт http://www.sun.com/software/star/staroffice/index.xml). На конец 2004 года

доступной была его 7 версия, по цене 75,95 $. В состав данного пакета входят программы для создания текстовых документов, электронных таблиц, рисунков, презентаций и работы с базами данных. StarOffice является кроссплатнформенным программным продуктом, так как доступен для Windows, Linux и Solaris. В его составе предусмотрен механизм использования шаблонов документов и автоматизация работы с помощью макросов. Пакет от Sun предлагает пользователю открытые стандарты хранения документов в формате XML (сжатом), а также качественные фильтры для работы с популярными форматами от Microsoft Office. Дополнительно присутствует возможность экспорта документов в формате PDF и сохранение презентаций в формате Macromedia Flash. Также StarOffice поддерживает двунаправленное письмо и языки народов Азии. Пакет StarOffice 7-й версии полностью совместим с форматами файлов StarOffice версий 5.x. Для X-терминала офисный пакет StarOffice будет достойной альтернативой, но не следует забывать о необходимости приобретения лицензии, а следовательно, в расчете экономической целесообразности эту сторону также необходимо учитывать.

KOffice

В состав среды KDE также входит неплохой офисный пакет – KOffice. Правда, очень функциональным я бы его не назвал, но для документооборота небольшой компании его должно с лихвой хватить. В его состав входят такие приложения:

KWord – текстовый процессор, который поддерживает работу с форматами KWord, MS Word, Word Perfect, AbiWord, Rich Text Format (RTF) и многими другими;

KSpread – табличный процессор, который умеет работать с такими популярными форматами электронных таблиц как MS Excel, Quattro Pro, Applix, CSV, Gnumeric и конечно KSpread;

KPresenter – средство создания презентаций, может работать с файлами MS Power Point;

KChart – автономная программа для создания диаграмм.

К недостаткам офисной среды KOffice, я бы все таки отнес неготовность работы с файлами очень большого размера. Так, если сравнивать запросы к памяти приложений KOffice и OpenOffice, то последние окажутся более чем в четыре раза скромнее. К тому же, к великому сожалению, KOffice еще не может похвалиться стабильностью в работе, которая так необходима любому офисному пакету.

Gnome Office

Проект Gnome не хочет отставать от KDE, поэтому есть даже такое понятие как Gnome Office, правда его приложения слабо объединены, но тем не менее, нам доступны табличный процессор – gnumeric и текстовый редактор – abiword. В своей практике, когда я только начал осваивать операционную систему Linux, я обратил внимание на эти два программных продукта. Мне понравился, во первых, их интерфейс (внешнее сходство с Word и Excel), а также скромность в потреблении системных ресурсов. Правда, возможности этих офисных программ пока оставляют желать лучшего. Особенно не достает им стабильности (как и аналогичным продуктам KOffice), а, как вы знаете, отсутствие именно этого качества часто является причиной излишней нервозности пользователей компьютера. Остается лишь надеяться, что в будущем ситуация изменится в лучшую сторону и у пользователей Linux появиться еще несколько надежных офисных приложений, так как потенциал есть как у Gnumeric так и AbiWord.

OpenOffice

Наверное, самый распространенный офисный пакет для Linux – это OpenOffice (сайт http://www.openoffice.org/). Базируясь на исходном коде StarOffice, он разрабатывается сообществом свободных программистов при спонсорской поддержке Sun Microsystems. Этот программный продукт распространяется под двумя лицензионными соглашениями, одно из которых GNU/GPL подразумевает его свободное использование, а, стало быть, это самое дешевое решение для организации офисной работы. Проект OpenOffice включает в себя 5 офисных программ:

OpenOffice Writer – текстовый редактор, аналог Microsoft Word;

OpenOffice Calc – табличный процессор, аналог Microsoft Excel;

OpenOffice Math – средство для создания математических формул;

OpenOffice Impress – пакет для создания презентаций;

OpenOffice Draw – программа для создания рисунков (векторный графический редактор).

OpenOffice может также похвастаться своей кросплатформенностью, так как его версии доступны для пользователей Linux, Windows, Solaris, Macintosh и FreeBSD, а доступность его исходного кода позволяет выполнить компиляцию и для большего числа операционных систем.Внешний вид окна OpenOffice представлен на рис. 7.17.

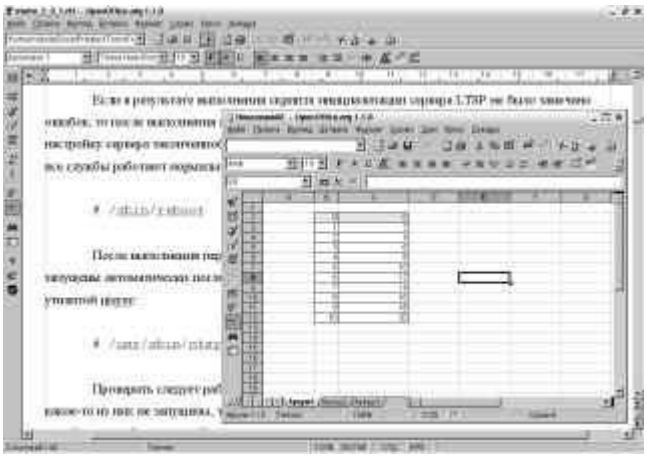

Рис. 7.17. Внешний вид офисного пакета OpenOffice

Конечно, OpenOffice тоже имеет свои недостатки: во-первых, он достаточно ресурсоемкое приложение. Так, OpenOffice Writer с открытым пустым текстовым документом занимает в виртуальной памяти 140 Мбайт, по-моему многовато для текстового редактора, а если включить поддержку орфографии у большого количества языков, то требования к памяти станут просто огромными. Совместимость с форматами Microsoft Office находится на очень высоком уровне, но все же имеются значительные недочеты. Запуск OpenOffice, открытие и сохранение документов происходят довольно медленно, и особенно это заметно на больших электронных таблицах, содержащих сотни листов. Стабильность приложений OpenOffice хоть и находится на серьезном уровне, но все же недостаточна для того, чтобы утверждать, что это на 100% устойчивый программный продукт.

Исходя из практического опыта работы в OpenOffice после использования Microsoft Office, можно утверждать, что практически все функции пакета от Microsoft можно выполнить средствами пакета OpenOffice. Хотя придется привыкнуть к новым горячим клавишам, расположению элементов меню и типу параметров в диалоговых окнах. У рядового пользователя на освоение функций работы в OpenOffice уходит примерно неделя.

При работе на X-терминале у пакета OpenOffice есть целый ряд преимуществ по сравнению с использованием его на традиционном персональном компьютере. Во-первых, требовательность к системным ресурсам подразумевает наличие у пользовательского компьютера как минимум 256 Мбайт оперативной памяти. Во-вторых, пользователю нужен быстрый процессор для комфортной работы с OpenOffice, что значительно увеличивает стоимость персонального компьютера. В X-терминальной модели, сервер и так имеет большой объем оперативной памяти и мощный процессор, следовательно требования OpenOffice для него обычны. К тому же, сервер обслуживает многопользовательскую среду, а стало быть использование памяти у него намного эффективнее. Рассмотрим данное утверждение более детально.

Если бы программы с внешнего носителя информации просто переписывалась бы в оперативную память и выполнялась там, то мы имели бы дело с очень простой и нерациональной схемой использования ресурсов компьютера. На самом деле программы в памяти компьютера состоят из программного кода, данных и стека. Если посмотреть на эти составляющие более внимательно, то можно заметить, что во-первых, данные подразделяются на изменяемые (переменные) и не изменяемые(константы). Стек, служит для хранения временных данных, таких, как адреса возврата из подпрограмм, локальных переменных, аргументов функций, а, следовательно, его содержимое уникально для каждого экземпляра программы. Код программы никогда не изменяется во время выполнения, а потому достаточно иметь один экземпляр кода в памяти, для всех запущенных программ. В погоне за эффективным использованием оперативной памяти компьютеров, создатели операционных

систем пошли еще дальше, и вместо использования статически скомпилированных программ, они придумали способ повторного использования кода путем создания разделяемых библиотек. В состав динамических разделяемых библиотек выносятся наиболее часто используемые функции, чтобы все программы, работающие в операционной системе, могли использовать их во время выполнения, а не включать их код в тело программы на стадии компиляции.

На примере офисного пакете OpenOffice давайте рассмотрим эффективность использования приложений в X-терминальной среде. Ниже приводится результат работы команды top на реальной X-терминальной системе, упорядоченный по используемой оперативной памяти (комбинация клавиш Shift + M):

```
$ top
 15:04:00  up 94 days,  5:42,  3 users,  load average: 0,13, 0,30, 0,50
188 processes: 166 sleeping, 22 running, 0 zombie, 0 stopped
CPU states:   1,5% user   0,5% system   0,0% nice   0,0% iowait  97,8% idle
Mem:  1030108k av,  966692k used,   63416k free,       0k shrd,   45172k buff
                    669796k actv,   29804k in_d,   18808k in_c
Swap: 2048248k av,  398488k used, 1649760k free                295592k cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME CPU COMMAND
 9690 ira       15   0  146M 144M 39216 S     0,3 14,3   2:12   0 soffice.bin
 8589 ula       15   0 94404  87M 28880 S     0,1  8,6   0:25   0 soffice.bin
 2451 sasha     15   0 84028  76M  3016 S     0,0  7,5 104:34   0 galeon-bin
 9769 serg      15   0 72700  63M 31816 S     0,3  6,2   0:12   0 soffice.bin
30548 sasha     15   0 88288  63M 27960 S     0,3  6,2   3:28   0 soffice.bin
 9624 nata      15   0 76056  61M 25368 S     0,0  6,1   0:11   0 soffice.bin
 6768 mikola    15   0 89932  56M 28600 R     0,0  5,5   0:52   0 soffice.bin
 6319 disp      15   0 31484  18M  3520 S     0,0  1,8   0:04   0 galeon-bin
 9741 ira       15   0 18632  15M  9780 S     0,0  1,5   0:03   0 galeon-bin
```

```
 8302 mikola    15   0 14120  13M  7108 S     0,0  1,3   0:02   0 evolution-mail
 9799 ira       15   0 14440  12M 10432 S     0,0  1,2   0:00   0 ark
 8120 mikola    15   0 12508  10M  8504 S     0,0  1,0   0:04   0 sim
 8296 mikola    15   0  9584 9552  4744 S     0,0  0,9   0:01   0 evolution
 9809 ira       15   0  9024 9024  7468 S     0,0  0,8   0:00   0 kdeinit
 9739 disp      15   0  9160 8092  5888 S     0,0  0,7   0:13   0 sol
 9807 ira       15   0  7736 7724  6488 S     0,0  0,7   0:00   0 kdeinit
 9804 ira       15   0  6904 6900  5688 S     0,0  0,6   0:00   0 kdeinit
 9839 ira       15   0  7156 6852  5880 S     0,0  0,6   0:00   0 kdeinit
 8318 mikola    15   0  6832 6776  3628 S     0,0  0,6   0:00   0 evolution-addre
 9801 ira       15   0  6564 6560  5392 S     0,0  0,6   0:00   0 kdeinit
 7491 ula       15   0  3228 3184  1996 S     0,0  0,3   0:00   0 wombat
10330 root      15   0  4520 3148   448 S     0,0  0,3   0:14   0 cupsd
 6912 mikola    15   0  3184 3032   960 R     0,0  0,2   0:00   0 xterm
  854 xfs       15   0  6148 3004   436 S     0,0  0,2   3:38   0 xfs
 8441 mikola    15   0  2928 2924  1732 S     0,0  0,2   0:00   0 wombat
...
```

Так как нас интересуют только процессы soffice.bin (OpenOffice), то заметим, что в системе их запущено 6. Столбец “SIZE” указывает на общий размер программы в памяти, то есть “код”+”данные”+”стек”. Исходя из этого показателя, только на запуск офисного пакета нужна память в размере: 146 + 92 + 71 + 86 + 74 + 88 = 557 Мбайт. Но, если взять во внимание значение столбца “SHARE”, которое равняется объему памяти в разделяемых динамических библиотеках, то реальная память под офисный пакет OpenOffice сократится до 557 - (38 + 28 + 31 + 27 + 25 + 28) = 380 Мбайт. Итого, имеем экономию оперативной памяти почти 32%, а с увеличением количества пользователей эта экономия стает еще более ощутимой.

OpenOffice я бы назвал главным офисным приложением, так как без текстового редактора Writer и табличного процессора Calc, очень сложно работать. И в моей практике был случай, когда после 4-х месяцев исправной работы на X-терминалах эти приложения стали, мягко выражаясь, “безбожно

тормозить”. Пользователи начали жаловаться, что они ждут открытия файла по несколько минут, и что печать на принтере тоже осуществляется очень медленно. Начав анализировать сложившуюся ситуацию, я стал замечать, что, действительно, на запуск любого из приложений OpenOffice версии 1.1.0. тратится больше 15 секунд, а также при открытии и печати файлов возникает пауза в 1-2 минуты, в результате которой загрузка процессора держится на отметке в 100%, причем занят процессор именно процессом soffice.bin. В поисках возможного решения данной проблемы я обнаружил, что не у всех пользователей возникли такие неприятности. Так, пользователи, которые недавно стали работать на X-терминалах, подобных временных задержек не испытывали, а опираясь на эту информацию я сделал вывод, что в процессе работы пользователя приложения OpenOffice собирают какую-то информацию, на анализ которой потом и тратится процессорное время, что, безусловно, мешает нормальной работе пользователей. Исходя из такой предпосылки, я начал сравнивать содержимое каталогов ~/.openoffice у пользователей, у которых были временные задержки, и тех у которых их не было. В конфигурационных файлах принципиальных отличий мне найти не удалось, а вот размер файла ~/.openoffice/user/psprint/pspfontcache у некоторых пользователей меня приятно порадовал, так как достигал 8 Мбайт. Наблюдалась интересная закономерность: чем больше был этот файл, тем медленнее была работа приложений OpenOffice. Открыв содержимое этого файла, я увидел нечто похожее на:

FontCacheDirectory:1077711251:/usr/X11R6/lib/X11/fonts/Type1

File:l047036t.pfa

1;1051256346;71

Luxi Mono

1088236480;;2;8;5;1;1;0;0;0;0;0;0;0;0;0;l047036t.afm

Luxi Mono

1088236480;;2;8;5;1;1;0;0;0;0;0;0;0;0;0;l047036t.afm

Luxi Mono

1088236480;;2;8;5;1;1;0;0;0;0;0;0;0;0;0;l047036t.afm

Luxi Mono

1088236480;;2;8;5;1;1;0;0;0;0;0;0;0;0;l047036t.afm

...

Судя по названию и содержимому файла, это своеобразных кэш шрифтов, которые использовались в документах OpenOffice и установлены в системе. Не долго думая, я переместил этот файл в другую временную папку и снова запустил OpenOffice. Офисный пакет запустился практически мгновенно, также быстро он открывал файлы и печатал их на принтере. Самое интересное то, что файл ~/.openoffice/user/psprint/pspfontcache был создан снова, но теперь он был значительно скромнее в размерах, всего навсего 13,5 Кбайт. Из главы "Администрирование сети X-терминалов" вы узнаете, как настроить систему на автоматическое удаление этого файла при увеличении его размера.

OpenOffice (продолжение)

История личной оптимизации пакета OpenOffice началась после выхода в свет версии 1.1.0, тогда я как раз устанавливал новый сервер X-терминалов. Компьютер был достаточно мощный (Intel Pentium 2.4 GHz cache L2 512 Mb/ FSB 800 MHz / 1 Gb RAM) и я вправе был думать, что OpenOffice будет на нем летать, но не тут-то было. После установки, которая прошла с небольшими проблемами (не оказалось нужного пакета db4-4.1 для работы с базами данных, который пришлось скачивать из Интернета), я запустил новую версии этого офисного пакета. Первое разочарование было при запуске программы ooffice – она запускалась более 10 секунд, но я отнес такую задержку на создание первичных пользовательских настроек. Далее я увидел главное окно программы OpenOffice, в котором все элементы меню были написаны квадратиками вместо букв. Понятно, что работать в таком виде с офисным пакетом было нельзя, а единственным выходом было внесение изменений в его конфигурацию.

Изменения конфигурации я начал с предположения, что авторы ASPLinux не могли выложить на свой официальный сайт не проверенный офисный пакет, следовательно конфигурация моей системы должна чем-то отличаться от той, на которой его проверяли. Самым главным отличием, на мой взгляд, были локальные настройки, так как у меня выбран системным украинский язык

(uk_UA.CP1251), а у разработчиков ASPLinux – русский. Тут следует оговориться, что поддержка украинского языка в ASPLinux находится на достаточно хорошем уровне, но, конечно, уступает аналогичной поддержке русского языка. Недолго думая, я запустил OpenOffice в режиме русской системной локализации:

$ LANG=ru_RU.cp1251 ooffice

Он запустился прекрасно и все пункты меню и элементы диалоговых окон были на русском языке. Единственно, что мне не понравилось, так это шрифты по умолчанию в офисных документах и элементах интерфейса.Я начал постепенно изучать содержимое каталога /usr/lib/openoffice и нашел очень интересный документ /usr/lib/openoffice/share/registry/data/org/openoffice/VCL.xcu. Ниже приведен фрагмент данного документа, который отвечает за используемые шрифты для системной локализации ru-RU:

Nimbus Sans L;Albany;Arial;Luxi Serif;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Nimbus Sans L;Albany;Arial;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Nimbus Mono L;Cumberland;Courier New;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Nimbus Roman No9 L;Thorndale;Times New Roman;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Nimbus Sans L;Albany;Arial;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Nimbus Roman No9 L;Thorndale;Times New Roman;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Nimbus Sans L;Albany;Arial;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Nimbus Roman No9 L;Thorndale;Times New Roman;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Nimbus Mono L;Cumberland;Courier New;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Nimbus Mono L;Cumberland;Courier New;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Nimbus Sans L;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma;Luxi Sans;Interface User;WarpSans;Geneva;Tahoma;MS Sans Serif;Helv;Dialog;Albany;Lucida;Helvetica;Charcoal;Chicago;Arial;Helmet;Interface System;Sans Serif

Как видно из списка шрифтов, которые используются в офисном пакете, на первом месте всегда стоят шрифты Nimbus (Nimbus Sans L, Nimbus Mono L и Nimbus Roman No9 L). Многим привычнее работать со шрифтами Arial, Courier New и Times New Roman, поэтому добавляем к данному файла раздел для не представленной в нем системной локализации uk-UA (для русскоязычных пользователей можно изменить уже существующий раздел):

Arial;Nimbus Sans L;Albany;Luxi Serif;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Arial;Nimbus Sans L;Albany;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Courier New;Nimbus Mono L;Cumberland;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Times New Roman;Nimbus Roman No9 L;Thorndale;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Arial;Nimbus Sans L;Albany;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Times New Roman;Nimbus Roman No9 L;Thorndale;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Arial;Nimbus Sans L;Albany;Luxi Sans;Helvetica;Lucida;Geneva;Helmet;SansSerif;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma

Times New Roman;Nimbus Roman No9 L;Thorndale;Luxi Serif;Times;Lucida Serif;Lucida Bright;Timmons;New York;Serif

Courier New;Nimbus Mono L;Cumberland;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Courier New;Nimbus Mono L;Cumberland;Luxi Mono;Courier;Lucida Sans Typewriter;Lucida Typewriter;Monaco;Monospaced

Arial;Nimbus Sans L;Andale Sans UI;Arial Unicode MS;Lucida Sans Unicode;Tahoma;Luxi Sans;Interface User;WarpSans;Geneva;Tahoma;MS Sans Serif;Helv;Dialog;Albany;Lucida;Helvetica;Charcoal;Chicago;Helmet;Interface System;Sans Serif

Запускаем OpenOffice и смотрим на знакомые очертания привычных шрифтов. Более того, в моем случае OpenOffice заработал в украинской локализации и никаких квадратиков в меню!

Разобравшись с элементами интерфейса я понял, что на этом оптимизация настройки OpenOffice не заканчивается. Очень разочаровала меня скорость проверки орфографии. Так, при первом запуске текстового редактора OpenOffice я щелкнул на слове с ошибкой (подчеркнутом красной волнистой линией) правой кнопкой “мыши” и вынужден был ждать 10 секунд, пока на экране появится контекстное меню с вариантами правильно написанных слов (при этом загрузка центрального процессора была на отметке в 100%). Конечно, при повторном обращении за помощью к компьютерному словарю эта операция была выполнена довольно быстро, но при проверке орфографии размер программы в памяти достиг 223 Мбайта, и это при 33 Мбайтах при запуске. Углубившись в изучения ситуации поддержки орфографии в OpenOffice версии 1.1.0. я обнаружил, что он поддерживает проверку орфографии, расстановку переносов и тезаурус у довольно большого количества языков, причем большинство из которых не представляет интереса для конечных пользователей. Согласитесь, редкий человек будет составлять документы одновременно на русском, украинском, английском, немецком и, например, итальянском языках, а список поддерживаемых языков намного больше. Для сравнения хочу привести суммарный размер файлов словарей для OpenOffice версии 1.1.0. - более 44 Мбайт. Я посчитал, что будет рациональным отключить проверку орфографии некоторых языков, которыми не владеют пользователи. Для этого в файле /usr/lib/openoffice/share/dict/ooo/dictionary.lst достаточно поставить символы “#” перед неиспользуемыми словарями. Чтобы еще увеличить эффект я также запретил использование тезауруса и автоматической расстановки переносов, так как эти функции мало востребованные. В результате у меня получился такой файл (для проверки орфографии трех языков: русского, украинского и английского США):

```
# List of All Dictionaries to be Loaded by OpenOffice
# ---------------------------------------------------
# Each Entry in the list have the following space delimited fields
#
# Field 1: Entry Type "DICT" - spellchecking dictionary
#                    "HYPH" - hyphenation dictionary
#                    "THES" - thesaurus files
#
# Field 2: Language code from Locale "en" or "de" or "pt" ...
#
# Field 3: Country Code from Locale "US" or "GB" or "PT"
#
# Field 4: Root name of file(s) "en_US" or "hyph_de" or "th_en_US
#          (do not add extensions to the name)

#HYPH ru RU hyph_ru_RU
#DICT it IT it_IT
#HYPH it IT hyph_it_IT
#HYPH de DE hyph_de_DE
#DICT en GB en_GB
#HYPH en GB hyph_en_GB
#HYPH da DK hyph_da_DK
DICT en US en_US
#HYPH en US hyph_en_US
#THES en US th_en_US
#DICT ca ES ca_ES
#DICT cs CZ cs_CZ
#DICT da DK da_DK
#DICT de CH de_CH
#DICT de DE de_DE
#DICT el GR el_GR
```

```
#DICT en CA en_CA
#DICT es ES es_ES
#DICT fr FR fr_FR
#DICT hr HR hr_HR
#DICT hu HU hu_HU
#DICT it IT it_IT
#DICT nl NL nl_NL
#DICT pl PL pl_PL
#DICT pt PT pt_PT
#DICT sv SE sv_SE
#DICT bg BG bg_BG
#DICT en GB en_GB
#DICT pt BR pt_BR
#DICT sk SK sk_SK
#DICT ga IE ga_IE
#DICT gl ES gl_ES
#DICT lt LT lt_LT
DICT ru RU ru_RU
#DICT sl SI sl_SI
DICT uk UA uk_UA
#DICT no NO no_NO
#DICT la ANY la
#HYPH fr FR hyph_fr
#HYPH it IT hyph_it
#HYPH nl NL hyph_nl
#HYPH cs CZ hyph_cs
#HYPH en CA hyph_en_CA
#HYPH en GB hyph_en_GB
#HYPH es ES hyph_es
#HYPH pt BR hyph_pt_BR
#HYPH pt PT hyph_pt_PT
```

#HYPH sk SK hyph_sk

#HYPH sl SI hyph_sl

#HYPH uk UA hyph_uk

Запускаем после “орфографической” оптимизации OpenOffice. Проверяем размер программы: 33132 Кбайт, из них 25844 Кбайт в общих библиотеках (Share). Создаем чистый документ, при этом размер программы в памяти увеличивается до 42232 Кбайт и 33288 Кбайт в разделяемых библиотеках. Набираем текст с ошибкой – память программы увеличивается до 61248 Кбайт и 35476 Кбайт (Share). Теперь нажимаем на слове с ошибкой правой кнопкой “мыши”. Меню вариантов появляется моментально, и в памяти программа занимает всего 72504 Кбайт (35596 Кбайт). Согласитесь есть разница между размерами программы в 72 Мбайта после оптимизации проверки орфографии и 223 Мбайта без оптимизации. Таким образом, снизив потребность в памяти для OpenOffice более чем в 3 раза, мы добились более эффективного расходования системных ресурсов сервера, а тем самым обеспечили более комфортные условия для пользователей.

Графический редактор Gimp

В Linux графический редактор Gimp (см. рис. 7.18) настолько популярный продукт, что не слышал про него разве что только совсем начинающий пользователь. Иногда его даже приводят в качестве аналога Adobe Photoshop для платформы Linux. По моему, такое утверждение несколько преувеличивает достоинства Gimp и занижает характеристики Photoshop, но тем не менее Gimp остается полнофункциональным растровым графическим редактором.

Рис. 7.18. Внешний вид графического редактора GIMP

Архиватор

Для операционной системы Linux стандартными являются программы архивирования и сжатия, которые она унаследовала от UNIX-систем: tar и gzip. Архивация с их помощью файлов и каталогов проблем не представляет, правда есть один нюанс: требуется умение работать в командной строке, а добиться этого от обычного пользователя задача практически нереальная. К тому же, очень большое распространение получили форматы ZIP и RAR, а для их разархивирования нужны совсем другие утилиты (unzip, unrar). Поэтому желательно использовать программу для архивирования (разархивирования) с графическим интерфейсом. Некоторые файловые менеджеры предоставляют возможность работать с архивами как с папками, например, Konqueror и Midnight Commander, что также очень удобно. Из изученных мною программ архиваторов под Linux мне больше всего понравилась функциональность Ark (архиватор среды KDE). Эта утилита умеет работать со всеми популярными форматами архивных файлов, но сама является лишь интерфейсом к различным архиваторам, установленным в системе. Так в версии 2.1.9 она отказывалась работать с архивами RAR, мотивируя отсутствие в системе установленного пакета RAR. Но, так как пакет unrar присутствовал, то небольшая “хитрость” помогла Ark работать с файлами с расширением *.rar:

```
# cd /usr/bin
# ln -s unrar rar
```

Теперь Ark прекрасно открывает файлы, созданные в архиваторе RAR.

Использование архивных файлов, созданных пользователями в операционных системах семейства MS Windows, сопряжено с еще одной скрытой проблемой: буквами национальных алфавитов в именах файлов сжатых в архиве. "Уши" этой проблемы растут, с одной стороны, от двойственного характера хранения имен файлов в одной кодировке системами от Microsoft , а их содержимого - в другой (для русских версий Windows файлы именуются в кодировке cp866, а кодовая страница системы по умолчанию cp1251), а с другой стороны, отсутствием в архивных файлах такого понятия, как кодировка имен файлов, то есть после архивирования эта информация

считается полностью утерянной. В результате получаем абракадабру в названиях файлов, состоящих из русских букв. Найти архиватор, который бы мог изменять кодовую страницу имен файлов в архиве, мне пока не удалось.

Дополнительные параметры файла lts.conf

Довольно часто настройка локальной сети X-терминалов требует более гибкого подхода, нежели поведение по умолчанию, предусмотренное проектом LTSP. Так, главный конфигурационный файл /opt/ltsp/i386/etc/lts.conf может содержать такие дополнительные настройки для каждого X-терминала:

SYSLOG_HOST – этот параметр указывает, какой из компьютеров должен принимать и обрабатывать сообщения демона syslogd данного X-терминала. По умолчанию для этих целей используется сервер X-терминалов. Использование данной функции может быть оправданно в тестовых целях. Параметр подразумевает IP-адрес, например, SYSLOG_HOST = 192.168.1.50.

TELNET_HOST – используется для задания альтернативного хоста, к которому будет подключаться терминал в режиме Telnet-сессии (RUNLEVEL = 4). По умолчанию TELNET_HOSTсовпадает со значением параметра SERVER.

TELNET_SESSIONS – этот параметр также связан с работой X-терминала в режиме Telnet-сессии (RUNLEVEL = 4). Он определяет сколько сессий необходимо запустить, при этом каждая сессия будет располагаться в отдельном виртуальном экране. Для перехода между виртуальными экранами можно использовать комбинации клавиш Alt+F1, Alt+F2, и т.д.

DNS_SERVER – этот параметр используется для построения файла resolv.conf на терминале. Если он непосредственно не установлен, то используется значение по умолчанию из параметра SERVER.

SEARCH_DOMAIN – также нужен для построения файла resolv.conf на терминале. Значение по умолчанию отсутствует.

MODULE_01 – определяет модуль ядра, который необходимо загрузить утилитой insmod. Всего можно использовать до 10 подгружаемых модулей опциями от MODULE_01 до MODULE_10. В основном используется для работы с дополнительным аппаратным обеспечением X-терминала, таким как USB-мышь, USB-принтер, звуковая карта и т.п. Ниже приведен классический

пример использования AGP-видеоадаптера и звуковой карты ISA на чипсете opl3:

MODULE_01 = agpgart.o

MODULE_02 = uart401.o

MODULE_03 = sb.o io=0x220 irq=5 dma=1

MODULE_04 = opl3.o

RAMDISK_SIZE – при запуске X-терминала создается виртуальный диск в оперативной памяти и монтируется как каталог /tmp. Размер этого виртуального диска устанавливается в килобайтах, например, RAMDISK_SIZE = 4096, соответствует виртуальному диску в 4 Мбайта. Если вы меняете этот параметр, то соответствующие изменения нужно сообщить также ядру X-терминала. Это можно сделать путем его перекомпиляции, или через параметры ядра в загрузчике Etherboot. Также размер виртуального диска можно передать через параметр option-129 файла /etc/dhcpd.conf. По умолчанию, размер виртуального диска равен 1 Мбайт: RAMDISK_SIZE = 1024.

NFS_SERVER – определяет сервер для монтирования файловой системы (/usr, /home). Если параметр не указан, то используется значение по умолчанию SERVER. Позволяет строить более гибкую распределенную систему, особенно при использовании локального запуска приложений на X-терминалах.

USE_NFS_SWAP – если значение данного параметра равно “Y”, то это означает использование файла подкачки через локальную сеть. Особенно может быть полезно, когда X-терминал имеет в своем распоряжении очень мало оперативной памяти (например, 8 Мб). По умолчанию USE_NFS_SWAP = N. Рекомендуется использовать этот параметр даже для рабочих станций с 32 Мбайтами оперативной памяти. Наиболее распространенной причиной необходимости использования сетевого файла подкачки является проблема так называемого "сброса" работающего X-терминала к графическому приглашению ввода пользователя и пароля. Более детально про эту и другие связанные проблемы вы сможете узнать в главе "Администрирование сети X-терминалов".

SWAP_SERVER – файлы подкачки могут располагаться на любом сервере в локальной сети, значение SWAP_SERVER указывает IP-адрес сервера для их хранения. По умолчанию используется значение параметра NFS_SERVER.

NFS_SWAPDIR – на сервере этот параметр указывает расположение каталога, для хранения файлов подкачки. По умолчанию используется директория /var/opt/ltsp, но может быть использована любая другая. Важно помнить, что при изменении этого параметра следует также внести соответствующие изменения в файл /etc/exports и перезапустить демон nfs.

SWAPFILE_SIZE – позволяет определить размер файла подкачки для X-терминала. Значение по умолчанию SWAPFILE_SIZE = 64m (64 Мбайт). После изменении этого значения при следующей загрузки X-терминала будет создан новый файл подкачки, и в зависимости от его размера на это может понадобиться от нескольких секунд до нескольких минут.

RCFILE_01 – хороший способ добавить функциональности X-терминалам, используя от одного до десяти пользовательских сценариев, которые автоматически будут выполнены при загрузке. Можно использовать параметры от RCFILE_01 до RCFILE_10. Примером такого файла может служить RCFILE_01 = floppyd, который помогает использовать на X-терминале флоппи дисковод. Следует помнить, что выполняться такие сценарии должны достаточно быстро и не иметь в себе бесконечных циклов, иначе загрузка X-терминала остановится на выполнении одного из таких файлов.

SOUND – при установленном пакете поддержки звука на X-терминалах можно использовать параметр SOUND = Y, для активации звуковой карты рабочей станции. При использовании ISA-устройств придется также вручную подгружать модули ядра для них, а с настройкой звуковых карт с интерфейсом PCI система LTSP должна справиться сама.

LOCAL_APPS – по умолчанию использование локального запуска приложений отключено (LOCAL_APPS = N). Эта дополнительная возможность применяется для того, чтобы более эффективно использовать аппаратные ресурсы X-терминала. Локальный запуск приложений сопряжен с

необходимостью дополнительной настройки системы LTSP, в частности обеспечению способа авторизации пользователей X-терминалов.

NIS_DOMAIN – этот параметр имеет смысл только при использовании локального запуска приложений (LOCAL_APPS = Y). Он определяет имя домена для NIS сервера. Значение по умолчанию “ltsp”.

NIS_SERVER – по умолчанию через широковещательные запросы X-терминал сам находит NIS-сервер в локальной сети. Но можно также и жестко задать его IP-адрес.

PRINTER_0_DEVICE – указывает символическое устройство, к которому подключен принтер X-терминала. Всего может быть определено 3 принтера для одной рабочей станции параметрами: PRINTER_0_DEVICE, PRINTER_1_DEVICE и PRINTER_2_DEVICE. Наверное, самым популярным значением будет /dev/lp0, которое соответствует принтеру, подключенному к параллельному порту (LPT1: для DOS и Windows). Всвязи с ростом популярности USB-принтеров, для них используется значение /dev/usb/lp0, правда, его использование требует загрузки дополнительных модулей ядра (как правило, usb-uhci и printer).

PRINTER_0_TYPE – этим параметром определяется тип принтера. “P” - означает параллельный интерфейс для устройств типа /dev/lp0, /dev/lp1 и т.п., а “S” - означает последовательный доступ для устройств типа /dev/ttyS0, /dev/ttyS1 или /dev/usb/lp0. Как и для любого параметра принтер,а можно устанавливать значения для 3-х принтеров, что подключены к X-терминалу.

PRINTER_0_PORT – определяет, какой из портов TCP/IP будет использоваться принтером. По умолчанию, 9100. Требует обязательной установки, если используется более одного принтера для одной рабочей станции, так как значения по умолчанию для PRINTER_1_PORT = 9101, а для PRINTER_2_PORT = 9102.

PRINTER_0_SPEED – этот параметр применим только для принтеров с последовательным интерфейсом. Значение по умолчанию 9600 бод. Для каждого из 3-х принтеров можно установить отдельное значение.

PRINTER_0_FLOWCTRL – определяет тип контроля передачи данных и применим только для принтеров с последовательным интерфейсом. По умолчанию используется значение “S”, которое соответствует “Software (XON/XOFF)”. Опционально можно выбрать также “H” - “Hardware (CTS/RTS)”.

PRINTER_0_PARITY – данный параметр применим также только для последовательных принтеров, и определяет тип контроля четности. Допустимые значения: “E” - Even, “O” - Odd или “N” - None. Значение по умолчанию PRINTER_0_PARITY = N. Применимо к каждому принтеру отдельно.

PRINTER_0_DATABITS – также применимо только для последовательных принтеров. Означает номер бита данных, может принимать значения: 5, 6, 7 или 8. Значение по умолчанию 8. Для каждого из 3-х доступных принтеров можно устанавливать независимые значения.

PRINTER_0_WRITE_ONLY – некоторые принтеры могут определять, что порт, к которому они подключены, поддерживает двунаправленный обмен данными. И в своей работе они могут использовать эту возможность, то есть ожидать обратного ответа от драйвера принтера через этот порт. Так как система печати операционной системы Linux не поддерживает двунаправленный обмен данными между принтером и компьютером, то установка этого параметра в “Y” позволит запретить принтеру такой обмен. Значение по умолчанию “N”.

Глава 8. Администрирование сети X-терминалов

Администрирование локальной сети X-терминалов поначалу сложнее, а потом проще чем обслуживание локальной сети, состоящей из персональных компьютеров. Кажущаяся сложность скрывается за необычностью администрирования бездисковых терминалов, а упрощение администрирования в дальнейшей работе – это заслуга однотипных конфигураций X-терминалов и централизованного доступа к их настройке. Для успешного внедрения X-терминалов от системного администратора требуется проведение разъяснительной работы с конечными пользователями. При этом не стоит

вдаваться в технические подробности реализации терминальной среды, а, скорее всего, лучше подойдет краткая инструкция, в которой “понятным” языком будут описаны характерные особенности работы на X-терминале.

Благодаря специфичности запуска терминальных приложений и их выполнению на сервере, от администратора требуется также выполнение профилактических работ по анализу загруженности локальной сети и распределению процессорного времени. Если своевременно не позаботиться об этом, то рабочий день системного администратора будет потрачен на выслушивание жалоб пользователей по поводу низкой производительности их рабочих станций.

Информация, представленная далее в данном разделе, поможет вам успешно справиться с наиболее распространенными проблемами при использовании X-терминалов совместно с проектом LTSP.

Если что-то не работает – порядок выполнения диагностики

Для того, чтобы вам было проще ориентироваться в этой главе, всевозможные проблемы в работе X-терминалов сгруппированы по причине их возникновения. В зависимости от того, в какой момент работы X-терминала случилась ошибка, ее решение следует искать в одном из перечисленных ниже разделов:

Проблемы загрузочного образа

Неполадки DHCP

Проблемы с TFTP

Устранение проблем с NFS

Проблемы X-сервера

Проблемы менеджера дисплея

Ошибки при работе в графическом режиме

Итак приступим к детальному рассмотрению наиболее популярных проблем в работе LTSP и X-терминалов.

Проблемы загрузочного образа

Под загрузочным образом, мы будет иметь в виду образ Etherboot на дискете (компакт-диске, “винчестере” и т.п.) или в микросхеме BootROM

сетевого адаптера, а также загрузку штатными способами PXE-сетевой карты. Так как инициализация загрузочного образа X-терминала это самая первая стадия его работы, то необходимо, чтобы она прошла без ошибок. К несчастью, иногда такие ошибки случаются. Наиболее распространенная из них выглядит так:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:45:2B:3F:46

Searching for server (DHCP) .....

Загрузка X-терминала останавливается на строчке “Searching for server (DHCP) ...”. Данное сообщение указывает на то, что с загрузочным образом все в порядке и он успешно загрузился, но возникла проблема с подключением к DHCP-серверу. Очень часто такие ошибки появляются после того, как пользователь самостоятельно передвигал свой X-терминал либо "убирал" провода, которые мешали ему работать. Наиболее вероятная причина возникновения данной ошибки – выдернутый сетевой кабель из гнезда сетевого адаптера. Интересно отметить, что подключение кабеля не решает данной проблемы, а X-терминалу требуется принудительная перезагрузка. Если после проверки кабеля (при включении X-терминала на сетевой карте должны гореть индикаторы) данная проблема осталась нерешенной, то вам следует посмотреть возможную причину в разделе “Неполадки DHCP”.

В результате успешного запуска загрузочного образа Etherboot на экране X-терминала вы должны увидеть что-то похожее на:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.5. (GPL) http://etherboot.org Tagged ELF for [W89C840F]

Relocation _text from [00013e20, 00024230) to [01eefb70, 01f00000)

Boot from (N)etwork or (Q)uit?

Если вы увидели строчку “Boot from (N)etwork or (Q)uit?”, значит образ загрузился успешно. На данном этапе еще неизвестно соответствует ли образ сетевому адаптеру и будет ли правильно выполнен процесс загрузки ядра операционной системы Linux через локальную сеть, но с дискеты или микросхемы BootROM он прочитан без ошибок.

Иногда при загрузке X-терминала (особенно часто при использовании дискеты) на экране быстро пробегают строки, похожие на эти:

Loading ROM image .....

08000 AX:0212 BX:B200 CX:0201 DX:0100

08000 AX:0212 BX:B200 CX:0201 DX:0100

08000 AX:0212 BX:B200 CX:0201 DX:0100

...

Такое сообщение появляется при ошибках чтения загрузочного образа. Если загрузка производилась с дискеты: запишите загрузочный образ на новую дискету и повторите загрузку. Если это не поможет, попробуйте заменить флоппи-дисковод. Иногда ошибки чтения с дискеты приводят даже к появлению серого текстового экрана с немигающим белым курсором. Чтобы исключить причины возникновения таких ошибок, рекомендуется в X-терминалах использовать заведомо исправные дисководы и дискеты, а в будущем вообще отказаться от последних в пользу BootROM сетевых адаптеров.

Об ошибочном образе загрузки (несоответствующем сетевому адаптеру), могут свидетельствовать такие сообщения на экране:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.5. (GPL) http://etherboot.org Tagged ELF for [W89C840F]

Relocation _text from [00013e20, 00024230) to [01eefb70, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

Probing isa nic ...

...

В данном случае сетевой загрузчик пытается инициализировать PCI, а затем и ISA-устройства, но его попытка заканчивается неудачей, о чем свидетельствует строка “”. Если подобная ошибка случилась при использовании дискеты, то это полбеды, так как вам достаточно просто переписать правильный загрузочный образ. Хуже, если такое случилось при загрузке через BootROM, так как это означает, что микросхема EPROM испорчена и нужно прошивать новую.

Для того чтобы в процессе выяснения причин возникновения ошибки исключить проблемы загрузочного образа, вы должны на экране увидеть физический или, как его еще называют, MAC-адрес сетевого адаптера. Например, в приведенных ниже сообщениях при загрузке можно с большой долей вероятности утверждать, что загрузочный образ соответствует микросхеме сетевого адаптера:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:43:2F:6A:CD

О правильности выбора образа загрузки, в частности, говорит MAC-адрес устройства (00:02:43:2F:6A:CD).

Неполадки DHCP

После правильной инициализации загрузочного образа он посылает через сетевой интерфейс широковещательные DHCP-запросы, в надежде найти работающий в локальной сети DHCP-сервер. Если вашему X-терминалу

улыбнулась удача и на его запросы откликнулся DHCP-сервер, то на экране это должно выглядеть примерно так:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:44:2B:02:4E

Searching for server (DHCP) ...

..Me: 192.168.1.6, Server: 192.168.1.1, Gateway 192.168.1.1

Loading 192.168.1.1:/lts/vmlinuz-2.4.24-ltsp-1 ... (NBI) .....

.....

Об успешности получения ответа от DHCP-сервера свидетельствует строка, которая начинается с “Me:”. Далее в ней располагаются адрес, присвоенный X-терминалу (192.168.1.6), адрес сервера (Server: 192.168.1.1) и шлюз по умолчанию (Gateway 192.168.1.1). Теперь смело можно переходить к разделу “Проблемы с TFTP”.

Если вы не увидели на экране строки с присвоенным X-терминалу сервером IP-адреса, то это означает, что налицо ошибки в функционировании DHCP-сервера. О неполадках в работе DHCP говорит наличие такого текста на экране X-терминала (сообщение “Searching for server (DHCP) ...” и появляющиеся точки с интервалом в несколько секунд):

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:44:2B:02:4E

Searching for server (DHCP) ...

.....

Найти причину подобных ошибок иногда бывает очень сложно, но с помощью простых рекомендаций можно значительно сократить количество возможных вариантов.

Перед началом диагностики необходимо проверить:

Подключен ли физически X-терминал к той же локальной сети, что и DHCP-сервер?

Горят ли при включении индикаторы сетевого адаптера X-терминала?

Каким кабелем подключен X-терминал к серверу? Если используется схема без сетевого коммутатора (концентратора), то следует использовать перекрестный кабель (cross-over), а не прямой (straight-thru), и наоборот;

В больших организациях часто можно встретить неординарные способы построения локальных сетей, когда из-за неправильного проектирования на практике встречаются схемы, где между удаленными участками локальной сети расположен целый каскад концентраторов. В таком случае к удаленному компьютеру могут попросту не доходить широковещательные запросы. Для предотвращения подобных проблем попробуйте подключить X-терминал самым простым способом к DHCP-серверу.

Если предпринятые меры не привели к положительному результату, значит проблема в самом DHCP-сервере. Для начала следует проверить запущен ли вообще данный демон на сервере X-терминалов. Выполняем команду:

```
$ netstat -an | grep ":67"
udp        0      0 0.0.0.0:67              0.0.0.0:*
```

Полученный ответ свидетельствует, о том что на сервере в фоновом режиме работает программа, которая ожидает датаграммы на UDP-порту с номером 67. Также можно утверждать, что датаграммы ожидаются на всех сетевых интерфейсах (0.0.0.0:*). В принципе такое поведение типично для DHCP-сервера, но чтобы точно удостовериться в этом, давайте выполним еще одну команду:

$ ps aux | grep dhcpd

root 28617 0.0 0.0 2304 512 ? S 2004 0:00 /usr/sbin/dhcpd

mikola 14987 0.0 0.0 3768 600 pts/0 S 16:21 0:00 grep dhcpd

В данном примере первая строка (/usr/sbin/dhcpd) означает, что DHCP-сервер работает. Удостовериться в этом можно также, выполнив интуитивно более понятную команду:

$ /sbin/service dhcpd status

dhcpd (pid 28617) running...

Если вы не увидели ничего похожего, то сейчас самое время вручную запустить на выполнение демон dhcpd. В операционной системе ASPLinux это можно сделать такой командой:

/sbin/service dhcpd start

Starting dhcpd: [OK]

В данном примере демон dhcpd запустился нормально, и сейчас самое время снова проверить загрузку удаленного X-терминала. Если при ручном запуске вы увидели на экране ошибку, то это означает, что в конфигурационном файле /etc/dhcpd.conf допущены неточности, исправить которые необходимо в первую очередь. Просмотреть сообщения об ошибках при запуске демона dhcpd возможно, если выполнить такую команду:

grep dhcpd /var/log/messages | tail

Feb 1 10:30:36 xtserver dhcpd: No subnet declaration for eth1 (192.168.0.26).

Feb 1 10:30:36 xtserver dhcpd: ** Ignoring requests on eth1. If this is not what

Feb 1 10:30:36 xtserver dhcpd: you want, please write a subnet declaration

Feb 1 10:30:36 xtserver dhcpd: in your dhcpd.conf file for the network segment

Feb 1 10:30:36 xtserver dhcpd: to which interface eth1 is attached. **

Feb 1 10:30:36 xtserver dhcpd:

Feb 1 10:30:36 xtserver dhcpd: Listening on LPF/eth0/00:80:48:25:a8:ca/WORKSTATIONS

Feb 1 10:30:36 xtserver dhcpd: Sending on LPF/eth0/00:80:48:25:a8:ca/WORKSTATIONS

Feb 1 10:30:36 xtserver dhcpd: Sending on Socket/fallback/fallback-net

Feb 1 10:30:36 xtserver dhcpd: запуск dhcpd succeeded

Пример работоспособной конфигурации DHCP-сервера и файла /etc/dhcpd.conf можно найти в разделе "Конфигурирование сетевых служб" главы "Настройка сервера", которая как раз и посвящена настройке сервера X-терминалов.

Если ручной запуск демона dhcpd помог устранить проблему с загрузкой X-терминала, то сейчас необходимо проверить настройку автоматического запуска демона dhcpd в уровне выполнения номер 5 (runlevel 5):

$ /sbin/chkconfig --list dhcpd

dhcpd 0:off 1:off 2:off 3:off 4:off 5:on 6:off

В данном примере с уровнем выполнения все в порядке. Если вы увидели другой результат, выполните от имени суперпользователя такую команду:

/sbin/chkconfig --level 5 dhcpd on

Или, если вы больше склоняетесь к утилитам с пользовательским интерфейсом, можете воспользоваться другой командой (см. рис. 8.1):

/usr/sbin/ntsysv

Рис. 8.1. Установка автозапуска демона dhcpd

Если демон dhcpd работает, но X-терминал так и не получил от него IP-адреса, следует еще раз проверить содержимое файла /etc/dhcpd.conf.

Возможно, вы ошиблись в написании физического MAC-адреса сетевого адаптера. Например, если при загрузке X-терминала на экране вы увидели такое сообщение:

```
Loading ROM image ......................
ROM segment 0x0000 length 0x0000 reloc 0x00020000
Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]
Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)
Boot from (N)etwork or (Q)uit?
Probing pci nic ...
[rtl8029]
NE2000 base 0xfcc0, addr 03:42:60:FF:AC:0D
```

В файле /etc/dhcpd.conf ему должна соответствовать запись, похожая на:

```
host ws003 {
   hardware ethernet     03:42:60:FF:AC:0D;
   fixed-address         192.168.1.3;
   filename              "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

Очень тщательно проверяйте соответствие MAC-адресов, так как ошибка даже в одном символе сделает невозможной загрузку X-терминала. После внесения изменений в файл /etc/dhcpd.conf не забудьте перезапустить демон dhcpd:

```
# /sbin/service dhcpd restart
```

После двойной проверки конфигурационного файла DHCP-сервера (/etc/dhcpd.conf) и при все той же неработоспособности X-терминала нужно удостовериться, что на X-терминал сервере не выполняется программное обеспечение, блокирующее DHCP-запросы. Такая ситуация часто возникает при некорректной настройке брандмауэра (firewall). В операционной системе Linux программный брандмауэр может быть настроен при помощи утилит: iptables, реже ipchains и еще реже ipfwadm.

Первое, что необходимо сделать – это проверить работают ли данные программы. Так как принцип работы у программных брандмауэров

одинаковый, то последующие примеры будут приводиться только для программы iptables. Итак, выполняем команду проверки запуска iptables:

```
# /sbin/service iptables status
Table: filter
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

В данном примере для всех цепочек применяется политика ACCEPT, что означает отсутствие правил фильтрации сетевых пакетов, а следовательно iptables не блокирует DHCP-запросы X-терминалов.

Так как программный брандмауэр iptables работает с сетевыми пакетами на очень низком уровне, то из соображений производительности он реализован в виде модуля ядра, а как результат - его присутствие в оперативной памяти сервера нельзя определить анализируя список выполняющихся процессов. Поэтому методика ps aux | grep ip_tables нам не подойдет. Нужно проверить какие модули ядра загружены и выполняются:

```
$ /sbin/lsmod | grep ip
iptable_filter         2412   0 (autoclean) (unused)
ip_tables             15776   1 [iptable_filter]
```

Говоря откровенно, разобраться в большом списке правил для iptables без специальной подготовки и опыта довольно сложно, а поэтому, чтобы исключить из числа возможных причин некорректно настроенный брандмауэр, для проверки я предлагаю временно отключить его:

```
# /sbin/service iptables stop
```

Можно даже для повышенной надежности принудительно выгрузить из оперативной памяти модули ядра, принадлежащие пакету iptables:

/sbin/rmmod iptable_filter

/sbin/rmmod ip_tables

Если после отключения брандмауэра загрузка X-терминала успешно прошла стадию получения IP-адреса от DHCP-сервера, то причина найдена – ваш брандмауэр iptables блокирует запросы DHCP и необходимо подкорректировать цепочки его правил.

При отрицательном результате от выключения брандмауэра у вас в арсенале есть еще один способ проверки работы протокола DHCP. Для этого при получении на экран X-терминала строки “Searching for server (DHCP) ...” следует проследить, какие записи появляются в системном журнале. Чтобы видеть вновь поступающие в него записи, достаточно выполнить такую команду от имени суперпользователя:

tail -f /var/log/messages

Например, если вы увидите там нечто подобное этому:

Feb 1 11:07:34 xtserver dhcpd: DHCPDISCOVER from 00:02:FD:72:FE:A0 via eth0

Feb 1 11:07:34 xtserver dhcpd: no free leases on subnet WORKSTATIONS

Feb 1 11:07:59 xtserver dhcpd: DHCPDISCOVER from 00:02:FD:72:FE:A0 via eth0

Feb 1 11:08:00 xtserver dhcpd: no free leases on subnet WORKSTATIONS

Это означает, что запросы от X-терминала с MAC-адресом сетевого адаптера 00:02:FD:72:FE:A0 поступают на сетевой интерфейс сервера с символическим именем eth0, и сервер-демон DHCP успешно их принимает, но, к сожалению, не располагает никакой информацией по поводу возможности присвоения IP-адреса X-терминалу с таким физическим MAC-адресом. Если вы столкнулись именно с такой ситуацией, то налицо ошибка в конфигурационном файле /etc/dhcpd.conf. Исправляете ошибку, перезапускаете демон dhcpd и проверяете загрузку X-терминала. Если загрузка X-терминала дошла до строки “Me:” на экране, и дальше остановилась, то это означает успешное минование протокола DHCP. Проблему теперь нужно искать в настройках сервера TFTP.

Проблемы с TFTP

Как вы уже знаете, сервер TFTP в X-терминальной сети предназначен для загрузки ядра X-терминала через локальную сеть. Если с доступом по протоколу TFTP возникли проблемы, то об этом может свидетельствовать такая строка на экране:

Loading ROM image .....

ROM segment 0x0000 length 0x0000 reloc 0x00020000

Etherboot 5.2.2. (GPL) http://etherboot.org Tagged ELF for [NE2000/PCI]

Relocation _text from [00013d70, 00022800) to [01ef1570, 01f00000)

Boot from (N)etwork or (Q)uit?

Probing pci nic ...

[rtl8029]

NE2000 base 0xfcc0, addr 00:02:44:2B:02:4E

Searching for server (DHCP) ...

..Me: 192.168.1.6, Server: 192.168.1.1, Gateway 192.168.1.1

Loading 192.168.1.1:/lts/vmlinuz-2.4.24-ltsp-1

При нормальной загрузке ядра операционной системы Linux через локальную сеть после имени файла (например, vmlinuz-2.4.24-ltsp-1) должны последовательно появляться символы точки “.” вплоть до появления строки “done”, которая свидетельствует, что загрузка ядра по протоколу TFTP успешно завершена. Если же символы точек не появляются, или прекратили появляться, а на экране X-терминала мигает текстовый курсор, то это указывает на возникшую проблему загрузки по протоколу TFTP.

Диагностику данной проблемы целесообразно начать с проверки того, работает ли демон tftpd. В операционной системе ASPLinux, как и в других системах, основанных на дистрибутиве RedHat, запуск демона tftpd осуществляется посредством использования возможностей суперсервера xinetd. Поэтому для начала проверим, а запущен ли xinetd. Сделать это можно командой:

$ ps aux | grep xinetd

root 707 0.0 0.0 2184 80 ? S 2004 0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid

mikola 22439 0.0 0.0 3768 600 pts/0 S 15:42 0:00 grep xinetd

Если вы получили похожий результат, то демон xinetd работает. В противном случае принудительно запускаем его:

```
# /sbin/service xinetd start
Starting xinetd:                                   [  OK  ]
```

После успешного запуска xinetd проверяем наличие файла, отвечающего за запуск демона tftpd:

```
$ cat /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        disable = no
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

Выше представлено его нормальное содержимое. Если файл /etc/xinetd.d/tftp присутствует и с его содержимым все нормально, а ядро X-терминала все равно не загружается, обратите внимание на строчку “server = /usr/sbin/in.tftpd”. По указанному в ней пути должен находиться исполняемый файл сервера TFTP. Еще одной интересной строкой файла запуска tftpd

является строка серверных аргументов: “server_args = -s /tftpboot”. Она означает, что для демона tftpd корневой директорией является каталог /tftpboot. Такое ограничение введено из-за соображений безопасности и для пользователей X-терминалов оно подразумевает, что при доступе по протоколу TFTP они могут получить только файлы, принадлежащие каталогу /tftpboot и его подкаталогам. На практике это означает, что если в файле /etc/dhcpd.conf записан путь к файлу с образом ядра операционной системы Linux для X-терминала, например так:

```
host yanax {
    hardware ethernet     00:02:44:2B:02:4E;
    fixed-address         192.168.1.6;
    filename              "/lts/vmlinuz-2.4.24-ltsp-1";
}
```

С параметром “-s /tftpboot” демон tftpd будет пытаться загрузить не файл /lts/vmlinuz-2.4.24-ltsp-1, а файл /tftpboot/lts/vmlinuz-2.4.24-ltsp-1. Неправильное расположение образов операционной системы для загрузки X-терминалов - довольно распространенная причина ошибок при настройки протокола TFTP.

В данном разделе я специально не упомянул о проблемах в работе локальной сети, так как если бы они имели место, то вы вряд ли прошли бы стадию получения ответа от DHCP-сервера. К тому же, эксплуатацию X-терминалов целесообразно выполнять на заранее проверенной и работоспособной локальной сети.

Устранение проблем с NFS

Файловая система NFS нужна X-терминалу для монтирования корневой файловой системы сервера, а также для обеспечения доступа к файлам подкачки через локальную сеть. При работе сервера NFS возникает довольно большое количество ошибок, связанных с неправильной настройкой данной сетевой файловой системы. Более того, иногда проблемы связаны не с самой NFS, а с ошибочной работой других сетевых служб. В подтверждение этих слов вы лично можете искусственно воссоздать неработающий X-терминал сервер, если неправильно настроите маршрутизацию, фильтрацию пакетов или

попросту присвоите противоречивые параметры сетевым интерфейсам. Поэтому в этом разделе содержатся советы по ликвидации проблем, которые непосредственно связаны с использованием сетевой файловой системы NFS.

Мне доводилось сталкиваться с ситуацией, когда при попытке запуска демонов NFS останавливалась загрузка сервера X-терминалов. При этом на экране отображалась такие строки:

```
...
Starting xinitd                                [  OK  ]
Starting NFS services
```

На этом загрузка сервера благополучно останавливалась и идти дальше отказывалась. Если вы столкнетесь с подобной проблемой, то в первую очередь проверьте значение файла /etc/resolv.conf, так как возможно причина именно в его содержимом. В моем случае, там был указан адрес сервера имен (DNS-сервера провайдера), доступ к которому был временно недоступен. Если это не поможет, то для выяснения причин проанализируйте вывод команды, которая показывает направление маршрутизации сервера X-терминалов, например:

```
$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
0.0.0.0         192.168.0.22    0.0.0.0         UG        0 0          0 eth1
```

Если на экране X-терминала при загрузке вы увидели сообщение типа:

```
Kernel panic: No init found.  Try passing init= option to kernel.
```

Это означает, что вы пытаетесь примонтировать неправильную корневую директорию, или что каталог /opt/ltsp/i386/ - пуст.

При запуске X-терминала на экране появилось сообщение такого содержания:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

Его смысл означает, что директория /opt/ltsp/i386 не описана в файле параметров экспортирования /etc/exports. Если этот так, то сейчас самое время ее туда вписать. Об ошибках такого рода могут свидетельствовать сообщения в системном журнале, похожие на:

Feb 1 08:15:49 xtserver rpc.mountd: refused mount request from ws001 for /opt/ltsp/i386 (/): no export entry

Сетевая файловая система NFS использует для своих нужд три вспомогательные программы: nfsd, mountd и portmap (иногда еще и rquotad). Поэтому при возникновении исключительных ситуаций, в первую очередь следует проверить наличие их в памяти сервера X-терминалов. Воспользуемся для этих целей такими командами:

```
$ ps aux | grep nfsd
root      7556  0.0  0.0     0    0 ?        SW    2004   0:27 [nfsd]
root      7557  0.0  0.0     0    0 ?        SW    2004   0:24 [nfsd]
root      7558  0.0  0.0     0    0 ?        SW    2004   0:24 [nfsd]
root      7559  0.0  0.0     0    0 ?        SW    2004   0:26 [nfsd]
root      7560  0.0  0.0     0    0 ?        SW    2004   0:28 [nfsd]
root      7561  0.0  0.0     0    0 ?        SW    2004   0:25 [nfsd]
root      7562  0.0  0.0     0    0 ?        SW    2004   0:23 [nfsd]
root      7563  0.0  0.0     0    0 ?        SW    2004   0:24 [nfsd]
mikola   28403  0.0  0.0  3764  596 pts/0    S    16:09   0:00 grep nfsd
```

Если результат работы команды похож на вывод, что представлен выше, то демон nfsd запущен.

```
$ ps aux | grep mountd
root      7571  0.0  0.0  1768  316 ?        S     2004   0:00 rpc.mountd
mikola   28414  0.0  0.0  3768  600 pts/0    S    16:11   0:00 grep mountd
```

С запуском mountd также все в порядке. Проверить работу демонов NFS-сервера можно и таким способом:

```
$ /sbin/service nfs status
rpc.mountd (pid 7571) running...
nfsd (pid 7563 7562 7561 7560 7559 7558 7557 7556) running...
```

rpc.rquotad (pid 7552) running...

Для portmap применяем стандартную команду:

```
$ ps aux | grep portmap
rpc       640  0.0  0.0  1556  164 ?        S    2004   0:00 portmap
mikola   28445  0.0  0.0  3764  596 pts/0    S    16:12   0:00 grep portmap
```

Как видно по ее ответу, программа portmap также работает. Проверить работу portmap можно еще и следующим образом:

```
$ netstat -an | grep ":111 "
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:111            0.0.0.0:*
```

Если вы обнаружите, что один из демонов NFS не запущен, то его следует запустить одной из следующих команд:

```
# /sbin/service portmap start
Starting portmapper:                          [  OK  ]
# /sbin/service nfs start
Starting NFS services:                        [  OK  ]
Starting NFS quotas:                          [  OK  ]
Starting NFS daemon:                          [  OK  ]
Starting NFS mountd:                          [  OK  ]
```

На отсутствие в оперативной памяти сервера процесса portmap может в частности свидетельствовать такие строки на экране X-терминала:

```
Looking up port of RPC 100003/2 on 192.168.1.1
portmap: server 192.168.1.1 not responding, timed out
Root-NFS: Unable to get nfsd port number from server, using default
Looking up port of RPC 100005/2 on 192.168.1.1
portmap: server 192.168.1.1 not responding, timed out
Root-NFS: Unable to get mountd port number from server, using default
mount: server 192.168.1.1 not responding, timed out
Root-NFS: Server returned error -5 while mounting /opt/ltsp/i386
VFS: unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device "nfs" or 02:00
```

Please append a correct "root=" boot option

Kernel panic: VFS: Unable to mount root fs on 02:00

Бывают ситуации, когда все необходимые сетевой файловой системе NFS демоны запущены, а X-терминал все равно не запускается. В таком случае нужно проверить зарегистрированы ли демоны NFS в службе portmap. Поможет нам в этом такая команда:

```
$ /usr/sbin/rpcinfo -p localhost
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  32768  status
    100024    1   tcp  32768  status
    391002    2   tcp  37614  sgi_fam
    100011    1   udp    807  rquotad
    100011    2   udp    807  rquotad
    100011    1   tcp    810  rquotad
    100011    2   tcp    810  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100021    1   udp  39135  nlockmgr
    100021    3   udp  39135  nlockmgr
    100021    4   udp  39135  nlockmgr
    100005    1   udp  39136  mountd
    100005    1   tcp  44446  mountd
    100005    2   udp  39136  mountd
    100005    2   tcp  44446  mountd
    100005    3   udp  39136  mountd
    100005    3   tcp  44446  mountd
```

Как и было обещано, проблем с использованием NFS-сервера хватает. Но следует отметить, что один раз правильно настроенная сетевая файловая система в дальнейшем работает, как правило, без сбоев.

Проблемы X-сервера

Проблемы, которые возникают при запуске X-сервера X-терминалов, часто являются следствием неправильной настройки других системных служб сервера. Иногда бывает очень сложно найти причину неработоспособности графической части X-терминала, особенно если в данной области у администратора мало практического опыта. Помню, когда я впервые столкнулся с необходимостью правки файла конфигурации графической системы XFree86, то мною по-неволе овладел ужас написать что-то не так, и окончательно своими некомпетентными действиями зарубить операционную систему. Но со временем, до меня начала доходить идеология UNIX-систем, и, признаться честно, сейчас она мне намного более по душе, чем, например, подход предлагаемый Microsoft. Если знать где в операционной системе брать информацию, то настроить X-сервер не сложнее других подсистем Linux.

Для того, чтобы сократить к минимуму ваши усилия по настройке графической части X-терминала, неплохо было бы изначально удостовериться, что используемая вами видеокарта поддерживается проектом XFree86. В противном случае все ваши усилия будут напрасными, за исключением той невероятной ситуации, когда вы сами в состоянии написать драйвер для своего видеоадаптера.

Итак, начнем ликвидацию проблем X-сервера с наиболее распространенных на практике случаев.

О проблеме запуска графической подсистемы X-терминала свидетельствует мигание монитора, когда текстовый экран входа в систему сменяется черным, и так в течении нескольких минут, вплоть до появления сообщения об ошибке. Инициализирующий скрипт X-терминала в проекте LTSP будет пытаться перезапустить неработающий X-сервер 10 раз, после чего прекратит это бесполезное занятие. Чтобы не ждать так долго, при устранении проблем запуска X-сервера рекомендуется перевести X-терминал в текстовый режим. Сделать это очень просто, достаточно в параметрах файла /opt/ltsp/i386/etc/lts.conf, которые соответствуют разделу нужного X-терминала,

записать текстовый режим выполнения (runlevel = 3). Например, для X-терминала с именем pashax там должны быть такие строки:

[pashax]

RUNLEVEL = 3

В текстовом режиме, после получения приглашения от командной оболочки для запуска X-сервера используется команда:

/tmp/start_ws

Если запуск X-сервера из текстового режима завершится неудачей, то об этом будут свидетельствовать сообщения на экране и управление вновь перейдет к командной строке.

Просмотреть все сообщения X-сервера на экране можно, если взглянуть на содержимое файла /var/log/XFree86.0.log. Так как содержимое этого файла довольно большое и не всегда содержит информативные данные, можно ознакомиться только с сообщениями об ошибках и предупреждениями:

egrep "^\(EE|^\(WW" /var/log/XFree86.0.log

(WW) System lacks support for changing MTRRs

(EE) MGA(0): Static buffer allocation failed, not initializing the DRI

(EE) MGA(0): Need at least 5184 kB video memory at this resolution, bit depth

Все вышесказанное относится к версии X-сервера 4.x, а если для вашей видеокарты используется сервер XFree68 версии 3.3.6, то для получения полного отчета об ошибках его запуска необходимо применить немного модифицированную команду:

/tmp/start_ws 2>/tmp/Xlog.txt

Теперь после запуска X-сервера в файле /tmp/Xlog.txt находятся все сообщения об ошибках, которые имели место. Просмотреть данный файл можно, использовав команду more:

more /tmp/Xlog.txt

Ниже представлено типичное содержание файла Xlog.txt после неудачного запуска X-сервера версии 3.3.6:

XFree86 Version 3.3.6 / X Window System

(protocol Version 11, revision 0, vendor release 6300)

```
Release Date: January 8 1999
  If the server is older than 6-12 months, or if your card is newer
  than the above date, look for a newer version before reporting
  problems.  (see http://www.XFree86.Org/FAQ)
Operating System: Linux 2.2.14 i686 [ELF]
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      newmmio, mmio_928, s3_generic
(using VT number 3)

XF86Config: /etc/XF86Config
(**) stands for supplied, (--) stands for probed/default values
(**) XKB: keycodes: "xfree86"
...
(**) S3: Graphics device ID: "My Video Card"
(**) S3: Monitor ID: "My Monitor"
(--) S3: Mode "640x480" needs vert refresh rate of 100.02 Hz. Deleted.
...
(--) S3: Mode "1800X1440" needs hsync freq of 104.52 kHz. Deleted.
(**) FontPath set to "tcp/192.168.1.1:7100"

 *** None of the configured devices were detected.***



Fatal server error:
no screens found

When reporting a problem related to a server crash, please send
the full server output, not just the last messages
```

Проблемы X-сервера (продолжение)

Столкнувшись с проблемой запуска X-сервера на X-терминале и после изучения содержимого лог-файлов работы XFree86, в первую очередь необходимо изучить файл /etc/XF86Config. В большинстве случаев, этот конфигурационный файл X-сервера был автоматически построен скриптом /opt/ltsp/i386/etc/rc.setupx для версии XFree86 4.x или /opt/ltsp/i386/etc/rc.setupx3 для версии XFree86 3.3.6 соответственно. Вполне допустимо, что данные скрипты не учли всех особенностей вашей аппаратной конфигурации X-терминала, и им в этом необходимо помочь.

Изучая содержимое файла /etc/XF86Config, обратите внимание на тип видеодрайвера (для X-сервера версии 4.x), настройки монитора, глубину цвета и разрешение экрана по умолчанию. Иногда следует для начала выбрать настройки по минимуму (например, разрешение 640x480, глубину цвета 8 бит, а частоту вертикальной разверстки 60 Гц), а уже потом постепенно увеличивать значение этих параметров для достижения оптимальной производительности.

Об успешном запуске видеоадаптера может свидетельствовать переход экрана в графический режим и появление на нем курсора “мышки” в виде буквы “X”. Даже если после этого вы больше ничего не увидите, это говорит о том, что X-сервер успешно загрузился. Дальнейшие указания по ликвидации проблем с X-терминалом нужно искать в разделе “Проблемы менеджера дисплея”.

Следует также упомянуть об особенностях использования устаревших X-серверов версии 3.3.6. Как более старая разработка, они не поддерживают часть современных возможностей XFree86 версий 4.x. В первую очередь это относится к таким элементам как сглаженность экранных шрифтов и потенциальные проблемы безопасности. С настройками видеокарт там также не все гладко, на практике мне довелось работать с видеоадаптерами, которые не хотели работать на большей чем 72 Гц частоте вертикальной разверстки, хотя, установленные на компьютерах с операционной системе Windows, они прекрасно давали все 85 Гц. Усугубляет данную ситуацию также то, что работая на X-терминале с X-сервером версии 3.3.6 вы можете столкнуться со значительными проблемами при работе с приложениями среды KDE.

Свидетельствовать о потенциальных недостатках могут появляющиеся сообщения в файле ~/.xsession-errors, такого содержания:

Xlib: extension "RENDER" missing on display "oxanax.xterm.net:0.0".

Если вы увидите на экране нечто подобное, то это означает, что вы используете KDE-приложение (точнее, следует сказать, приложение использующее графическую библиотеку Qt), которое было собрано с поддержкой современного расширения Render, но ваша видеокарта или X-сервер его, к сожалению, не поддерживают. Для конечного пользователя это означает, что приложения работающие с библиотекой Qt работать будут значительно медленнее на таком X-терминале, а в ряде случаев и вовсе неудовлетворительно.

Если вы столкнетесь с неразрешимыми проблемами при попытке настроить X-сервер конкретного X-терминала, прежде чем прекратить ваши усилия и оставить это утомительное занятие, попробуйте заменить видеоадаптер на модель, которая поддерживается проектом XFree86 версии 4.x. Также рекомендую использовать параметр файла /opt/ltsp/i386/etc/lts.conf "XSERVER = vesa" для того, чтобы попробовать работать в VESA-совместимом режиме.

Проблемы менеджера дисплея

Когда X-сервер на X-терминале успешно запустился, он пытается соединиться с менеджером дисплея сервера X-терминалов для получения графического приглашения ввода имени пользователя и пароля. Если с настройками менеджера дисплея что-то не так, то пользователь X-терминала будет лицезреть на сером фоне экрана большой X-курсор “мышки”. Это наверное самая распространенная проблема, которая встречается при настройке удаленного подключения к менеджеру дисплея, используя протокол XDMCP.

Если вам "посчастливилось" увидеть этот унылый серый фон рабочего стола и одинокий курсор в виде буквы “X”, и вы смотрите на него уже более 1 минуты, значит проблема подключения к менеджеру дисплея сервера налицо. Для выявления причин такого поведения X-терминала, проверим запущена ли

программа менеджера дисплея. Посмотрите на результат выполнения этой команды:

```
# netstat -ap | grep xdm
udp        0      0 *:xdmcp                *:*                          912/gdm-binary
```

Как видно, с протоколом XDMCP действительно связан процесс с идентификатором 912 и именем программного файла gdm-binary. В противном случае сервер просто не обрабатывает запросы по протоколу XDMCP или менеджер дисплея не запущен.

Если программа менеджера дисплея работает, а подключение к ней X-терминалов все равно не происходит, то причину ошибки следует искать в ее настройках. Так, в некоторых случаях, менеджер дисплея может попросту игнорировать запросы от удаленных компьютеров.

В операционной системе Linux используются в основном 3 менеджера дисплея: XDM, GDM и KDM. XDM – это старейший представитель подобных программ, включается практически во все дистрибутивы X-Windows. GDM и KDM - это соответственно менеджеры дисплеев таких оконных менеджеров как Gnome и KDE. Они отличаются от XDM более широкими возможностями по настройке экрана графического входа в систему. В зависимости от того, каким менеджером дисплея пользуетесь вы, вам следует проверить его файлы конфигурации.

XDM

В файле /etc/X11/xdm/xdm-config не должно быть такой строки:

```
DisplayManager.requestPort:     0
```

Иначе удаленные XDMCP-запросы на 177 порту будут попросту игнорироваться.

В файле /etc/X11/xdm/Xaccess должна присутствовать такая строка:

```
*          #any host can get a login window
```

Обратите особое внимание, что данная строчка не должна быть закомментирована символом “#”.

KDM

Проверьте содержимое файла /etc/X11/xdm/kdmrc (в некоторых дистрибутивах /etc/kde/kdm/kdmrc). Вас в частности должны интересовать такие строки раздела [Xdmcp]:

[Xdmcp]

Whether KDM should listen to XDMCP requests. Default is true.

Enable=true

The UDP port KDM should listen on for XDMCP requests. Don't change the 177.

#Port=177

Если опция XDMCP “Enable=true” и порт для XDMCP запросов 177, то менеджер дисплеев KDM должен отвечать удаленным пользователям X-терминалов.

GDM

Для менеджера дисплеев GDM находим в файле /etc/X11/gdm/gdm.conf раздел [xdmcp]:

[xdmcp]

Distributions: Ship with this off. It is never a safe thing to leave

out on the net. Alternatively you can set up /etc/hosts.allow and

/etc/hosts.deny to only allow say local access.

Enable=true

Honour indirect queries, we run a chooser for these, and then redirect

the user to the chosen host. Otherwise we just log the user in locally.

HonorIndirect=true

Maximum pending requests

MaxPending=4

MaxPendingIndirect=4

Maximum open XDMCP sessions at any point in time

MaxSessions=16

Maximum wait times

MaxWait=15

MaxWaitIndirect=15

```
# How many times can a person log in from a single host.  Usually better to
# keep at 1 to fend off DoS attacks by running many logins from a single
# host
DisplaysPerHost=2
# The port.  177 is the standard port so better keep it that way
Port=177
# Willing script, none is shipped and by default we'll send
# hostname system id.  But if you supply something here, the
# output of this script will be sent as status of this host so that
# the chooser can display it.  You could for example send load,
# or mail details for some user, or some such.
Willing=/etc/X11/gdm/Xwilling
```

В первую очередь проверяем, разрешены ли вообще XDMCP-запросы (опция “Enable=true”, в некоторых старых версиях GDM “Enable=1”). Потом, смотрим на каком порту ожидаются эти запросы (должен быть “Port=177”). Если вы наблюдаете ситуацию, когда X-терминал иногда подключается к удаленному менеджеру дисплеев GDM, а иногда нет, то проверьте также параметры “MaxSessions” и “DisplaysPerHost”. Значение MaxSessions означает, сколько всего удаленных пользователей может подключиться к менеджеру дисплеев GDM, а значение DisplaysPerHost позволяет ограничивать число параллельных запросов на подключение с одного узла (IP-адреса) локальной сети. Хотя из соображений безопасности (например, для борьбы с DoS атаками) целесообразно устанавливать DisplaysPerHost=1, иногда при аварийном выключении X-терминала, он не может повторно подсоединиться к менеджеру дисплея. Поэтому, чтобы лишний раз не отвечать на вопросы пользователей, почему их X-терминал не включается, я устанавливаю DisplaysPerHost равным 2.

Иногда проблема подключения удаленного пользователя к менеджеру дисплея связана с неправильными настройками DNS или тем, что вы просто забыли описать X-терминал в файле /etc/hosts. При похожих проблемах обязательно проверьте правильно ли сопоставляется доменное имя X-

терминала и его IP-адрес. Например, для X-терминала, с IP-адресом 192.168.1.23 и доменным именем natax это можно протестировать при помощи команды ping:

```
$ ping -c 3 192.168.1.23
PING 192.168.1.23 (192.168.1.23) 56(84) bytes of data.
64 bytes from 192.168.1.23: icmp_seq=1 ttl=64 time=0.132 ms
64 bytes from 192.168.1.23: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 192.168.1.23: icmp_seq=3 ttl=64 time=0.119 ms

--- 192.168.1.23 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.113/0.121/0.132/0.012 ms

$ ping -c 3 natax
PING natax.alker.net (192.168.1.23) 56(84) bytes of data.
64 bytes from natax.alker.net (192.168.1.23): icmp_seq=1 ttl=64 time=0.118 ms
64 bytes from natax.alker.net (192.168.1.23): icmp_seq=2 ttl=64 time=0.114 ms
64 bytes from natax.alker.net (192.168.1.23): icmp_seq=3 ttl=64 time=0.489 ms

--- natax.alker.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.114/0.240/0.489/0.176 ms
```

Ошибки при работе в графическом режиме

После того, как X-терминал успешно загрузился и на его экране показалось приглашение для ввода имени пользователя и пароля от удаленного менеджера дисплея, можно считать, что настройка его закончена. Работая в графическом режиме, пользователь практически не будет замечать разницы между использованием X-терминала и традиционного персонального компьютера с операционной системой Linux. Но если более пристально взглянуть на особенности работы на X-терминале, то стают заметными принципиальные

отличия, главное из которых – это работа действительно в многозадачной среде.

Наверное самая распространенная ошибка, которая происходит при работе пользователя на X-терминале в графическом режиме – это внезапное завершение работы X-сервера и сброс его к экрану ввода имени пользователя и пароля. Главной причиной, которая вызывает такое поведение X-терминала, является нехватка виртуальной памяти для работы X-сервера. Особенно часто такую картину можно наблюдать на терминалах с маленьким объемом оперативной памяти (16 – 32 Мбайт) и не использующих файл подкачки через локальную сеть. Самым простым решением данной проблемы является разрешение использования файла подкачки через локальную сеть для X-терминала или увеличение размера swap-файла. Для того, чтобы это сделать, нужно в разделе файла /opt/ltsp/i386/etc/lts.conf указать такие параметры:

USE_NFS_SWAP = Y

SWAPFILE_SIZE = 32m

Это приведет к созданию файла подкачки для выбранного X-терминала размером в 32 Мбайта при следующем его включении. Для создания файла подкачки требуется определенное время, поэтому при первой загрузке X-терминала будет замечена небольшая пауза. Если при установленном размере swap-файла пользователь все еще жалуется на периодические сбросы X-сервера, то следует в файле /opt/ltsp/i386/etc/lts.conf увеличить размер файла подкачки и перегрузить X-терминал.

На втором месте среди проблем при работе в графическом режиме, я бы назвал появление артефактов на экране или, выражаясь более понятным языком - искажений в отображении графической информации на экране. Иногда это вызвано некорректной работой или настройкой драйвера видеокарты, а иногда ошибками в работе программного обеспечения и X Window System. Бороться с неправильно настроенными драйверами необходимо, поэтому, если вы столкнулись с подобными проблемами, попробуйте отключить аппаратную акселерацию вашего X-сервера, сделать это можно при помощи такой опции файла /opt/ltsp/i386/etc/lts.conf:

X_DEVICE_OPTION_01 = "NoAccel"

Если после ее применения артефакты исчезли, но скорость отображения информация стала чересчур низкой, попробуйте поработать с драйвером vesa. Для этого в параметрах файла /opt/ltsp/i386/etc/lts.conf укажите такую строку:

XSERVER = vesa

Конечно не забудьте удалить строчку отключения акселерации (X_DEVICE_OPTION_01 = "NoAccel").

С ошибками в работе прикладного программного обеспечения бороться довольно трудно. Так, при работе за X-терминалом было замечено частичное искажение содержимого программных окон при вертикальной прокрутке документа, а также после закрытия модальных диалоговых окон. Конечно, ошибки в отображении данных на экране проявляются довольно редко, но если пользователь столкнется с ними, то наилучшим советом в такой ситуации будет обновление экрана оконного менеджера. Или, что еще проще, сворачивание окна в пиктограмму и восстановление его исходного положения, для срабатывания эффекта перерисовки содержимого окна.

Исходя из дистанционной природы обработки данных, которые посылают пользовательские устройства ввода ("мышь" и клавиатура), ответные результаты сервера приходят с небольшим опозданием по сравнению с работой на традиционном персональном компьютере. Особенно это заметно при работе по низкоскоростной локальной сети (до 10 Мбит/с), а также в режиме отключенной акселерации видеокарты X-терминала или при слабом и загруженном X-терминал сервере. В такой ситуации пользователи жалуются на сложность выделения больших фрагментов текста или групп ячеек электронной таблицы. Также неудовлетворительной остается и прокрутка документов при помощи скроллеров. Для таких пользователей можно посоветовать больше пользоваться функциональными клавишами клавиатуры. Например, текст выделять при помощи нажатой клавиши Shift и клавиш со стрелками, а прокрутку осуществлять нажатием на клавиши PageUp и PageDown.

Если после всех ваших действий избавиться от артефактов не удалось, то пора подумать о использовании другого видеоадаптера. Был в моей практике

случай с видеокартой Matrox G100 AGP 4Mb. Так вот, это устройство иногда при переходе в графический режим показывало на правой половине экрана пеструю хаотическую картинку, после появления которой X-терминал окончательно зависал. С заменой дефектного видеоадаптера появление подобных явлений прекратилось.

Иногда (за год эксплуатации X-терминала только два раза), мне доводилось наблюдать ничем не обоснованное замедление работы программ на X-терминале. При этом ни локальная сеть, ни центральный процессор сервера не были загружены. Такое странное поведение было у офисного пакета OpenOffice и выражалась в очень замедленной реакции на действия пользователей. Например, щелчок "мышкой" в текстовом редакторе приводил к переводу курсора в новое место только спустя 2 секунды. Также очень замедлялась прокрутка документов и переход между окнами офисного приложения и их прорисовка на экране. Закрытие OpenOffice ни к чему не приводило, так как после повторного запуска он также работал с задержками. Экспериментальным путем было установлено, что только завершение сеанса пользователя, и повторный вход в систему успешно ликвидировали эти неудобства. Интересно, что перезагрузка X-терминала не требовалась. Скорее всего это внутренняя ошибка офисного приложения или одной из используемых им разделяемых библиотек. Самое интересное, что на работе других программ это никак не сказывалось.

Подготовка пользователей для работы на X-терминале

Перед тем как предложить пользователю поработать на X-терминале, необходимо объяснить ему особенности использования этой техники. Так как ваши конечные пользователи уже имеют опыт работы с персональными компьютерами, то особых проблем в понимании быть не должно. В вводном курсе подготовки целесообразно сделать акцент на таких пунктах:

Порядок включения и выключения X-терминала

X-терминал включается подобно обычному персональному компьютеру нажатием кнопки "Power" на системном блоке. На экране пробегают строки загрузки операционной системы и через несколько секунд пользователь увидит

приглашение вводи имени пользователя и пароля (рис. 8.2). На этом загрузку X-терминала считаем успешной.

Рис. 8.2. Экран ввода имени пользователя и пароля

Для выключения X-терминала в конце рабочего дня следует закрыть все используемые приложения на всех виртуальных рабочих столах и выйти из своего сеанса соответствующей командой оконного менеджера. Появление на экране изображения рис. 8.2 означает готовность X-терминала к выключению. Для того чтобы выключить X-терминал, достаточно нажать кнопку "Power" на системном блоке.

Вход в систему и завершение сеанса работы

На приглашение системы ввести имя и пароль, пользователь X-терминала должен набрать на клавиатуре имя пользователя, которое ему присвоил администратор, нажать клавишу Enter. Затем пользователь вводит свой пароль и завершает ввод также нажатием клавиши Enter. Если имя пользователя и пароль введены правильно, то на экране X-терминала покажется рабочий стол оконного менеджера. При ошибочном вводе имени пользователя или пароля система откажет в авторизации и попросит повторить ввод.

Для минимизации ошибок при работе с диалогом ввода имени пользователя и пароля следует убедиться, что не нажата клавиша "Caps Lock", а также, что используется английская раскладка клавиатуры (индикатор "Scroll Lock" не горит). Имя пользователя всегда вводят в нижнем регистре. Пароль не должен быть короче 6-ти символов. В целях безопасности желательно иметь длинный пароль, который не является словом английского языка, а также содержит цифры и символы в разных регистрах.

Когда пользователь закончил свою работу он должен завершить свой сеанс, то есть выйти из него. Для этого необходимо сохранить измененные файлы и закрыть работающие прикладные программы. Далее следует нажать кнопку меню IceWM и выбрать пункт Выход. В диалоговом окне подтверждения необходимо нажать кнопку ОК (пример завершения сеанса приведен для оконного менеджера IceWM). Об успешном выходе из своего сеанса свидетельствует появление на экране изображения, представленного на рис. 8.2.

Проверка наличие сетевого кабеля

Работа X-терминала зависит от постоянного подключения к локальной сети. Поэтому пользователи X-терминалов должны заботиться о том, чтобы сетевой кабель всегда находился в соответствующем разъеме сетевой карты (см. рис. 8.3). Про успешность подключения к локальной сети, как правило, свидетельствуют мигающие индикаторы сетевого адаптера.

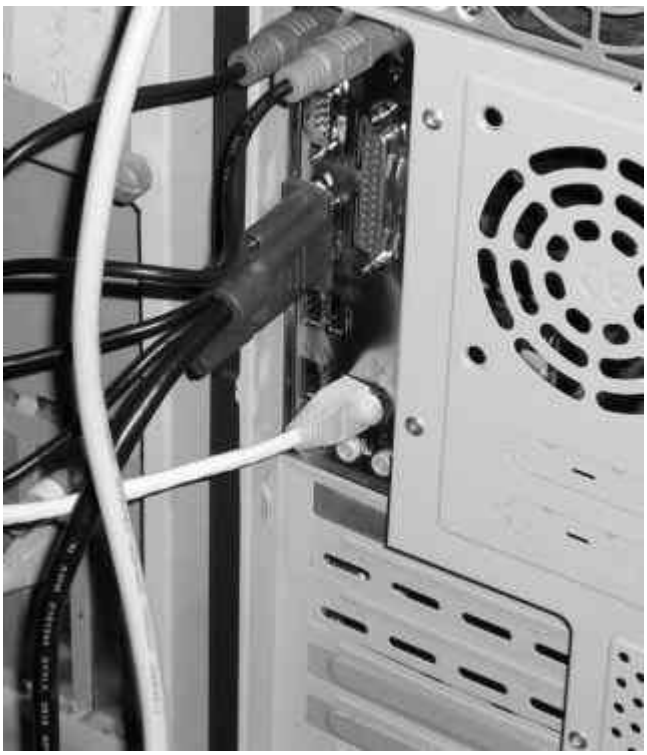

Рис. 8.3. Подключение X-терминала к локальной сети

В случае использования беспроводного подключения к локальной сети сетевой кабель будет отсутствовать. Но использование таких X-терминалов достаточно большая редкость.

Узнать о том, что X-терминал потерял связь с локальной сетью, можно по замиранию изображения на экране, а также в том случае, если он не включился.

Практическая бесполезность перезагрузки

Иногда при возникновении проблем с программным обеспечением, пользователи привыкли выполнять перезагрузку персональных компьютеров. Для X-терминала эта операция практически абсолютно бесполезна, так как неправильно работающие программы будут продолжать выполняться в оперативной памяти сервера. Вместо выполнения перезагрузки X-терминала, при появлении подобных проблем рекомендуется выполнить сброс X-сервера, нажатием комбинации клавиш Ctrl+Alt+BackSpace. Затем заново ввести имя пользователя и пароль и продолжать работу. Перед сбросом X-сервера необходимо сохранить все открытые документы.

Общая рабочая среда

Каждому пользователю X-терминала необходимо объяснить, что работая на своем компьютере, он является участником коллективной среды. Поэтому, кроме его приложений, в оперативной памяти сервера выполняются задачи других пользователей. В зависимости от нагрузки на центральный сервер, выполнение операций происходит быстрее или медленнее. Такой колебательный характер работы обуславливает переменную, изменчивую производительность X-терминалов. Страшного в этом ничего нет. Например, если пользователь открывал свой файл на 5 секунд дольше чем обычно, то это не повод для паники, а лишь говорит о том, что кто-то параллельно также работал с жестким диском. Возникновение пиков загрузки в течении более 30 секунд достаточно редкое явлении в сетях, насчитывающих до 20 X-терминалов и обслуживающихся одним сервером. По индикатору загрузки центрального процессора можно всегда узнать о наличии свободных вычислительных ресурсов сервера.

Универсальность рабочих мест

Все X-терминалы представляют собой единообразные рабочие места. Вне зависимости от того, за каким X-терминалом предпочтет работать пользователь, после вводи своего имени и пароля, он получит доступ к своему рабочему столу и личным документам. Такая универсальность позволяет пользователям не быть привязанным к компьютерам на их рабочем месте.

Особенно это интересно для организаций, у которых пользователей компьютеров больше, чем самих X-терминалов.

Не смотря на универсальный характер X-терминалов, следует удержаться от одновременной регистрации одного пользователя на разных компьютерах. Некоторые программы (например, OpenOffice, Galeon и т.д.) не будут работать в таком режиме. Правильно сначала завершить свой сеанс на одном X-терминале, а уже после этого вводить имя пользователя и пароль на другом. Во избежание несанкционированного доступа к вашим личным файлам возьмите себе за правило, выходить из сеанса каждый раз, когда завершаете работу на X-терминале. Это утверждение особенно справедливо в отношении многопользовательских рабочих мест. В случае монопольной работы за компьютером, во время продолжительного отсутствия пользователя на рабочем месте, следует активировать функцию блокирования рабочей станции (например, командой xlock).

Особенности работы с устройствами хранения данных и передача файлов

X-терминалы специфическим образом обращаются к устройствам хранения данных со сменными носителями такими как CD-ROM или флоппи-дисковод. Устройствами чтения компак-дисков они, как правило, не оборудованы, а для работы с дискетами используется специальная программа MToolsFM. Во избежание проблем с доступом к информации записанной на дискету, рекомендуется называть файлы и папки символами латинского алфавита. Попытки монтирования каталогов /mnt/cdrom и /mnt/floppy к положительному результату не приведут.

Для передачи файлов между X-терминалами следует использовать общие папки, которые специально выделены для этого системным администратором. Передачу информации между удаленными партнерами целесообразно проводить через услуги электронной почты, клиентами доставки мгновенных сообщений, а также посредством web и ftp-серверов.

Если документ передается человеку за пределы организации, в которой работает сеть X-терминалов, то следует позаботиться, чтобы данные были записаны в доступном ему формате. В связи с недостаточной

распространенностью операционной системы Linux и совместимых с ней программных продуктов, данные перед передачей необходимо экспортировать в один из следующих форматов:

Текстовые документы: формат Microsoft Word 97 (*.doc), Rich Text Format (*.rtf), Adobe Acrobat (*.pdf), простой текст в кодировке windows-1251, cp-1251 (*.txt);

Электронные таблицы: формат Microsoft Excel 97 (*.xls), Adobe Acrobat (*.pdf);

Гипертекстовые документы (*.html, *.htm, *.xml) экспортировать не нужно.

При переносе данных на накопителе USB Flash, можно специальным USB-удлинителем присоединить его к X-терминал серверу и сделать его монтирование доступным пользователям (опция user в файле /etc/fstab).

Новая операционная система Linux и ее прикладные программы

Если ваши пользователи до этого имели дело только с операционными системами компании Microsoft, то одновременно с освоением работы в X-терминальном режиме, им придется осваивать еще и новую для себя операционную систему, а также совершенно не знакомые прикладные программы. Построить этот рассказ целесообразно с перечисления ключевых отличий этих программных продуктов, а также сделать акцент на большом количестве сходных черт. Для того, чтобы сделать этот процесс менее болезненным, рекомендуется переводить одновременно не всех пользователей, а разбить работу на несколько этапов. Тогда в помощь системному администратору придет еще и коллективное сознание, когда более продвинутые пользователи будут делиться знаниями со своими коллегами "чайниками".

Когда я внедрял систему X-терминалов на практике, мне повезло, так как пользователи уже имели опыт работы в операционной системе Linux. А так как производительность X-терминалов была значительно лучше, чем у их предыдущих персональных компьютеров с операционной системой Linux, то идея X-терминальной сети была воспринята почти на ура.

Анализируя собственный опыт перевода пользователей с Windows на Linux, хотелось бы заметить, что у большинства неудовольство связано не с

самой операционной системой Linux, а офисным пакетом OpenOffice, который предлагается как альтернатива Microsoft Office. Более того, от пользователей часто слышатся фразы типа: "Некогда мне разбираться в этом Линуксе, мне работать надо!" или "Я хочу, чтобы все было как раньше!". В этих возгласах отчетливо видно нежелание принятия нового, а также естественная человеческому роду лень. Поэтому, для успешного внедрения системы X-терминалов необходима административная поддержка руководства компании, иначе, это нововведение может быть отброшено еще на первой стадии своего развития.

Принципиальных отличий в работе операционной системы Linux практически нет. Следует сделать акцент на именах файлов, которые чувствительны к регистру, также упомянуть о структуре файловой системы, и плавно перейти к описанию главных аспектов работы в оконном менеджере. Затем по отдельности описать все программные продукты, которые пользователи будут использовать чаще всего, а их вовсе не много: офисный пакет, файловый менеджер, архиватор, почтовый клиент, браузер и возможно программа отправки мгновенных сообщений. О наличии командной оболочки (shell) следует знать только грамотным пользователям, так как в руках непосвященного человека от нее больше вреда чем пользы (недаром говорят, что дураку грамота вредна).

Инструкция для пользователя

Желательно, чтобы все пожелания для пользователей X-терминалов были оформлены в виде инструкции. На дело повышения компьютерной квалификации сотрудников вашей организации не жалко потратить 20 страниц бумаги и несколько часов времени системного администратора. К тому же, некоторые люди лучше усваивают информацию, если она написана на бумаге, чем воспринимают ее на слух.

Когда составление инструкции будет закончено, очень важно дать ее почитать самому не подготовленному пользователю. По его замечаниям вы сможете узнать, насколько вы переусердствовали в употреблении специфических, для компьютерной отрасли, терминов и сленга.

Наличие хорошо составленной и понятной инструкции в будущем сэкономит системному администратору уйму времени, избавив его от ответов на наиболее часто повторяющиеся вопросы пользователей X-терминалов. Правда, самые дотошные из них будут не один раз донимать системного администратора жалобами, что они, видите ли, хотели сделать то-то и то-то, а в инструкции про то, как это сделать, ничего не написано.

Аудит сети X-терминалов

По большей части X-терминальная среда работает очень стабильно. В моей практике (три раза стучу по дереву) не было ни одного случая зависания сервера X-терминалов. Тем не менее, это не избавляет системного администратора от необходимости поддержания системы в оптимальном работоспособном состоянии. А лучший путь для достижения этого, внедрение системы аудита X-терминальной сети. Под аудитом я понимаю контроль за загрузкой локальной сети, процессорным временем и также удаление неработоспособных программ из памяти X-терминал сервера. Обязательно при этом следить за безопасностью работы пользователей и выявлять попытки проникновения в вашу локальную сеть извне.

"Уборка мусора"

Не смотря на то, что по своей природе компьютеры лишены иррациональности, и все их поведение должно объясняться четко сформулированными правилами, иногда они ведут себя, мягко говоря, странно. Дело в том, что программы создаются людьми, и, видимо, поэтому от людей, которым, как вы знаете, свойственно ошибаться, им передаются черты, присущие живым организмам. Поэтому современные программы "виснут", делают ошибки, "болеют" (имеется в виду распространение компьютерных вирусов), и даже иногда по прихоти своих создателей совершают откровенно деструктивные действия. Пользователи уже смирились с такими "достоинствами" программного обеспечения, но чтобы их терпение не лопнуло, системному администратору необходимо максимально сгладить негативные последствия от работы таких программ. Частично такого эффекта можно добиться от внедрения системы "уборки мусора" операционной системы. Под

термином "уборка мусора" следует понимать ряд мероприятий, которые помогают операционной системе удалять ненужную информацию, такую как временные файлы, повисшие в памяти процессы, не использующиеся сервисы-демоны и т.д. Точно так, как современное общество не может жить без дворников, подметающих улицы, так само и операционная система требует от администратора помощи в ее обслуживании.

Для X-терминалов помощь в уборке мусора требуется в большей степени чем для обычных традиционных персональных компьютеров, так как X-терминал сервер может работать без выключения и перезагрузок на протяжении нескольких месяцев, а может даже и лет. С другой стороны, его бесперебойная работа позволяет создать автоматическую уборку мусора, которая будет выполняться сама без вмешательства системного администратора. Другими словами, один раз настроил и забыл.

За чем же должна следить система "уборки мусора" X-терминал сервера? Все зависит от того, какие программные продукты используют ваши пользователи в своей работе. Мне и вам очень повезло, так как в качестве серверной операционной системы мы выбрали Linux, которая сама по себе довольно надежна, так что про головную боль, вызванную "зависаниями" именно операционной системы, можно забыть. Но остаются прикладные программы, часть из которых иногда дает сбои. Лично я рекомендовал бы настроить систему X-терминалов и дать пользователям поработать с ней в течении месяца. За это время системному администратору нужно постараться собрать как можно больше информации про некорректное поведение программных продуктов. Для реализации задуманного лучше всего попросить пользователей X-терминалов сообщать системному администратору о каждом случае неправильного поведения их компьютеров. Тем самым мы убьем сразу двух зайцев: поможем пользователям освоиться в новой X-терминальной среде, а также соберем и систематизируем достаточно данных для повышения работоспособности и надежности сервера X-терминалов.

У системы X-терминалов, с которой мне довелось работать, у пользователей были такие претензии:

-иногда после включения X-терминала не запускался OpenOffice, а попытки перегрузить его ни к чему не приводили. Также иногда работа с ним завершалась сообщением об ошибке или "зависанием" данного программного продукта;

-иногда аварийно завершал свою работу браузер Galeon, особенно, когда одновременно было открыто очень много окон;

-не запускался почтовый клиент Evolution и браузер Galeon, как правило, всегда после аварийного завершения работы X-терминала (выключения питания в середине работы).

Начнем рассмотрение жалоб пользователей по порядку. Итак, офисный пакет OpenOffice предоставляет пользователю достаточно широкий спектр возможностей, но иногда ведет себя не совсем корректно. В частности, известны случаи "зависания" его в середине работы, а также аварийного завершения при определенных обстоятельствах. Особенно тяжело пользователю в случае "зависания" OpenOffice, так как даже нажатие на кнопку закрытия окна не имеет никакого результата. Более того, если пользователь даже воспользуется утилитой xkill, нет никакой гарантии, что офисный пакет повторно успешно запустится. С "зависаниями" офисного пакета ситуация понятна, к сожалению, он еще не может похвастаться 100% надежностью в работе, но случаются они очень редко и пока другой достойной альтернативы OpenOffice нет с этим приходится мириться. А вот с невозможностью его повторного запуска бороться нужно.

Рассмотрев ситуацию более пристально, мною было выявлено причину невозможности повторного запуска после аварийного выключения X-терминала или "зависания" OpenOffice. Дело в том, что в памяти сервера продолжает работать процесс soffice.bin, который и мешает повторному запуску. Так выглядит список процессов пользователя test, после аварийного выключения его X-терминала (на нем был запущен офисный пакет):

$ ps -u --User test

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

test 9536 0.0 0.0 3696 892 ? S 10:26 0:00 /usr/bin/icewm-session

test 9590 0.0 0.0 3216 628 ? S 10:26 0:00 /usr/bin/ssh-agent /usr/share/apps/switchdesk/Xclients.icewm

test 9591 0.0 0.1 3716 1136 ? RN 10:26 0:00 icewmbg

test 9592 0.0 0.1 6128 1264 ? R 10:26 0:00 icewmtray

test 9593 0.0 0.1 6620 1976 ? S 10:26 0:00 icewm

test 9608 0.6 4.6 136516 47712 ? S 10:26 0:02 /usr/lib/openoffice/program/soffice.bin

test 9632 0.0 0.3 9292 3156 ? S 10:26 0:00 /usr/apps/ROX-Filer/Linux-ix86/ROX-Filer

Как видно, физическое выключение X-терминала никак не повлияло на прекращение выполнения на сервере процессов пользователя. Офисный пакет soffice.bin продолжает работать в процессе под номером 9608. Для того, чтобы пользователь мог заново запустить OpenOffice, необходимо завершить выполнение этого процесса. Лаконичное и универсальное решение от имени суперпользователя:

```
# killall soffice.bin
```

Применять нельзя ни в коем случае, так как вместе с "повисшим" офисным приложением пользователя test будут принудительно закрыты и все другие, нормально работающие, экземпляры OpenOffice остальных пользователей X-терминальной сети. Необходимо завершить процесс от имени пользователя его запустившего. Для реализации задуманного я написал такой простой сценарий, а также ассоциировал его выполнение с кнопкой в панели быстрого запуска (см. рис. 8.4):

```
#!/bin/sh

oobin='soffice.bin'

killall $oobin
sleep 5
pids=$(ps -a --user=$USER | grep $oobin | awk '{print $1}')
```

```
for pid in $pids
do
   kill -9 $pid
   sleep 1
done
```

Запускать данный сценарий следует только от имени обычного пользователя. Как видно из его содержания, он сначала пытается завершить OpenOffice, посылая сигнал TERM командой killall, а затем после 5 секундной паузы, проверяет перестали ли работать процессы soffice.bin. Если это не так, то они завершаются уже командой kill, посылающей сигнал KILL (-9). Перед использованием данного сценария не забудьте сделать его файл исполняемым.

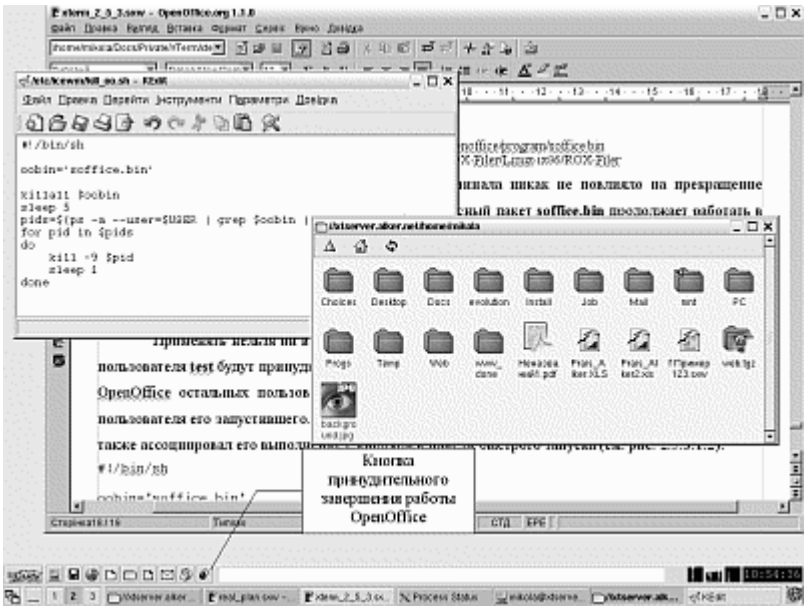

Рис. 8.4. Кнопка для принудительного завершения работы OpenOffice

Аварийному завершению браузера Galeon помочь никак нельзя. Единственным компромиссным решением является использование другого (более надежного браузера), а также совет пользователям не открывать одновременно слишком много его окон. А вот с невозможностью запуска этого браузера, как и почтового клиента Evolution, бороться можно. Причиной такого поведения этих двух программ является некорректное поведение сервисов-демонов, с которыми они связаны, а также аварийное завершение работы X-терминала. Вот так выглядит список процессов в памяти сервера при аварийном завершении работы пользователя X-терминала с именем guser:

```
$ ps -u --User guser
```

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

guser 10502 0.0 0.1 3692 1444 ? S 11:29 0:00 /usr/bin/icewm-session

guser 10556 0.0 0.0 3216 840 ? S 11:30 0:00 /usr/bin/ssh-agent /usr/share/apps/switchdesk/Xclients.icewm

guser 10557 0.0 0.1 3716 1492 ? RN 11:30 0:00 icewmbg

guser 10558 0.0 0.1 6132 1640 ? R 11:30 0:00 icewmtray

guser 10559 0.0 0.2 6636 2348 ? S 11:30 0:00 icewm

guser 10560 0.2 1.1 26668 12192 ? S 11:30 0:00 evolution

guser 10562 0.0 0.2 5996 2328 ? S 11:30 0:00 oafd --ac-activate --ior-output-fd=9

guser 10565 0.0 0.5 23252 6008 ? S 11:30 0:00 wombat --oaf-activate-iid=OAFIID:Bonobo_Moniker_wombat_Factory --oaf-ior-fd=10

guser 10569 0.0 0.3 11788 4036 ? S 11:30 0:00 bonobo-moniker-xmldb --oaf-activate-iid=OAFIID:Bonobo_Moniker_xmldb_Factory --oaf-ior-fd=12

guser 10573 0.1 1.2 51876 12372 ? S 11:30 0:00 evolution-mail --oaf-activate-iid=OAFIID:GNOME_Evolution_Mail_ShellComponent --oaf-ior-fd=14

guser 10575 0.0 0.2 3788 2216 ? S 11:30 0:00 /usr/bin/gconfd-1 12

guser 10582 0.0 0.6 23992 6684 ? S 11:30 0:00 evolution-addressbook --oaf-activate-iid=OAFIID:GNOME_Evolution_Addressbook_ShellComponent --oaf-ior-fd=

guser 10586 0.0 0.6 24408 6672 ? S 11:30 0:00 evolution-calendar --oaf-activate-iid=OAFIID:GNOME_Evolution_Calendar_ShellComponent --oaf-ior-fd=15

guser 10591 0.1 0.9 24160 9316 ? S 11:30 0:00 evolution-executive-summary --oaf-activate-iid=OAFIID:GNOME_Evolution_Summary_ShellComponentFactory --oa

guser 10593 0.0 0.5 21888 5552 ? S 11:30 0:00 evolution-alarm-notify --oaf-activate-iid=OAFIID:GNOME_Evolution_Calendar_AlarmNotify_Factory --oaf-ior-

guser 10633 0.3 1.9 57528 20000 ? S 11:31 0:00 /usr/bin/galeon-bin

Процессы с именем icewm относятся к оконному менеджеру, поэтому нас они не очень интересуют, в вот процессы с идентификаторами 10560 (evolution) и 10633 (galeon-bin) соответствуют почтовому клиенту и Интернет браузеру. Браузер Galeon связан с процессом gconfd-1. Чтобы узнать, что это за программа, выполним команду определения принадлежности файла к пакету:

$ rpm -qf /usr/bin/gconfd-1

GConf-1.0.9-10

Согласно информации про содержимое пакета:

$ rpm -qi GConf

Name : GConf Relocations: /usr

Version : 1.0.9 Vendor: ASPLinux

Release : 10 Build Date: Птн 25 Кві 2003 18:28:16

Install Date: Срд 25 Лют 2004 14:13:58 Build Host: arena.asplinux.ru

Group : System Environment/Base Source RPM: GConf-1.0.9-10.src.rpm

Size : 1901746 License: LGPL

Signature : DSA/SHA1, Пнд 05 Тра 2003 10:10:10, Key ID c96aa8e69b3c94f4

Packager : ASPLinux Team

URL : http://www.gnome.org/projects/gconf/

Summary : The Gnome Config System.

Description :

GConf is the GNOME Configuration database system.

GNOME is the GNU Network Object Model Environment. It is an easy to use, powerful, and highly configurable desktop environment.

GConf– это конфигурационная база данных проекта Gnome, а так как Galeon это Gnome-браузер, то ничего удивительного в том, что он использует gconfd-1 в своей работе. А как результат, для того чтобы браузер Galeon успешно запустился, достаточно завершить процессы с именами выполняющихся программ galeon-bin и gconfd-1.

"Уборка мусора"

С почтовым клиентом Evolition дела обстоят несколько сложнее, так как кроме уже известного демона gconfd-1, он использует также oafd. Чтобы разобраться к какому пакету принадлежит эта программа, воспользуемся уже знакомой нам командой:

```
$ rpm -qf /usr/bin/oafd
oaf-0.6.10-5
$ rpm -qi oaf
```

Оказывается, OAF – это система активации объектов среды Gnome, которая, в свою очередь, использует ORBit. Понятно, что без нее запустить почтовый клиент не получится. Как и для успешного запуска браузера Galeon, перед повторным использованием почтового клиента Evolution необходимо принудительно завершить процессы, которые с ним связаны. Выполнять завершение процессов в памяти сервера вручную не разумно, целесообразно написать сценарий для демона cron, который, например, ночь, когда никто не работает, будет удалять из памяти все процессы пользователей. Тем самым мы добьемся того, что каждое утро пользователь будет работать с абсолютно свежей средой, лишенной принадлежащих ему аварийно завершенных программ. Ниже представлен пример такого сценария, который я назвал kill_uj.sh:

```
#!/bin/sh

echo 'Killing all users jobs'

listusers=`ls /home/`
# Цикл по всем пользователям
for username in $listusers
do
    if [ $username != 'lost+found' ]; then
  echo "Work with user: $username"
  # Определение PID процессов пользователя
```

```
    jobs=`ps --User=$username -o "pid"| grep -v 'PID'`
    qntt_jobs=0
    # Попытка завершить выполняющиеся процессы
    for jobid in $jobs
    do
        echo "Terminating job with PID=$jobid"
        qntt_jobs=`expr $qntt_jobs + 1`
        `kill $jobid`
        sleep 1
    done
    # Принудительное завершения оставшихся процессов
    kjobs=`ps --User=$username -o "pid"| grep -v 'PID'`
    for kjobid in $kjobs
    do
        echo "Killing job with PID=$kjobid"
        `kill -9 $kjobid`
        sleep 1
    done
    if [ $qntt_jobs = 0 ]; then
        echo $'\tNo active jobs'
    else
        echo $'\t'"Were killed $qntt_jobs jobs"
    fi
      fi
done
echo "Done."
```

Теперь достаточно настроить автоматический запуск данного сценария каждый день, желательно в ночное время. Например, для запуска его каждый день в 1 час 15 минут следует от имени суперпользователя выполнить такую команду:

```
# EDITOR=mcedit crontab -e
```

Далее в появившемся любимом текстовом редакторе Midnight Commander необходимо добавить строку такого вида:

15 1 * * * /home/mikola/Temp/kill_uj.sh > /var/log/kill_uj.log

Сохранить файл и завершить работу с редактором (клавиша F10). Если вы все сделаете правильно, то на экране должна появиться надпись:

crontab: installing new crontab

Переменную окружения EDITOR можно не определять, если вы хорошо умеете работать с текстовым редактором по умолчанию. Например, я с ужасом смотрю на людей которые в своей работе используют vi (vim).

Задача crontab, которую мы создали, будет каждый день в 1:15 выполнять сценарий /home/mikola/Temp/kill_uj.sh, а результат его работы записывать в файл /var/log/kill_uj.log. Так что с утра по содержимому файла /var/log/kill_uj.log можно будет судить, сколько ненужных процессов было завершено.

Правда следует заметить, что используя такой сценарий, мы обязываем пользователей завершать свои программы перед уходом домой. Например, вместе со всеми процессами пользователя будет удалена также программа, скачивающая большой файл из сети Интернет, которую пользователь хотел оставить включенной всю ночь. Также автоматический запуск такого сценария может нарушить расписание пользовательских заданий crontab. Правда, я очень сомневаюсь, что ваши пользователи будут использовать crontab, да и к тому же всегда можно договориться, чтобы они запускали свои задания в другое время, например, с 21 до 23 часов.

Систему сбора "мусора" можно настроить также и на инспектирование процессов, запущенных и от имени суперпользователя root, а также других вспомогательных пользователей таких как: nobody, apache, gdm и т.д. Но следует заметить, что с подобными процессами проблем обычно всегда меньше.

Проверять также нужно присутствие двух копий одной и той же программы в памяти сервера, которые запущены от имени одного пользователя.

Иногда это может указывать на аномальное поведение программы. Ниже приводится пример команды, которая помогает отследить такие процессы:

```
$ ps -ax -o "user comm pid" | grep -v root | sort | uniq -d -w 26
apache   httpd        32473
mikola   bash         16033
mikola   kdeinit      11088
mikola   less         16716
mikola   man          16705
mikola   sh           16708
mikola   xterm        16031
nata     kdeinit      8927
ula      kdeinit      11014
```

Как видно по результату ее работы, ничего подозрительного в системе нет, так как для процессов httpd, bash, kdeinit, less, man, sh, xterm и kdeinit наличие нескольких копий в памяти считается вполне нормальным. Для уменьшения количества результатов были проигнорированы процессы, запущенные от имени супер-пользователя root (grep -v root).

Иногда на X-терминал сервере используется программное обеспечение, которое не совсем стабильное, но без которого никак нельзя обойтись. Это, например, может быть демон, который в процессе своей работы использует все больше и больше оперативной памяти, или просто программа, которая после долгого и интенсивного использования просто зависает или начинает работать неправильно. Для того чтобы бороться с последствиями работы подобных программ, можно, например, настроить систему так, чтобы через определенные промежутки времени она автоматически перезапускала проблемные демоны. В принципе, ничто не запрещает пойти на еще более радикальный метод, и выполнять перезагрузку сервера каждую ночь, но лично мне это не кажется очень разумным, так как компьютеры не любят процессов включения и выключения, а следовательно, чтобы продлить им жизнь, их нужно либо не выключать, либо не включать.

В главе, посвященной советам по настройке прикладных программ для X-терминалов, упоминалось о проблеме открытия и печати файлов в пакете OpenOffice. Причина скрывалась в разрастании файла ~/.openoffice/user/psprint/pspfontcache в домашнем каталоге пользователя. Давайте напишем сценарий для сервиса crontab, который будет автоматически удалять файлы pspfontcache при достижении ими больших размеров. Используя уже имеющиеся в нашем распоряжении наработки, получим такое содержимое этого сценария (файл kill_pscache.sh):

```
#!/bin/sh

echo 'Removing pspfontcache file for all users'

listusers=`ls /home/`
# Цикл по всем пользователям
for username in $listusers
do
   if [ $username != 'lost+found' ]; then
  echo "Work with user: $username"
  # Определение существования большого файла pspfontcache
  filepscache='/home/'$username'/.openoffice/user/psprint/pspfontcache'
  ksize=200 # минимальный размер файла в Кбайтах
  if [ -f $filepscache ]; then
     echo 'File exists'
     fsize=`find $filepscache -size +${ksize}k`
     if [ -z $fsize ]; then
          echo "But file too small"
     else
          echo 'Removing '$fsize
          `rm -f $fsize`
     fi
  else
```

```
        echo 'There is no file'
      fi
        fi
    done
    echo "Done."
```

Данный сценарий ищет файлы ~/.openoffice/user/psprint/pspfontcache размером более 200 Кбайт и удаляет их. После создания этого файла давайте добавим его в список задач утилиты crontab, например, для запуска во вторник и пятницу каждую неделю. Для этого выполним команду:

```
# crontab -e
```

В текстовом редакторе добавим строку, содержащую такие данные:

```
25 2 * * 2,5 /home/mikola/Temp/kill_pscache.sh > /var/log/kill_pscache .log
```

После сохранения файла расписания crontab, оптимизации запуска и открытия файлов OpenOffice, будет автоматически выполняться в 2:25 ночи каждую неделю по вторникам и пятницам.

Практически все современные дистрибутивы операционной системы Linux умеют выполнять ротацию системных журналов. Более того, чтобы упростить жизнь пользователям, они делают это по умолчанию сразу после установки операционной системы, так что настраивать практически ничего не нужно. Если же вас не устраивают установки ротации в вашей системе, то можете смело менять параметры файла конфигурации /etc/logrotate.conf и, конечно, файлы для ротации конкретных служб непосредственно из каталога /etc/logrotate.d/.

Из собственного опыта знаю, что в обслуживании операционная система Linux проще, чем это может показаться на первый взгляд. Построенная на открытых стандартах, она позволяет системному администратору получить ответ практически на любой вопрос, который касается состояния любой из ее составляющих. Просто нужно знать, где искать этот ответ, а это, как правило, постигается изучением документации и закрепляется опытом.

Оптимальный выбор программного обеспечения

Работа сервера X-терминалов напрямую зависит от используемого программного обеспечения. Поэтому очень важно правильно подобрать набор используемых прикладных программ. В выборе программного обеспечения системный администратор должен, в первую очередь, руководствоваться интересами удобства работы пользователей, а уже потом прилагать все усилия для обеспечения его правильной настройки. Это означает, что хорош тот программный продукт, который устраивает пользователя, даже если обслуживание его сложнее для системного администратора. В конце концов, компьютеры должны работать для нас, а не мы для них.

Также следует помнить, что хотя у пользователей операционной системы Linux и ограничены возможности установки программного обеспечения, при небольших модификациях они могут самостоятельно поставить программу в свой домашний каталог и пользоваться ею без разрешения системного администратора. Для X-терминальной сети, наверное, самой страшной будет ситуация, когда неправильно работающая программа будет нерационально потреблять ограниченные системные ресурсы, особенно такие как память. После того, как исчерпается оперативная память, сервер будет вынужден обращаться к файлу подкачки, а это довольно медленная операция, которая негативно отразится на производительности труда всех пользователей X-терминалов. Если страничный обмен в виртуальной памяти затянется надолго, то X-терминалы будут работать в дискретном режиме, то есть рывками: отобразил часть экрана, пауза, нарисовал окошко, пауза и т.д. Понятно, что такая ситуация мало кого из пользователей устроит. Выхода два: нарастить объем оперативной памяти или обеспечить механизм отслеживания "прожорливых" к памяти программ и принудительно завершать их. Субъективно могу заметить, что механизм виртуальной памяти в операционной системе Linux реализован не очень удачно. При обращении к разделу подкачки производительность системы очень сильно снижается. Следовательно, необходимо постоянно следить, чтобы размер используемой части файла подкачки не приближался к его общему размеру. Для этих целей можно

воспользоваться утилитой free. Приведу пример ее вывода на реальном сервере X-терминалов:

```
$ free -m -t
             total       used       free     shared    buffers     cached
Mem:          1005        983         22          0        191        408
-/+ buffers/cache:        383        622
Swap:         2000         72       1927
Total:        3006       1056       1949
```

Как видно из приведенного выше фрагмента, из 1 Гбайта оперативной памяти использовано 983 Мбайта, что составляет почти 96%. В то же время из файла подкачки почти в 2 Гбайта использовано только 72 Мбайта, или всего 3,6 %. Для того, чтобы файл подкачки не увеличивался очень быстро, необходимо следить за процессами, занимающими в памяти наибольший объем. Сделать это можно при помощи команды:

```
$ ps ax -o "user pid vsize comm" --sort vsize | tail
gdm      20077 33520 gdmgreeter
mikola   18185 78484 evolution-mail
mikola   18243 82792 galeon-bin
root     19953 85952 X
disp     17924 95272 galeon-bin
sasha    18604 108592 galeon-bin
sasha    16356 156932 soffice.bin
mikola   20169 168356 soffice.bin
ira      16192 179028 soffice.bin
ula      16964 193592 soffice.bin
```

Как видите, мы получили десятку самых требовательных к памяти процессов. Ни один из них даже не использует 200 Мбайт оперативной памяти. Вполне реально написать сценарий слежения за такими процессами и при достижении определенного порога (например, 500 Мбайт) используемой памяти прекращать их выполнение.

При выборе программного обеспечения для X-терминала необходимо сначала изучить его работу, в первую очередь обращая внимание на стабильность. Затем следует измерить требования к оперативной памяти. Если окажется, что памяти потребляется слишком много, то можно попытаться уменьшить этот показатель путем редактирования файлов конфигурации этого программного продукта. Хорошим примером может послужить оптимизация офисного пакета OpenOffice, что уже была описана в главе "Настройка терминалов".

К сожалению, очень сложно оценить требования программного обеспечения к пропускной способности локальной сети. Так, некоторые программы более чувствительны к этому фактору, а некоторые менее. Как правило, чем проще интерфейс программного продукта, тем лучше он работает на X-терминале. Особенно это касается программ с элементами анимации в интерфейсе, например, мигающими кнопками или подвижными иконками. Если настройки программы позволяют отключить эти эффекты, то это настоятельно рекомендуется сделать, так как тем самым вы значительно улучшите работу пользователей X-терминалов.

Также необходимо проверить работу программы в многозадачном режиме. В принципе, в операционной системе Linux я не встречал программы, которая бы не могла работать одновременно от имени нескольких пользователей, но определенные нюансы имеются. Представим ситуацию, когда один и тот же пользователь вошел в систему с двух различных X-терминалов. Как поведет себя программа в этом случае? Однозначный ответ на этот вопрос дать не получится. Например, если пользователь запустит на одном X-терминале OpenOffice, а потом попробует на другом открыть документ, то окно с документом отобразится на первом X-терминале. Иногда запущенная на одном X-терминале программа повторно попросту не открывается на другом. В общем, больших проблем с многозадачностью у X-терминалов нет, но проверить еще раз не мешает.

Для того, чтобы оценить качество выбранного программного обеспечения, которое будет работать на X-терминалах, необходимо изучить ситуацию с

загрузкой локальной сети и распределением процессорного времени. Опираясь на эти данные, можно утверждать о эффективности выбора того или иного программного продукта.

Анализ использования локальной сети

Локальная сеть для X-терминальной среды – это своего рода магистраль, которая соединяет все компьютеры в одно целое, если на каком либо из ее участков случается авария, то это сказывается на работе всей системы. Для управления локальной сетью предназначен даже специальный протокол SNMP (Simple Network Management Protocol – простой протокол управления сетью), который поддерживается практически всеми маршрутизаторами. Для работы с данным протоколом в операционной системе Linux существует специальный демон – smnpd. Кроме этого, существует еще довольно много способов учета сетевой статистики, которые можно использовать в своей работе. Также много и специального программного обеспечения, которое используется в этих целях. Одной из таких программ является MRTG (Multi Router Traffic Grapher), найти которую можно на ее официальном сайте http://www.mrtg.org.

Для тех пользователей, которые любят делать все своими руками, и при этом полностью контролировать процесс, я предлагаю построить простую систему учета трафика между X-терминалами и X-терминал сервером. При этом в своей работе мы воспользуемся программой iptables. Ну что ж, приступим.

Для начала нужно создать разрешающие правила для всех IP-адресов X-терминалов, причем как на входящие (цепочка INPUT), так и исходящие пакеты (цепочка OUTPUT). Для этого можно воспользоваться таким простым сценарием:

```
#!/bin/sh

iptables='/sbin/iptables'

echo 'Creating chaines for IP-accounting'
```

```
term_list='192.168.1.3 192.168.1.6 192.168.1.8
 192.168.1.13 192.168.1.14 192.168.1.15
 192.168.1.16 192.168.1.18 192.168.1.23 '

# Удаление всех существующих цепочек
$iptables -F
# Цикл по всем цепочкам
for term_ip in $term_list
do
   echo "Chain for IP: $term_ip"
   $iptables -A OUTPUT -d $term_ip -j ACCEPT
   $iptables -A INPUT -s $term_ip -j ACCEPT
done
echo "Done."
```

В приведенном выше примере в переменной term_list содержится список IP-адресов X-терминалов, за которыми ведется наблюдение. После запуска этого сценария (файл create_rip.sh) на экране должен быть такой вывод:

```
# ./create_rip.sh
Creating chaines for IP-accounting
Chain for IP: 192.168.1.3
Chain for IP: 192.168.1.6
Chain for IP: 192.168.1.8
Chain for IP: 192.168.1.13
Chain for IP: 192.168.1.14
Chain for IP: 192.168.1.15
Chain for IP: 192.168.1.16
Chain for IP: 192.168.1.18
Chain for IP: 192.168.1.23
Done.
```

Чтобы убедится в том, что цепочки установлены и работают правильно, следует от имени суперпользователя выполнить такую команду:

```
# /sbin/iptables -L -v -n
Chain INPUT (policy ACCEPT 121 packets, 10353 bytes)
 pkts bytes target     prot opt in     out     source               destination
  670 43744 ACCEPT     all  --  *      *       192.168.1.3          0.0.0.0/0
    0     0 ACCEPT     all  --  *      *       192.168.1.6          0.0.0.0/0
  741 47400 ACCEPT     all  --  *      *       192.168.1.8          0.0.0.0/0
  768 48836 ACCEPT     all  --  *      *       192.168.1.13         0.0.0.0/0
 1432  134K ACCEPT     all  --  *      *       192.168.1.14         0.0.0.0/0
  348 26960 ACCEPT     all  --  *      *       192.168.1.15         0.0.0.0/0
  765 48548 ACCEPT     all  --  *      *       192.168.1.16         0.0.0.0/0
 1064 70768 ACCEPT     all  --  *      *       192.168.1.18         0.0.0.0/0
 1071 97572 ACCEPT     all  --  *      *       192.168.1.23         0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 16084 packets, 5230K bytes)
 pkts bytes target     prot opt in     out     source               destination
45898   27M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.3
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.6
32880   16M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.8
57530   33M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.13
41469   19M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.14
32495   21M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.15
32172   17M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.16
90640   47M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.18
65262   30M ACCEPT     all  --  *      *       0.0.0.0/0            192.168.1.23
```

У себя вы должны получить похожие результаты. Особое внимание обратите на столбцы с именами "pkts" и "bytes" - они показывают количество и суммарный размер переданных пакетов в цепочке. Интересно, но с помощью iptables можно сразу узнать какие X-терминалы не работают (выключены), так

как их счетчики содержат нулевые значения (в примере это X-терминал с IP-адресом 192.168.1.6). Особое внимание следует обратить на отсутствие правил для пересылаемого трафика FORWARD. Это связано с тем, что общение X-терминалов с сервером идет по довольно простой схеме и надобность в пересылке пакетов на другие другие узлы локальной сети отсутствует. Очень часто программным брандмауэром вообще запрещена пересылка пакетов для сервера X-терминалов:

```
$ cat /proc/sys/net/ipv4/ip_forward
0
```

Важно также помнить, что отсутствие возможности пересылки пакетов отнюдь не мешает использовать ресурсы глобальной компьютерной сети пользователям X-терминалов. Это становится возможным благодаря тому, что программы пользователей непосредственно выполняются на сервере X-терминалов, который имеет подключение к Интернет.

Но вернемся к анализу использования сетевого трафика X-терминалами. Метод учета статистики использования локальной сети заключается в том, чтобы через равные промежутки времени получать значения счетчиков iptables и сохранять их в файле. Примером сценария, предназначенного для этих целей, может быть сценарий, написанный на языке Perl (имя файла acc_ips.pl), текст которого приведен ниже:

```
#!/usr/bin/perl -w
# - - - - - - - - - - - - - - - - - - - - -
# Сценарий для учета статистики использования
# локальной сети X-терминалами
# - - - - - - - - - - - - - - - - - - - - -

use strict;
use POSIX;

my $log_dir = '/var/log/ip_acc/';
my $fip_stat = $log_dir.'ip_comm_stat.txt';
```

```
my $fip_log = $log_dir.'ip_stat.log';
my $iptables = '/sbin/iptables';
my %stat = ();

# Считать информацию из файла со счетчиками
if ( open (FSLOAD, "<$fip_stat") ) {
    while () {
  chomp;
  my ($ip, $t_input, $ti_count, $t_output, $to_count) = split /\s/;
  ${$stat{$ip}}{total_input}=$t_input;
  ${$stat{$ip}}{total_icount}=$ti_count;
  ${$stat{$ip}}{total_output}=$t_output;
  ${$stat{$ip}}{total_ocount}=$to_count;
    }
    close (FSLOAD);
} else { `touch $fip_stat` }

# Обнулить счетчики INPUT и одновременно показать их значения
my @input_counters = `$iptables -L -v -n -x -Z INPUT`;
for (@input_counters) {
    next if (/INPUT|target/); # пропустить неинформативные строки
    my ($bytes, $ip) = ( split ) [1,7];
    ${ $stat{$ip} }{input} = $bytes;
    ${ $stat{$ip} }{input_count} = ($bytes) ? 1 : 0;
}

# Обнулить счетчики OUTPUT
my @output_counters = `$iptables -L -v -n -x -Z OUTPUT`;
for (@output_counters) {
    next if (/OUTPUT|target/); # пропустить неинформативные строки
    my ($bytes, $ip) = ( split ) [1,8];
```

```
    ${$stat{$ip}}{output} = $bytes;
    ${$stat{$ip}}{output_count} = ($bytes) ? 1 : 0;
}

# Подсчет статистики
for (keys %stat) {
    my $href = $stat{$_};
    ${$href}{total_input} += ${$href}{input};
    ${$href}{total_output} += ${$href}{output};
    ${$href}{total_icount} += ${$href}{input_count};
    ${$href}{total_ocount} += ${$href}{output_count};
}

# Сохранить данные в файлах
my @log_data = ();
push @log_data, strftime ("%Y-%m-%d %H:%M:%S", localtime());
open (FSSAVE, ">$fip_stat") or die "Cant save stat file $!\n";
for (sort keys %stat) {
    my $href = $stat{$_};
    print FSSAVE join("\t", ($_, ${$href}{total_input}), ${$href}{total_icount},
    ${$href}{total_output}, ${$href}{total_ocount}), "\n";
    push @log_data, ${$href}{input}, ${$href}{output};
}
close (FSSAVE);
unless (-f $fip_log) {
    my $header = "'Date       Time     ' . join (' ', sort keys %stat) .'"';
    `echo $header > $fip_log`
}
open (FLSAVE, ">>$fip_log") or die "Cant save log file $!\n";
print FLSAVE join(" ", @log_data), "\n";
close (FLSAVE);
```

Анализ использования локальной сети (продолжение)

Для того чтобы обеспечить его систематический запуск, можно воспользоваться демоном crond, добавив к его расписанию такое задание:

```
*/5 * * * * /home/mikola/Temp/acc_ips.pl > /dev/null
```

Теперь файл acc_ips.pl будет запускаться каждые пять минут.

Из текста сценария видно, что он для хранения итоговых данных использует файл /var/log/ip_acc/ip_comm_stat.txt примерно такого содержания:

```
$ cat /var/log/ip_acc/ip_comm_stat.txt
192.168.1.13    8467907        13      57711815        17
192.168.1.14    39284308       41      207864777       41
192.168.1.15    121265196      41      2309630116      41
192.168.1.16    43640700       41      175535249       41
192.168.1.18    47282564       41      208459130       41
192.168.1.23    37398265       41      190043178       41
192.168.1.3     27732053       41      210263758       41
192.168.1.6     0              0       0               0
192.168.1.8     133934451      41      272410910       41
```

Значение его первого столбца, я думаю, понятно – это IP-адрес X-терминала. Далее идет столбец с суммарным значением переданного от X-терминала трафика (измеряется в байтах) и количество значимых циклов измерений. В двух следующих столбцах содержится значение полученного X-терминалом трафика от сервера (измеряется в байтах), а также соответствующее значение результативных циклов передачи. Отдельно следует остановиться на значениях столбцов 3 и 5. В них накапливаются значения суммарного количества результативных сеансов работы X-терминала по передаче и приему информации, то есть если за исследуемый период передачи данных не было (трафик равен нулю), то такой период не засчитывается (это бывает, как правило, в моменты, когда X-терминал выключен). Такая методика позволяет определить реальные требования к пропускной способности сети работающего X-терминала. Для этого достаточно разделить суммарный трафик на количество циклов работы.

Интересные показатели получаются и при исследовании соотношения входящего к исходящему трафику X-терминалов. Из данных, приведенных в файле примера, они соотносятся от 2:1 до 19:1 (в среднем как 8:1), причем исходящий от X-терминала трафик всегда меньше, чем входящий со стороны сервера.

Опираясь на имеющиеся данные, можно рассчитать среднюю нагрузку на локальную сеть работающего X-терминала. Зная, что интервал измерений составлял 5 минут, получаем итоговое среднее значение в 2,54 Мбит/с для 8 работающих X-терминалов. Разброс значений по отдельным X-терминалам составляет от 0,1 Мбит/с до 1,51 Мбит/с. Если взять за базис значение, превышающее максимальный показатель в два раза (3 Мбит/с), то можно утверждать, что одного сетевого адаптера для локальной сети в 100 Мбит/с вполне хватит для обслуживания более 30 X-терминалов. При увеличении числа дистанционных пользователей при той же нагрузке на локальную сеть следует подумать о установке на сервере дополнительной сетевой карты и разделении сети X-терминалов на две части.

Кроме подсчета суммарной статистики использования сети, приведенный в этой главе сценарий протоколирует значения полученного и переданного трафика в разрезе времени по каждому X-терминалу в файле /var/log/ip_acc/ip_stat.log. Примерное содержимое этого файла следующее (фрагмент начала файла для первых трех X-терминалов):

```
$ head /var/log/ip_acc/ip_stat.log
Date       Time     192.168.1.13   192.168.1.14     192.168.1.15 ...
2005-03-16 11:00:00 202220 1974784 465832  2830744  599456  36231640 ...
2005-03-16 11:05:00 397308 2940188 750304  4893740  971704  53733504 ...
2005-03-16 11:10:00 363346 2956304 1134768 6435300  1020156 54497962 ...
2005-03-16 11:15:00 362792 3202772 1964460 15158388 921712  53991880 ...
2005-03-16 11:20:00 408920 3052312 1046284 8280276  864792  53805376 ...
2005-03-16 11:25:00 483872 3324800 1610860 9228680  2614080 56573740 ...
2005-03-16 11:30:00 462532 2896620 484736  2863668  1452416 54804268 ...
2005-03-16 11:35:00 390116 2809076 2908592 9010220  952960  53569736 ...
```

2005-03-16 11:40:00 427500 2803576 768200 8274688 894228 53615280 ...

Первые два столбца этого файла указывают на время сбора статистики, а последующие пары столбцов соответствуют значениям переданного и полученного сетевого трафика X-терминала, IP-адрес которого указан в строке заголовка. Для того, чтобы проанализировать нагрузку на локальную сеть в более наглядном виде, обратите внимание на диаграмму, что приведена на рис. 8.5.

Рис. 8.5. Диаграмма использования пропускной способности локальной сети X-терминалами

На диаграмме представлена нагрузка на локальную сеть от каждого из 9-ти X-терминалов в период с 11-00 до 16-35, которая была измерена в течении обычного рабочего дня. Как видно по результатам, ни один X-терминал не превысил порог в 2,5 Мбит/с, однако не следует забывать о том, что это усредненные данные. Из-за достаточно большого интервала измерений (5 минут), реальная потребность X-терминала в пропускной способности локальной сети может быть гораздо больше, поэтому полученные результаты вовсе не означают комфортной работы пользователя через сетевой адаптер на 10 Мбит/с. На графике отчетливо видно, что X-терминал с IP-адресом 192.1.1.15 потребляет трафика минимум 1,4 Мбит/с, в то время как требования к пропускной способности остальных пользователей гораздо скромнее, порядка 0,2 Мбит/с. Повышенные требования к пропускной способности сети от данного X-терминала объясняются особенностями программного обеспечения, которое использовал работающий на нем пользователь. Например,

анимационный баннер, содержащийся на открытой веб-странице, будет давать постоянный прирост сетевого трафика на X-терминале. К подобным эффектам приводят также прослушивание музыки и просмотр видео-роликов.

Изучая график интенсивности сетевой загрузки X-терминалов, можно точно определить, кто из пользователей ходил обедать, а кто предпочел проиграть на компьютере весь обеденный перерыв (время на графике с 12-00 до 13-00).

Распределение процессорного времени

Вы даже представить себе не можете, как "криво" сделанная программа может свести на нет всю мощь аппаратного обеспечения вашего компьютера. Хотите пример? Откройте в текстовом редакторе из проекта KDE (kedit) любую гипертекстовую страничку, в которой много табличных данных и выполните замену двух символов "TD" на что угодно без учета регистра. Например, я на подобную "замену" в файле размером 47 Кбайт потратил более 5 минут, и это на компьютере с центральным процессором Pentium-4 и тактовой частотой в 2,6 ГГц! Причем, что характерно, все 5 минут загрузка центрального процессора была почти 100%.

На подобные случаи можно не обращать внимание, работая за традиционным персональным компьютером, когда пользователь единолично распоряжается ресурсами своего процессора, но в распределенной X-терминальной среде дела обстоят значительно сложнее. Даже не смотря на то, что аппаратные возможности X-терминал сервера, как правило, на высоте, неэффективное программное обеспечение может значительно ухудшить производительность труда пользователей. Особенно печально, что это коснется всех пользователей без исключения. Если более пристально рассмотреть систему управления диспетчеризации процессов в операционной системе Linux, то окажется, что она сделана довольно грамотно. И по умолчанию, она сама прекрасно справится с подавляющим большинством ситуаций повышения нагрузки на сервер, но в случае "повисшей" (зациклившейся) пользовательской задачи она нам мало чем поможет.

Какие меры можно принять для минимизации негативного влияния "прожорливых" к процессорному времени программ? В ручном режиме наблюдения за системой сделать это очень просто, достаточно запустить программу мониторинга системных процессов типа top, и корректировать приоритеты выполнения того или иного процесса. Для достижения подобных целей у системного администратора есть утилита renice.

Например, от команды top вы получили такую информацию по процессам, которые запустил пользователь с именем mikola:

```
 13:11:40  up 18 days, 33 min,  5 users,  load average: 0,00, 0,13, 0,18
190 processes: 172 sleeping, 18 running, 0 zombie, 0 stopped
CPU states:   0,2% user   0,8% system   0,0% nice   0,0% iowait  98,8% idle
Mem:  1030108k av,  982128k used,   47980k free,       0k shrd,  168508k buff
                742092k actv,    6552k in_d,   21204k in_c
Swap: 2048248k av,   92916k used, 1955332k free                401504k cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME CPU COMMAND
 6370 mikola    15   0 63256  54M 20480 R     0,0  5,4   0:19   0 soffice.bin
 5519 mikola    15   0 24308  23M 13540 S     0,0  2,3   0:17   0 galeon-bin
 6843 mikola    15   0 15056  14M 11212 S     0,0  1,4   0:00   0 sim
 5543 mikola    15   0 14960  11M  5332 S     0,0  1,1   0:01   0 evolution-mail
 6858 mikola    15   0  8364 8364  6000 S     0,0  0,8   0:00   0 kdeinit
 5529 mikola    15   0  7804 7804  3224 S     0,0  0,7   0:00   0 evolution
 6853 mikola    15   0  5936 5936  4168 S     0,0  0,5   0:00   0 kdeinit
 8238 mikola    15   0  4588 4588  2320 R     0,0  0,4   0:00   0 xterm
```

Как видно из примера, все задания пользователя выполняются с приоритетом 15 (nice=0). Для того, чтобы лучше понять методику распределения приоритетов выполнения в операционной системе Linux, следует быть знакомым с параметром nice, который специфичен для каждого процесса в памяти. Значение nice может изменятся от -20 (наивысший приоритет) до 19 (самый низкий приоритет). Отрицательные значения nice

доступны только суперпользователю, поэтому с ними следует быть особенно осторожными, впрочем как и со значением 19. Граничные значения приоритетов выполнения с одной стороны приведут к тому, что процесс будет потреблять практически все процессорное время системы (мешая даже демонам), а с другой стороны, процесс у которого nice=19 будет "просыпаться" только в моменты полного бездействия системы, что в X-терминальной среде бывает очень редко.

Выполнять изменение приоритетов очень просто. Для того, чтобы понизить приоритет процесса с именем soffice.bin вам достаточно использовать такую команду:

```
# renice +1 6370
6370: old priority 0, new priority 1
```

Теперь данный процесс будет выполняться с приоритетом меньшим на единицу, чем другие процессы пользователя.

Кроме этого, команда renice позволяет изменять приоритеты выполнения не только отдельных процессов по их идентификационному номеру (PID), но также влиять на все процессы, запущенные определенным пользователем (параметр -u) или принадлежащих пользовательской группе (параметр -g).

Проблема в использовании утилиты renice заключается в том, что она влияет на приоритет выполнения уже запущенных задач. Например, вы как системный администратор понизили приоритет выполнения пользовательского процесса. Если пользователь завершит соответствующую программу и запустит ее снова, то процесс будет выполняться с прежним приоритетом (по умолчанию). С другой стороны, вместо активного наблюдения за поведением системы, у системного администратора должно быть много других занятий, поэтому ручное применение команды renice – это не самый хороший выбор. Неплохо было бы настроить автоматический механизм реагирования на "жадные" к процессорному времени процессы.

Для реализации системы автоматического наблюдения и реагирования за процессорным временем сервера X-терминалов необходимо сделать ряд предположений. Во-первых, в поле деятельности данного комплекса должны

попадать только процессы конечный пользователей (с распределением нагрузки между демонами операционная системы Linux справляется хорошо и без нашей помощи). Во-вторых, объектами управления должны быть процессы, которые на протяжении определенного интервала времени потребляют больше, например, 80% ресурсов центрального процессора сервера. В-третьих, если понижение приоритета выполнения процесса не помогло, то через определенное время (например, 30 минут) выполнение такого процесса принудительно прекращается.

Все исходные данные для построения подобной системы можно получить из системы /proc, а если не хотите разбираться в форматах ее файлов, то можете воспользоваться такими системными утилитами как, например, ps. За шаблон системы управления приоритетами пользовательских задач можно взять следующий сценарий, написанный на языке Perl:

```
#!/usr/bin/perl -w

use strict;

my $watch_users = 'nata,mikola,ira,ula,sasha,terminator,yana,serg,disp,mitya';
my %top_ps = ();
my $max_cpu_usage = 85; # Граничное значение использование ЦПУ
my $time_sleep = 20; # Время в секундах между измерениями
my $max_to_renice = 3; # К-во минут после, которых процессу понижается приоритет
my $times_to_kill = 15; # К-во минут для завершения процесса

print "Getting user process CPU usage\n";
while(1) {
   # Получаем список уникальных процессов
   my @procs = `ps h --User=$watch_users -o "pid user comm %cpu" --sort %cpu`;
   my %curr_ps = ();
```

```
  for (reverse @procs) {
 my ($pid, $user, $command, $cpu) = split;
 if ($cpu > $max_cpu_usage) {
    $top_ps{$pid}++;
    $curr_ps{$pid} = $cpu;
 } else { last }
  }
  for (keys %top_ps) {
 if ( !exists($curr_ps{$_}) ) { delete $top_ps{$_} }
 else { print "$_ => $top_ps{$_} ($curr_ps{$_})\n" }
  }
  # Проверка "жадных процессов"
  for (keys %top_ps) {
 if ( $top_ps{$_} == $max_to_renice*(60/$time_sleep) ) {
    `renice +1 -p $_`;
    print "PID $_ was reniced\n";
 }
 if ( $top_ps{$_} == $times_to_kill*(60/$time_sleep) ) {
    `kill $_`;
    print "PID $_ was send TERM signal\n";
 }
  }
  sleep($time_sleep);
}
```

Как видно из текста сценария, он получает и анализирует данные от команды ps, которая собирает информацию только о процессах, запущенных от имени реальных пользователей (переменная $watch_users). При достижении установленной границы (переменная $max_cpu_usage) использования ресурсов центрального процессора, процесс пользователя "ставится на учет", и если нагрузка будет сохраняться в течении тех минут, которые отведены в переменной $max_to_renice, то данному процессу приоритет будет понижен на

1. Если же и после этого использование центрального процессора будет в тех же пределах, то через время указанное в переменной $times_to_kill "провинившемуся" процессу будет послан завершающий сигнал TERM.

Конечно, данный сценарий нельзя рассматривать как панацею от работы зависших или неправильно работающих программных продуктов. Если всмотреться в него внимательнее, то очевидными станут потенциальные проблемы его использования. Также вызывает сомнение правильность установки значений времени работы процесса до понижения ему приоритета, и время принудительного завершения процесса, в прочем как и эмпирическая граница максимума процессорного времени. Для адекватного определения значений этих переменных следует исследовать работу конкретной среды X-терминалов, принимая во внимание аппаратные возможности сервера, используемое программное обеспечение и количество одновременно работающих пользователей (например, чем больше пользователей и мощнее сервер, тем ниже должно быть значение переменной $max_cpu_usage).

В любом случае я предлагаю рассматривать данный сценарий, как отправную точку в деле расширенного управления распределением процессорного времени сервера X-терминалов. И пускай его возможности очень скромны, зато он вполне способен разрешить целый ряд проблем, особенно в той ситуации, когда системный администратор находится далеко от X-терминальной среды.

Контроль за использованием дискового пространства

Чуть не забыл упомянуть о еще одном очень хорошем способе контроля за работой многопользовательской системы: дисковых квотах. Вообще-то, дисковые квоты очень любят системные администраторы, и практически также сильно не любят пользователи. Первые используют их как надежный способ контроля за пространством на диске, а вторые воспринимают их как средство ограничения своих прав и свобод. Объективно, дисковая квота – это действительно ограничение объема информации, создаваемой пользователем или группой пользователей, но при наличии такого ограничения гораздо легче управлять системой, особенно состоящей из бездисковых X-терминалов.

Первое, что необходимо сделать – это установить пользовательские квоты для каталога /home. Например, если подраздел жесткого диска, который примонтирован как /home имеет размер 20 Гбайт, то для системы с 20 пользователями, каждому из них можно установить ограничение (квоту) в 1 Гбайт. На самом деле можно установить даже более высокое ограничение, так как ситуация, когда все пользователи полностью используют отведенное им дисковое пространство – маловероятна. На мой взгляд, более рациональным способом установки пользовательских дисковых квот является метод, когда всем пользователям выделяется квота меньше, чем максимально доступная. Для приведенного выше примера, можно ограничиться 500 Мбайт. В случае, если пользователи исчерпают свою квоту, им в индивидуальном порядке можно увеличить ее значение. Если не хранить в домашнем каталоге видеофильмы и mp3-музыку, то места среднестатистическому пользователю нужно совсем не много. Посмотрите на пример использования дисковой подсистемы каталога /home, реальной X-терминальной среды:

```
# du -s --block-size=M /home/* | sort -n -r
5211M   /home/mikola
2450M   /home/sasha
355M    /home/test
194M    /home/ula
129M    /home/ira
71M     /home/serg
49M     /home/pasha
36M     /home/terminator
30M     /home/nata
18M     /home/disp
15M     /home/yana
11M     /home/luba
10M     /home/mitya
4M      /home/na
4M      /home/guser
```

1M /home/tanya

Как видно из результата приведенной выше команды, только некоторые пользователи превысили предел в 1 Гбайт, а остальные далеки даже от отметки в 500 Мбайт.

Отдельного внимания заслуживает установка групповых дисковых квот для каталогов совместного доступа к файлам. Применение в этой ситуации пользовательских квот не совсем оправдано, так как изменения в файлы вносятся коллективно и перекладывать за это ответственность на одного пользователя в принципе не справедливо. Выбор типа квоты (жесткая или мягкая) также зависит от конкретной ситуации, хотя в большинстве случаев предпочтение отдает именно мягкому ограничению.

Концепция квот прекрасно подходит к идеологии работы бездисковых X-терминалов. Благодаря тому, что вся информация хранится на сервере, системному администратору приходится фактически управлять единым хранилищем данных, в котором находятся файлы всех пользователей. Если сравнить это с обслуживанием традиционных персональных компьютеров, то для X-терминалов накладные расходы практически не зависят от количества пользователей, чего к сожалению нельзя сказать про полноценные компьютеры. Другими словами, квоты хороши там, где используется совместный доступ к файла и где в системе много пользователей, а X-терминал сервер - это как раз и есть тот случай.

К сожалению, практическое описание методов и приемов использования дисковых квот выходит за рамки данной книги. Более подробную информацию вы можете найти в специализированных изданиях или в документации к операционной системе Linux (команды quotaon, quotaoff, edquota и т.д.). Поверьте, настроить систему дисковых квот в Linux не сложнее, чем разобраться в других сервисных функциях этой операционной системы.

Завершая рассмотрение нюансов администрирования сети X-терминалов, мне бы хотелось сделать акцент на том, что в работе этих бездисковых станций совсем не много проблем. Упрошенные методы администрирования и надежность работы этого программно-аппаратного комплекса позволяют

системному администратору снизить удельные затраты на обслуживание пользователей. Более того, приобщаясь к наследию UNIX систем в лице операционной системы Linux, вы и ваши пользователи начнете видеть компьютерный мир в несколько ином свете. Такие понятия как работа с файлами, доступ к устройствам компьютера и права пользователей приобретут несколько другие значения. Даже, казалось бы, такое привычное слово, как персональный компьютер, и то будет теперь восприниматься как обыкновенная альтернатива в составе автоматизированного рабочего места.

Глава 9. Работа с периферией на терминале

Благодаря особенностям распределенной структуры X-терминалов, работать с периферией на них немного сложно. Это связано с тем, что для правильного использования локального устройства, нужен механизм доступа к нему, а как известно после запуска X-сервера терминала, пользователь практически работает на сервере, а как результат, доступ к локальным устройствам утрачен. Проблема заключается также в том, что X-сервер правильно работает с такими устройствами X-терминала как монитор, видеокарта, манипулятор “мышь” и клавиатура, но ему абсолютно все равно какой принтер вы используете или какая звуковая карта установлена на терминале. Фактически для работы с устройствами, которые не входят в “юрисдикцию” X-сервера, приходится придумывать изощренные методы доступа. Правда, создатели проекта LTSP значительно помогли нам в этом, и теперь доступ к принтерам, настройка звука на X-терминалах и даже запуск локальных приложений уже не являются чем-то сверхъестественным. Хотя нюансы использования все же имеются. Целью данной главы является предоставление читателям нужной информации, которая понадобится им при настройке работы с такими устройствами как: принтер, сканер, звуковая карта и флоппи-дисковод, а также отдельное внимание уделяется локальному запуску приложений.

Подключение к терминалу принтера (LPT, USB)

Хорошо если у вас есть настоящий сетевой принтер, для подключения и работы которого вообще не требуется персонального компьютера. "Воткнул"

его в локальную сеть через кабель RJ-45 и пользуйся в свое удовольствие. Но на практике получается, что данные принтеры не очень распространены, и в подавляющем большинстве случаев нам приходится работать на принтерах с интерфейсами USB или LPT, а уж им для работы персональный компьютер просто необходим. Как же подключить и использовать эти принтеры на X-терминалах, если настройка подсистемы печати выполняется на сервере. К счастью, в проекте LTSP предусмотрена методика использования локальных принтеров X-терминалов. Суть ее очень проста. Специальное программное обеспечение, которое запускается на X-терминале, слушает определенный порт TCP/IP (по умолчанию, 9100) и принимает через него удаленные задания на печать от всех клиентов локальной сети, включая и не X-терминалы. Фактически принтер, подключенный к X-терминалу, ведет себя подобно настоящему сетевому принтеру, так как из параметров настройки имеется только IP-адрес и порт. Лучше всего рассмотреть эту ситуацию на конкретном примере.

Перед тем как начать описание подключения и настройки принтера к X-терминалу, коротко остановимся на подсистеме печати операционной системы Linux. В мире “пингвинов” нет единой для всех системы печати, и из представленных на рынке между собой успешно конкурируют CUPS, LPRng, LPD, GNUlpr, PPR, PDQ, CPS, а также многие другие. Это разнообразие вносит небольшое смятение в процесс принятия решения, какую же систему печати выбрать, так как одновременно можно использовать только одну из них. С X-терминалами будет успешно работать практически любая из них, но лично я порекомендовал бы использовать именно CUPS (официальный сайт http://www.cups.org/). Можно сказать, что CUPS - это относительно новая разработка (1998 год) компании Easy Software Products (сайт http://www.easysw.com/). Эта фирма скромно называет ее переносимой и расширяемой системой печати для операционных систем UNIX. Первая стабильная версия была выпущена в свет в октябре 1999 года. Имея 30-ти летнюю историю развития операционных систем UNIX, при разработке CUPS были учтены все недостатки, которые неизбежно сопровождали подсистему

печати UNIX-подобных операционных систем. Для обеспечения обратной совместимости с уже существующими и популярными подсистемами печати, CUPS эммулирует команды как BSD Unix, так и команды System V. Имея дружественный веб-интерфейс управления, расширяемую модульную структуру и поддержку огромного числа новых принтеров, CUPS практически обеспечила себе место в компьютерах конечных пользователей. Даже если вы не имеете практического опыта использования CUPS, все равно на данный момент это лучшая подсистема печати и немедленно начать ее изучение просто необходимо. Иными словами, вы можете использовать любую подсистему печати, но примеры данной главы будут ориентированы на CUPS. Кто знает, может закончив чтение, вы так же, как и я, станете ее почитателем.

Для примера выполним настройку лазерного принтера HP LaserJet 1100 для терминала ulax, который имеет IP-адрес 192.168.1.3. Чтобы сразу показать все преимущества использования технологии CUPS, воспользуемся простым способом конфигурации принтера через Web-интерфейс.

Запускаем на сервере или X-терминале любимый браузер и вводим адрес доступа к Web-интерфейсу системы CUPS: http://localhost:631/ (см. рис. 9.1).

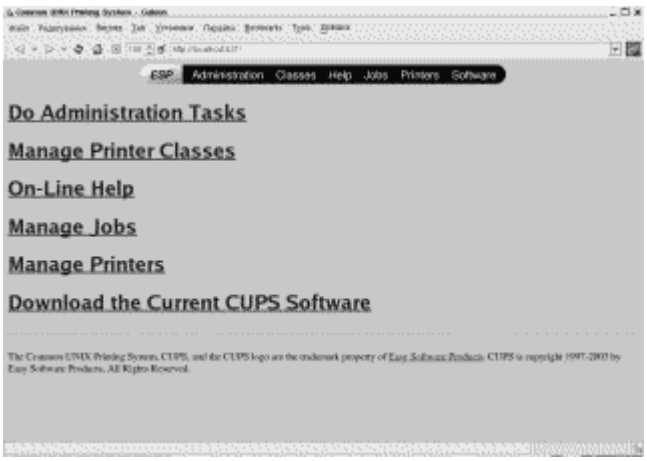

Рис. 9.1. Главная страница веб-интерфейса подсистемы печати CUPS

Для доступа к настройкам принтера нам нужны права администратора, для чего необходимо перейти по ссылке: “Do Administration Tasks”. В появившемся диалоговом окне авторизации вводим имя пользователя “root” и соответствующий пароль. Если пароль правильный, то на экране должна появиться похожая страничка (см. рис. 9.2):

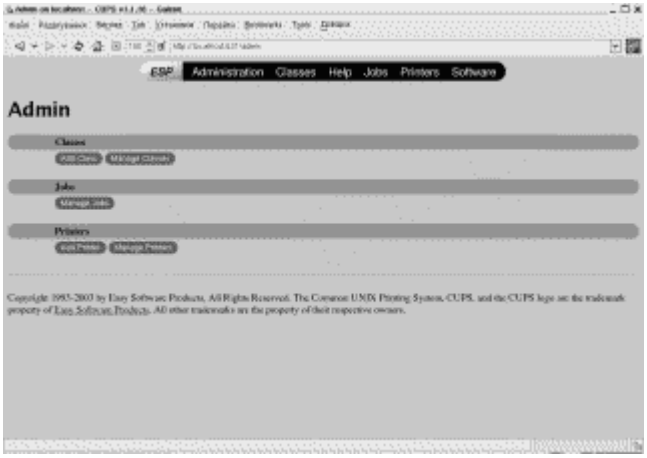

Рис. 9.2. Страница администрирования веб-интерфейса подсистемы печати CUPS

На странице администрирования http://localhost:631/admin переходим по ссылки “Add Printer”, которая соответствует диалогу добавления нового принтера (http://localhost:631/admin/?op=add-printer). При добавлении нового принтера на первом шаге необходимо выбрать для него имя, например, A_LJ110, а также заполнить необязательные поля “Location” и “Description” (см. рис. 9.3). Следует помнить, что использование в имени принтера символов национальных алфавитов, а также пробела запрещено. После заполнения информационных полей нажимаем кнопку “Continue”.

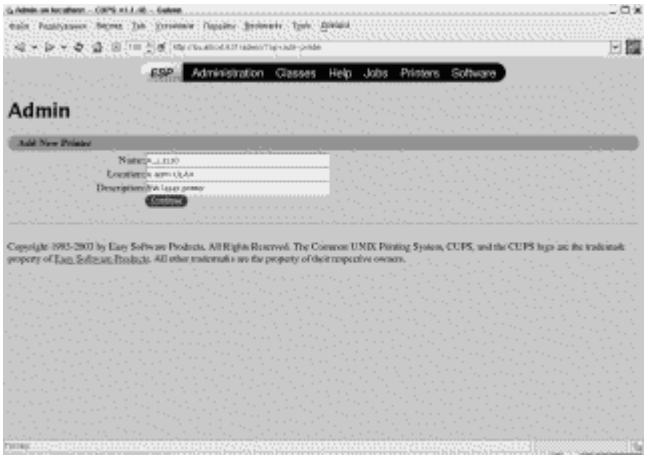

Рис. 9.3. Первый шаг в добавлении нового принтера через веб-интерфейс подсистемы печати CUPS

На втором шаге создания нового принтера выбираем для него тип подключения: AppSocket / HP JetDirect (см. рис. 9.4). И нажимаем кнопку “Continue”.

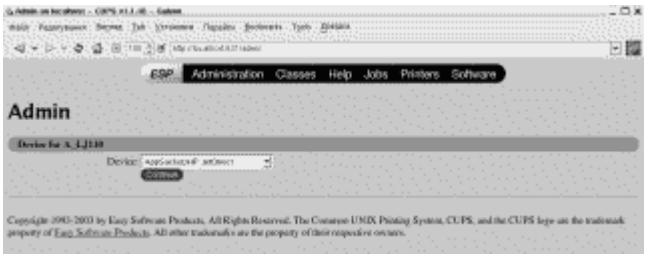

Рис. 9.4. Выбор типа подключения принтера через веб-интерфейс подсистемы печати CUPS

Подключение к терминалу принтера (LPT, USB) (продолжение)

Теперь осталось для нового принтера выбрать сетевой протокол (socket://), его IP-адрес (192.168.1.3) и, если нужно, порт TCP/IP (9100). В нашем случае это socket://192.168.1.3:9100 (см. рис. 9.5). Нажатием на кнопку “Continue” переходим на страницу выбора производителя принтера (см. рис. 9.6).

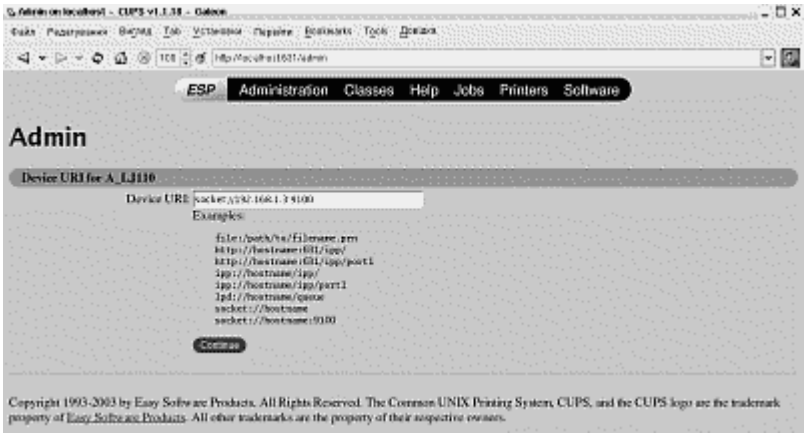

Рис. 9.5. Выбор сетевого адреса принтера через веб-интерфейс подсистемы печати CUPS

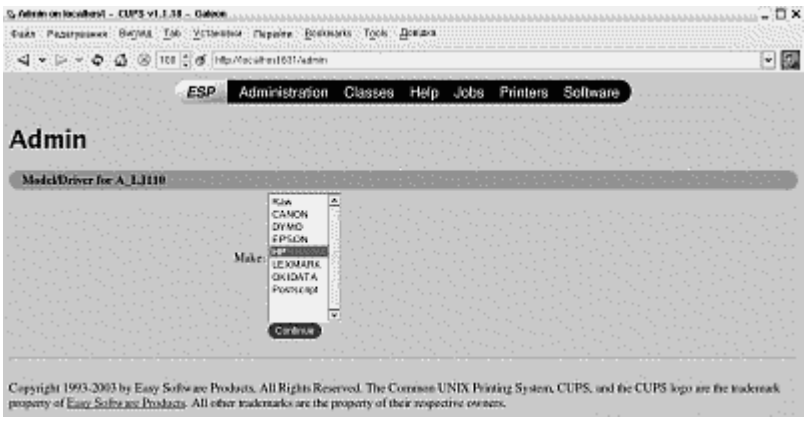

Рис. 9.6. Выбор производителя принтера через веб-интерфейс подсистемы печати CUPS

Когда производитель принтера выбран (в данном случае HP), следует определиться с драйвером, а точнее, фильтром печати для системы CUPS. Нажимаем на кнопку “Continue” и переходим на страницу выбора фильтра

печати (см. рис. 9.7). Для принтера HP LaserJet 1100 прекрасно подойдет драйвер “HP LaserJet Series CUPS v1.1 (en)”. Для подтверждения выбора служит кнопка “Continue”.

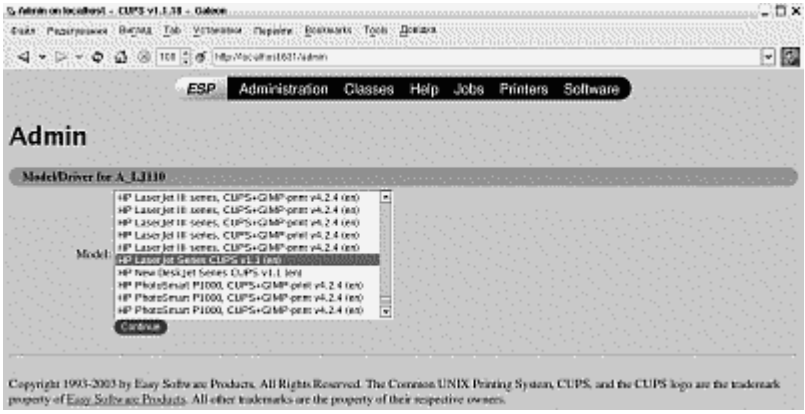

Рис. 9.7. Выбор драйвера (фильтра) печати для принтера через веб-интерфейс подсистемы печати CUPS

Про успешное добавление принтера в системе CUPS свидетельствует надпись “Printer A_LJ110 has been added successfully” (см. рис. 9.8). Для того, чтобы попасть в диалоговое окно свойств данного принтера, необходимо перейти по ссылке “A_LJ110”.

Рис. 9.8. Завершение добавления нового принтера через веб-интерфейс подсистемы печати CUPS

Как видно из рисунка 9.9, новый принтер с именем A_LJ110 готов к работе. Но для того, чтобы на нем действительно можно было печатать, необходимо выполнить еще два простых действия: проверить его конфигурацию и подготовить X-терминал к поддержке системы печати.

Рис. 9.9. Главная страница параметров принтера при доступе через веб-интерфейс подсистемы печати CUPS

Подключение к терминалу принтера (LPT, USB) (продолжение)

Для проверки конфигурации принтера нажимаем на кнопке “Configure Printer” и попадаем на страницу тонкой настройки параметров печати (см. рис. 9.10). Даже беглого взгляда достаточно для того, чтобы определить низкое 300 dpi качество печати по умолчанию (можно заменить на 600 dpi) и неправильный формат бумаги “US Letter” (нужно заменить на A4). Вносим необходимые изменения и нажимаем на кнопку “Continue”. Теперь смело можем утверждать, что настройка принтера средствами CUPS завершена. Дело за малым: объяснить X-терминалу, что принтер подключен именно к нему.

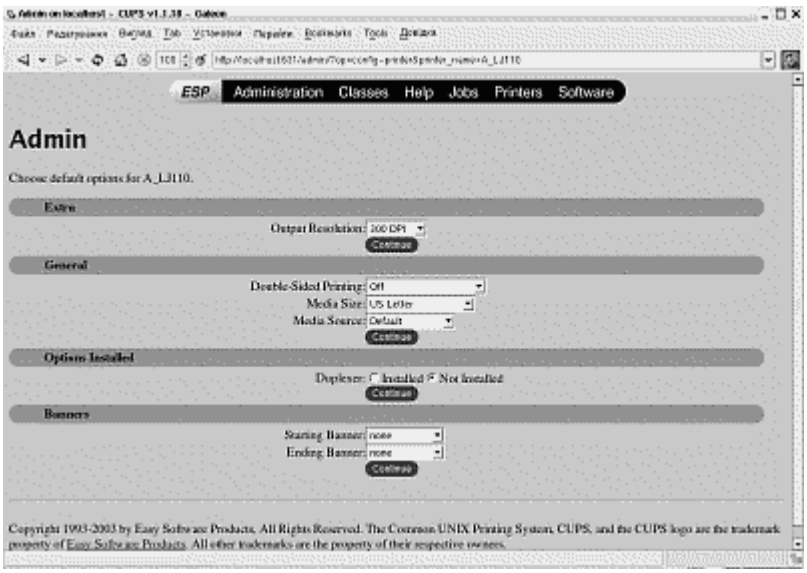

Рис. 9.10. Страница конфигурации принтера при доступе через веб-интерфейс подсистемы печати CUPS

Как вы уже наверное поняли, настройка принтера в CUPS никак не связана с его интерфейсом (LPT, USB или COM). Эти параметры необходимо указать для X-терминала, к которому подключен принтер. Перед настройкой, в первую

очередь, проверяем работает ли порт подключения на данном терминале. Для этого достаточно зайти в BIOS X-терминала и проверить соответствующий раздел в программе конфигурации. Для параллельного порта LPT можно выбирать любой режим SPP, EPP или ECP. Когда настройка BIOS завершена, переходим к внесению изменений в соответствующий раздел файла /opt/ltsp/i386/etc/lts.conf.

Подключение принтера к параллельному порту (LPT)

Так как принтер HP LaserJet 1100 имеет параллельный интерфейс, то в раздел ulax файла /opt/ltsp/i386/etc/lts.conf следует добавить такие строчки:

```
[ulax]
    PRINTER_0_DEVICE   = /dev/lp0
    PRINTER_0_TYPE     = P
```

А так как принтер у нас один, то используем параметры PRINTER_0_DEVICE, и PRINTER_0_TYPE. Вообще проект LTSP поддерживает до 3-х принтеров, подключенных к одному X-терминалу. Для второго и третьего принтера параметры были бы PRINTER_1_DEVICE, PRINTER_1_TYPE, и соответственно PRINTER_2_DEVICE, PRINTER_2_TYPE. В данном конкретном случае PRINTER_0_DEVICE = /dev/lp0 – означает подключение к параллельному порту (LPT1:), а PRINTER_0_TYPE = P – параллельный интерфейс (для принтеров, что подключены к последовательным портам типа COM или USB нужно использовать значение “S”).

Кроме этого, для настройки принтера могут использоваться также следующие опции файла /opt/ltsp/i386/etc/lts.conf:

PRINTER_0_PORT – определяет какой из портов TCP/IP будет использоваться принтером. По умолчанию, PRINTER_0_PORT = 9100, PRINTER_1_PORT = 9101, PRINTER_2_PORT = 9102.

PRINTER_0_SPEED – этот параметр применим только для принтеров с последовательным интерфейсом. Значение по умолчанию 9600 бод. Для каждого из 3-х принтеров можно установить отдельное значение.

PRINTER_0_FLOWCTRL – определяет тип контроля передачи данных и применим только для принтеров с последовательным интерфейсом. По умолчанию используется значение “S”, которое соответствует “Software (XON/XOFF)”. Опционально можно выбрать также “H” - “Hardware (CTS/RTS)”.

PRINTER_0_PARITY – данный параметр применим также только для последовательных принтеров, и определяет тип контроля четности. Допустимые значения: “E” - Even, “O” - Odd или “N” - None. Значение по умолчанию PRINTER_0_PARITY = N. Применимо к каждому принтеру отдельно.

PRINTER_0_DATABITS – также применимо только для последовательных принтеров. Означает номер бита данных, может принимать значения: 5, 6, 7 или 8. Значение по умолчанию 8. Для каждого из 3-х доступных принтеров можно устанавливать независимые значения.

PRINTER_0_WRITE_ONLY – некоторые принтеры могут определять, что порт, к которому они подключены, поддерживает двунаправленный обмен данными. И в своей работе они могут использовать эту возможность, то есть ожидать обратного ответа от драйвера принтера через этот порт. Установка этого параметра в “Y” позволит запретить принтеру использовать двунаправленный обмен данными. Значение по умолчанию “N”.

Подключение принтера к порту USB

К сожалению, не все X-терминалы имеют порт USB, особенно если они собраны из старого ПК на базе процессора i486 или Pentium. А подавляющее большинство современных принтеров комплектуется именно этим интерфейсом. В любом случае будем считать, что в вашем X-терминале есть порт USB и именно к нему нужно подключить USB-принтер. Первая часть настройки драйвера в системе CUPS ничем не отличается от описанной выше. Расхождения в настройки наблюдаются только в параметрах соответствующего раздела файла /opt/ltsp/i386/etc/ltsp.conf. Так, для работы USB-принтера этот файл должен содержать такие данные:

MODULE_01 = usb-uhci

MODULE_02 = printer

PRINTER_0_DEVICE = /dev/usb/lp0

PRINTER_0_TYPE = S

Как видно из примера, дополнительно в ядро X-терминала загружаются два модуля usb-uhci и printer (иногда usb-ohci или ehci-hcd). В отличии от принтера, подключенного к порту с параллельным интерфейсом, у USB-принтера используется другое символическое устройство /dev/usb/lp0. Также указан последовательный интерфейсдоступа к принтеру PRINTER_0_TYPE = S (Serial).

Если к одному X-терминалу подключены сразу 3 принтера, то их символические устройства будут называться /dev/usb/lp0, /dev/usb/lp1 и /dev/usb/lp2.

На X-терминале об успешной загрузке модулей для поддержки USB-принтера могут свидетельствовать такие сообщения:

$ dmesg | grep -i usb

usb.c: registered new driver usbdevfs

usb.c: registered new driver hub

usb-uhci.c: $Revision: 1.275 $ time 00:17:28 Feb 4 2004

usb-uhci.c: High bandwidth mode enabled

usb-uhci.c: USB UHCI at I/O 0xde00, IRQ 10

usb-uhci.c: Detected 2 ports

usb.c: new USB bus registered, assigned bus number 1

hub.c: USB hub found

usb-uhci.c: v1.275:USB Universal Host Controller Interface driver

usb.c: registered new driver usblp

printer.c: v0.11: USB Printer Device Class driver

hub.c: new USB device 00:1f.2-2, assigned address 2

printer.c: usblp0: USB Bidirectional printer dev 2 if 0 alt 1 proto 2 vid 0x03F0 pid 0x1017

Вообще, перед настройкой принтера на X-терминале, рекомендуется посетить сайт http://www.linuxprinting.org/, и узнать из него как данный принтер

поддерживается операционной системой Linux. Также на этом сайте вы найдете рекомендации по использованию того или иного фильтра печати для данного принтера, так как в среде Linux является нормой работа принтера с различными драйверами, которые в зависимости от модели печатающего устройства отличаются качеством и скоростью работы.

Следует также заметить, что кроме модуля usb-uhci, для старых компьютеров иногда следует использовать другой модуль: usb-ohci. Также необходимо помнить, что для обеспечения работы интерфейса USB 2.0 используется другой модуль: ehci-hcd (правда, я очень сомневаюсь в целесообразности сборки X-терминала на такой современной аппаратной основе).

Подключенный к X-терминалу принтер может использоваться всеми компьютерами локальной сети, включая даже рабочие станции с MS Windows (для этого необходимо наличие пакета Samba на сервере и его правильная настройка).

Музыка на терминале

Звук, а точнее поддержка звуковых карт на X-терминале более чем реальна. Для того, чтобы из колонок вашей рабочей станции лилась музыка нужно выполнить дополнительные настройки (прямо как в народной мудрости, без труда не выловишь и рыбку из пруда). В первую очередь нам необходим специальный пакет ltsp_sound. Для версии проекта LTSP 3.0 он содержится в файле ltsp_sound-3.0.1-i386.tgz. К сожалению, он не доступен в стандартном формате пакета rpm, а следовательно устанавливать его нужно по-другому. Первое, что нужно сделать, это разархивировать пакет:

```
$ tar xvzf ltsp_sound-3.0.1-i386.tgz
ltsp_sound/
ltsp_sound/etc/
ltsp_sound/etc/profile.d/
ltsp_sound/etc/profile.d/ltsp-sound.sh
ltsp_sound/i386/
ltsp_sound/i386/sbin/
```

ltsp_sound/i386/sbin/nasd

ltsp_sound/i386/sbin/esd

ltsp_sound/i386/bin/

ltsp_sound/i386/bin/aumix-minimal

ltsp_sound/i386/bin/getltscfg

ltsp_sound/i386/lib/

ltsp_sound/i386/lib/libaudiofile.so

ltsp_sound/i386/lib/libaudiofile.so.0

ltsp_sound/i386/lib/libaudiofile.so.0.0.1

ltsp_sound/i386/lib/libesddsp.so

ltsp_sound/i386/lib/libesddsp.so.0

ltsp_sound/i386/lib/libesddsp.so.0.2.22

ltsp_sound/i386/lib/libesd.so

ltsp_sound/i386/lib/libesd.so.0

ltsp_sound/i386/lib/libesd.so.0.2.22

ltsp_sound/i386/etc/

ltsp_sound/i386/etc/lts.conf.sound

ltsp_sound/i386/etc/rc.sound

ltsp_sound/i386/etc/README.sound

ltsp_sound/i386/etc/audiolist

ltsp_sound/install.sh

ltsp_sound/COPYING

ltsp_sound/README

ltsp_sound/naslibs/

ltsp_sound/naslibs/libaudio.a

ltsp_sound/naslibs/libaudiooss.so.1.0

ltsp_sound/naslibs/libaudio.so.2.2

ltsp_sound/nasbins/

ltsp_sound/nasbins/auinfo

ltsp_sound/nasbins/auplay

Делаем каталог ltsp_sound активным:

$ cd ltsp_sound

Теперь, как советует файл README, запускаем сценарий установки пакета поддержки звука для проекта LTSP (конечно от имени супер-пользователя root):

./install.sh

LTSP Sound v3.0.1 by Andrew Williams (mistik1@geeksinthehood.net)

Originally created by Boris Reisig (boris@microtrader.com)

Bash Version=2.05b.0(1)-release

./install.sh: line 1: ldconfig: command not found

You need libc.so.6 to continue.

Выполнив эту стандартную операцию, видим на экране сообщение об ошибке, которое означает отсутствие утилиты ldconfig и разделяемой библиотеки libc.so.6. Странно, вроде бы это стандартные части операционной системы. Осуществляем поиск в базе данных slocate файла ldconfig:

locate ldconfig

warning: locate: warning: database /var/lib/slocate/slocate.db' is more than 8 days old

/usr/share/man/man8/ldconfig.8.gz

/usr/share/man/ru/man8/ldconfig.8.gz

/sbin/ldconfig

Как видно, файл есть, значит проблема не в его отсутствии, а в скрипте install.sh. Делаем выборку из файла install.sh по слову ldconfig:

$ grep ldconfig install.sh

Exist=`ldconfig -v | grep $lib`;

Exist=`ldconfig -v | grep $filename`;

Сразу все становится на свои места. Используя команду su, я забыл передать параметр -l, который отвечает за вход суперпользователя в оболочку. Исправляем ошибку:

$ su -l

Password: ******

#

По новому запускаем программу установки пакета поддержки звука на X-терминале от проекта LTSP:

```
# ./install.sh
LTSP Sound v3.0.1 by Andrew Williams (mistik1@geeksinthehood.net)
Originally created by Boris Reisig (boris@microtrader.com)
Bash Version=2.05b.0(1)-release
Installing Base package.
Please make sure to unset LD_PRELOAD before you use this
if have a previous ltsp Sound package installed
About to install the libs, Are you sure you want to continue? [Y/N]:y
```

После проверки готовности системы для подтверждения установки необходимо нажать клавишу “Y”, что мы и делаем. Инсталятор копирует нужные файлы в каталог /opt/ltsp/i386/, а также устанавливает специальные переменные окружения:

```
Putting files in /opt/ltsp
Setting up Environment variables adding ltsp-sound.sh
Would you like to install the nasd sound utilities package? [Y/N]:n
```

Новый вопрос сценария install.sh касается установки утилит сервера звука nasd. Тут следует сразу оговориться, что пакет ltsp_sound поддерживает работу с двумя звуковыми серверами-демонами: nasd и esd. Они оба позволяют передавать звук по локальной сети, а так как esd знаком нам по проекту Gnome, то рациональнее использовать именно его. Поэтому от поддержки nasd мы отказываемся, нажатием на клавишу “N”:

```
cp: cannot stat `/nasbins/auplay': No such file or directory
chmod: failed to get attributes of `/bin/auplay': No such file or directory
cp: cannot stat `/nasbins/auinfo': No such file or directory
chmod: failed to get attributes of `/usr/sbin/auinfo': No such file or directory
Installation *Complete*

!!NOTE: Please read "/opt/ltsp/i386/etc/lts.conf.sound" for more info
        You *SHOULD* turn down your workstation speakers as this
```

package is installed default to ramp the sound volume
on loading of the sound daemon

Have Fun!!!

Не смотря на сообщения об ошибках (они относятся к пропущенной установке звукового демона nasd), можно считать установку поддержки звука для X-терминалов завершенной. Проверим наличие нового файла в директории /opt/ltsp/i386/etc/:

$ ls -l /opt/ltsp/i386/etc/lts.conf.sound

-rw-r--r-- 1 root root 7888 Лют 22 2002 /opt/ltsp/i386/etc/lts.conf.sound

Теперь перейдем к непосредственной настройке звука на конкретных X-терминалах. Для этого в раздел файла /opt/ltsp/i386/etc/lts.conf, который относится к нужному X-терминалу, следует добавить параметр "SOUND = Y", а также указать тип используемого звукового демона “SOUND_DAEMON = esd”. Для интегрированных звуковых устройств, а также для звуковых карт с интерфейсом PCI достаточно указать автоматическую конфигурацию, используя параметр “SMODULE_01 = auto”. При внесения изменений в файл /opt/ltsp/i386/etc/lts.conf рекомендуется также перевести X-терминал в текстовый режим (RUNLEVEL = 3), и посмотреть на сообщения системы поддержки звука. В моем случае при загрузке я увидел на экране следующую надпись:

...

Loading Sound Modules

Running Sound Server

- accepting connections on port 16001

aumix: error opening mixer

Типичное сообщение об ошибке свидетельствует про успешный запуск звукового сервера-демона ESD на порту TCP/IP номер 16001, и ошибке инициализации звуковой карты. Система тем самым сообщает, что ей не удалось самостоятельно определить тип звуковой карты и загрузить

соответствующий модуль ядра. Неудивительно, ведь я для примера использовал ISA карту Yamaha на чипсете opl3-sa2. Но если вы, как и я - счастливый обладатель звуковой карты ISA, то не стоит расстраиваться, так как после небольшого поиска нужного звукового модуля и указания правильных его параметров вам удастся заставить работать в X-терминале практически любую звуковую карту. Для ISA-карт обязательной является загрузка модулей: soundcore, sound, ad1848 и uart401. Последним загружается непосредственно модуль звуковой карты, в нашем случае opl3sa2. Раздел главного файла конфигурации /opt/ltsp/i386/etc/lts.conf посвященного поддержке звука на X-терминале будет выглядеть вот так:

```
SOUND           = Y
SOUND_DAEMON    = esd
SMODULE_01      = soundcore
SMODULE_02      = sound
SMODULE_03      = ad1848
SMODULE_04      = uart401
SMODULE_05      = opl3sa2
```

Перегружаем X-терминал и смотрим, что за сообщение появились на экране. К сожалению, сообщение об ошибке крайне не утешительное:

```
Loading Sound Modules
Loading: soundcore
Loading: sound
Loading: ad1843
ad1843/cs4248 codec driver Copyright © by Hannu Savolainen 1993-1996
Loading: uart401
Loading opl3sa2
opl3sa: io, mss_io, irq, dma, and dma2 must be set
/lib/modules/2.4.24-ltsp-1/kernel/drivers/sound/opl3sa2.o: init_module: Invalid argument
Hint: insmod error can be caused by incorrect module parameters, including invalid IO or IRQ parameters
```

/lib/modules/2.4.24-ltsp-1/kernel/drivers/sound/opl3sa2.o: insmod /lib/modules/2.4.24-ltsp-1/kernel/drivers/sound/opl3sa2.o failed

/lib/modules/2.4.24-ltsp-1/kernel/drivers/sound/opl3sa2.o: insmod opl3sa2 failed

ERROR! loading module: opl3sa2 failed !

Press to continue

Нажимаем клавишу “Enter” и продолжаем загрузку X-терминала. Из сообщения об ошибке понятно, что при загрузке модуля ядра /lib/modules/2.4.24-ltsp-1/kernel/drivers/sound/opl3sa2.o возникли ошибки, которые связаны с неправильным использованием обязательных параметров модуля:io, mss_io, irq, dma и dma2. Выбор правильных параметров для модуля ядра, соответствующего звуковой карте ISA, довольно сложное занятие, так как требует хорошего знания принципов работы устройств этого типа. Для начала следует перегрузить X-терминал и посмотреть какие параметры были присвоены звуковой карте системой BIOS и распознаны ядром Linux. Итак, на начальном экране загрузки промелькнули такие строки:

ISA/PNP device listing ...

Card No.	Device No.	DMA	IRQ	Device Name
1	0	0,1	5	OPL3-SA2 Sound Board
1	1	0,0	NA	OPL3-SA2 Sound Board
1	2	0,0	NA	OPL3-SA2 Sound Board
1	3	0,0	11	OPL3-SA2 Sound Board

Какую информацию можно извлечь из данного примера. Во-первых, irq=5. Во-вторых, dma=0, а dma2=1. В-третьих, нам осталось только узнать какой диапазон используется для параметров io и mss_io. Чтобы добыть эту информацию, обращаемся за помощью в глобальную сеть Интернет. Например, запрос для поисковой системы Google может выглядеть так: “linux opl3sa2 io mss_io” (см. рис. 9.11).

Рис. 9.11. Страница результатов поиска параметров модуля ядра opl3sa2

Буквально вторая ссылка в результатах поиска привела нас к нужной информации: выдержки из рабочего файла /etc/modules.conf:

alias sound-slot-0 opl3sa2

options opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1

options opl3sa2 io=0x388

Вооружившись новым знанием, добавляем обязательные параметры в соответствующий раздел файла /opt/ltsp/i386/etc/lts.conf, а также необязательный, но использующийся аргумент mpu_io:

SOUND = Y

SOUND_DAEMON = esd

SMODULE_01 = soundcore

SMODULE_02 = sound

SMODULE_03 = ad1848

SMODULE_04 = uart401

SMODULE_05 = opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=1 dma2=0

Перегружаем X-терминал и следим за сообщениями на экране:

Loading Sound Modules

Loading: soundcore

Loading: sound

Loading: ad1843

ad1843/cs4248 codec driver Copyright © by Hannu Savolainen 1993-1996

Loading: uart401

Loading opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=1 dma2=0

opl3sa2: Found OPL3-SA2 (YMF711)

Running Sound Server

- accepting connections on port 16001

Похоже, что все в порядке. Запускаем наш любимый музыкальный проигрыватель (у меня это xmms) и настраиваем его на работу через звуковой демон ESounD. Нажимаем кнопку “Play” и тишина ... . Слегка только подергивается эквалайзер и время проигрывания музыкального файла отсчитывается очень медленно. Очень похоже на ошибку в параметрах звукового модуля ядра. Проверим еще раз установленные параметры: точно, ошибка в значениях dma и dma2 (их аргументы перепутаны местами). В новой редакции правильный фрагмент файла /opt/ltsp/i386/etc/lts.conf должен выглядеть вот так:

SOUND = Y

SOUND_DAEMON = esd

SMODULE_01 = soundcore

SMODULE_02 = sound

SMODULE_03 = ad1848

SMODULE_04 = uart401

SMODULE_05 = opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1

Перегружаем X-терминал и видим, что теперь в модуль opl3sa2.o переданы новые параметры:

Loading opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1

opl3sa2: Found OPL3-SA2 (YMF711)

Running Sound Server

- accepting connections on port 16001

Да здравствует звук на X-терминале!

Узнать какие обязательные параметры требует модуль ядра можно и другим способом, при этом вовсе не обязательно обращаться за помощью к глобальной паутине. Достаточно вспомнить какие утилиты есть в самой операционной системе Linux. Так, узнать предназначение модуля ядра можно воспользовавшись такой командой:

$ /sbin/modinfo -d opl3sa2

"Module for OPL3-SA2 and SA3 sound cards (uses AD1848 MSS driver)."

А список его параметров покажет эта же утилита, только с другим параметром:

$ /sbin/modinfo -p opl3sa2

io int, description "Set I/O base of OPL3-SA2 or SA3 card (usually 0x370. Address must be even and must be from 0x100 to 0xFFE)"

mss_io int, description "Set MSS (audio) I/O base (0x530, 0xE80, or other. Address must end in 0 or 4 and must be from 0x530 to 0xF48)"

mpu_io int, description "Set MIDI I/O base (0x330 or other. Address must be even and must be from 0x300 to 0x334)"

irq int

dma int, description "Set MSS (audio) first DMA channel (0, 1, 3)"

dma2 int, description "Set MSS (audio) second DMA channel (0, 1, 3)"

ymode int, description "Set Yamaha 3D enhancement mode (0 = Desktop/Normal, 1 = Notebook PC (1), 2 = Notebook PC (2), 3 = Hi-Fi)"

loopback int, description "Set A/D input source. Useful for echo cancellation (0 = Mic Rch (default), 1 = Mono output loopback)"

isapnp int, description "When set to 0, ISA PnP support will be disabled"

multiple int, description "When set to 0, will not search for multiple cards"

В подавляющем количестве случаев принудительную загрузку обязательных модулей таких как soundcore, sound, ad1848 и uart401 выполнять не обязательно, так как через механизм зависимостей модулей ядра они будут загружены автоматически. Например, если в предыдушем примере ограничиться только непосредственно звуковым модулем карты в параметрах файла /opt/ltsp/i386/etc/lts.conf, то звук все равно будет работать:

SOUND = Y

SOUND_DAEMON = esd

SMODULE_01 = opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1

Убедимся в правильной загрузке модулей при помощи команды lsmod (модули, относящиеся к работе звука в ядре: opl3sa2, ad1848, mpu401, sound и soundcore):

```
# /sbin/lsmod
Module          Size  Used by
opl3sa2         6048   1
ad1848         19008   0 [opl3sa2]
mpu401         18832   0 [opl3sa2]
sound          52448   1 [opl3sa2 ad1848 mpu401]
soundcore       3248   6 [sound]
floppy         46972   0
nfsswap         2640   1
serial         48208   0 (unused)
ne2k-pci        4288   1
8390            5808   0 [ne2k-pci]
crc32           2848   0 [8390]
```

Конечно, использование звука на X-терминалах сопряжено с рядом ограничений. Во-первых, для запуска дополнительных модулей ядра, обеспечивающих поддержку звука и самого звукового демона (esd или nasd) требуется дополнительная оперативная память. Во-вторых, как было уже сказано раннее, передача звука по сети значительно увеличивает требования к ее пропускной способности. В любом случае не рекомендуется использовать параметр “SOUND = Y” в разделе [Default] файла /opt/ltsp/i386/etc/lts.conf. Хорошо, если пользователи будут использовать звук в качестве элемента интерфейса (звуки при нажатии на кнопки, разворачивание и сворачивание окна и т.п.), тогда нагрузка на сеть будет распределенной. Если же они все

одновременно будут слушать музыку, то звуковой трафик составит более 90% пропускной способности локальной сети X-терминалов.

Еще одним важным ограничением сетевого звука являются задержки при его передаче. Так как обрабатывается звуковой поток сервером, а затем в разжатом виде передается звуковому демону X-терминала, то небольшие временные задержки присутствуют.

Кроме упомянутых выше параметров настройки поддержки звука на X-терминалах (SOUND, SOUND_DAEMON и SMODULE_XX) в файле /opt/ltsp/i386/etc/lts.conf можно использовать еще и следующие аргументы:

VOLUME – значение громкости динамиков компьютера, установленное в процентах. Не смотря на процентное значение (от 0 до 100), знак % не следует добавлять после числа. Значение по умолчанию: VOLUME = 75;

MIC_VOLUME – определяет чувствительность микрофона X-терминала. Как и предыдущий параметр принимает процентное значение. По умолчанию, установлено в 75 (MIC_VOLUME = 75);

CD_VOLUME – громкость звука от проигрывателя компакт-дисков. Значение по умолчанию CD_VOLUME = 75.

Для закрепления методов настройки ISA-звуковых карт на терминалах приведу параметры для активации некоторых звуковых карт.

Звуковые карты на чипсете Crystal:

SMODULE_01 = cs4232 irq=5 io=0x220 dma=1

ESS ES1868:

SMODULE_01 = sound dmabuf=1

SMODULE_02 = opl3 io=0x388

SMODULE_03 = sb isapnp=1

Легендарный Creative Sound Blaster Vibra 16:

SMODULE_01 = sb io=0x220 irq=5 dma=1

Для PCI устройств и интегрированных звуковых контроллеров настройка намного проще. Достаточно указать в параметрах соответствующего раздела файла /opt/ltsp/i386/etc/lts.conf такие строки, если вы используете демон звука ESounD:

SOUND = Y

SOUND_DAEMON = esd

SMODULE_01 = auto

Для пользователей, предпочитающих работать с демоном nasd, достаточно изменить лишь одну строку:

SOUND = Y

SOUND_DAEMON = nasd

SMODULE_01 = auto

Смотрим видеофильмы на терминале

Если использование звука X-терминалами вас не остановило, то давайте пойдем еще дальше, и будем смотреть на X-терминалах видеофильмы. Настраивать практически ничего не придется, так как звук уже есть, а с графикой у X-терминалов и так все было в порядке.

Самым узким местом при просмотре фильмов на X-терминале является пропускная способность локальной сети. Тут распределенная структура клиент-сервер системы X-Windows сыграла с нами злую шутку, и все ее преимущества работают против нас. При передаче видеоизображения через локальную сеть полностью отсутствует какая-либо акселерация, и фактически видеофильм передается набором статических изображений (довольно больших) с определенной скоростью (кадров в секунду) через локальную сеть. Несложно выполнить простые вычисления необходимой пропускной способности для просмотра видеофильма размером 480x192 при скорости передачи хотя бы 20 кадров в секунду. Размер одного кадра при глубине цвета 16 бит (HiColor) равен 480 x 192 x 2 = 184320 байт. Следовательно, пропускная способность сети должна быть не ниже чем 184320 x 20 = 3686400 байт/с. Для сравнения: пропускная способность сетевой карты в 10 Мбит/с равна, 10000000 / 8 = 1250000 байт/с, то есть в три раза меньше, и не следует забывать также о параллельной передачи звука, который также будет претендовать на свой сетевой канал.

Для просмотра фильма вам понадобится сетевая карта как минимум с пропускной способностью в 100 Мбит/с (12,5 Мбайт/с). Для сетевых адаптеров

со скоростью в 10 Мбит/с просмотр фильма превратится в настоящий кошмар. Чтобы в этом убедиться, достаточно один раз попробовать. В качестве тестового примера я взял X-терминал с 10 Мбит/с сетевой картой и попытался запустить просмотр фильма (файл Shmatrixp1.AVI) командой (вывод команды приведен со значительными сокращениями):

```
$ mplayer Shmatrixp1.AVI
Using GNU internationalization
Original domain: messages
Original dirname: /usr/share/locale
Current domain: mplayer
Current dirname: /usr/share/locale

MPlayer 0.90rc5-3.2.2 (C) 2000-2003 Arpad Gereoffy (see DOCS)

CPU: Intel  (Family: 8, Stepping: 9)
Detected cache-line size is 64 bytes
CPUflags:  MMX: 1 MMX2: 1 3DNow: 0 3DNow2: 0 SSE: 1 SSE2: 1
...
Playing Shmatrixp1.AVI
AVI file format detected.
VIDEO:  [DX50]  480x192  24bpp  23,98 fps  667,6 kbps (81,5 kbyte/s)
==========================================================
=================
Opening audio decoder: [mp3lib] MPEG layer-2, layer-3
MP3lib: init layer2&3 finished, tables done
AUDIO: 32000 Hz, 2 ch, 16 bit (0x10), ratio: 10000->128000 (80,0 kbit)
Selected audio codec: [mp3] afm:mp3lib (mp3lib MPEG layer-2, layer-3)
==========================================================
=================
vo: X11 running at 1024x768 with depth 16 and 16 bpp ("192.168.1.14:0.0" =>
remote display)
```

...

Starting playback...

This file was encoded with DivX503 Build696

...

A: 6,0 V: 2,1 A-V: 3,910 ct: 0,213 52/ 52 17% 527% 3,6% 50 0 0%

****

**** Your system is too SLOW to play this! ****

****

...

Exiting... (Quit)

После нескольких секунд просмотра этого “слайд-шоу” на консоле появилось сообщение, говорящее о том, что данный компьютер для просмотра видеофильмов слишком медленный. Там же было приведено несколько вариантов для улучшения ситуации, например, использовать опцию -framedrop для пропуска кадров, если скорость вывода недостаточна. Также можно принудительно указать программе mplayer использовать вывод звука серез сервер ESounD (опция -ao esd). Новая команда просмотра фильма выглядит вот так:

$ mplayer -ao esd -framedrop Shmatrixp1.AVI

После применения этих параметров изображение стало двигаться немного быстрее, но все равно не достаточно. Да и звук жутко искажался, так как скорее всего не все фрагменты звукового потока доходили до X-терминала. Приговор окончательный, локальной сети в 10 Мбит/с для передачи видео большого разрешения недостаточно.

Для сети в 100 Мбит/с ситуация обстоит более или менее нормально. Отказываться от использования параметра -framedrop не стоит, так как следует помнить о многопользовательской природе работы на X-терминале, а следовательно пики нагрузки на сервер могут привести к заметным “заиканиям” воспроизведения видео. Когда вы смотрите видеоролик на X-

терминале, при переходе в полноэкранный режим не происходит масштабирования изображения. Это связано с тем, что вывод на удаленный дисплей не позволяет использовать аппаратные возможности видеокарты для выполнения масштабирования картинки. Поэтому фильмы остаются в оригинальных (исходных) размерах. Конечно, всегда можно использовать и программное масштабирование (опция mplayer -zoom), но при этом скорость отображения падает, а нагрузка на центральный процессор возрастает:

$ mplayer -ao esd -framedrop -zoom Shmatrixp1.AVI

Из этого раздела вы наверное уже догадались, что использование X-терминалов для просмотра фильмов, не самая лучшая идея. Передача по сети огромного объема информации не только приведет к ее чрезмерной загрузке, но и значительно увеличит нагрузку на центральный процессор сервера X-терминалов, а как результат общее падение производительности всех пользователей без исключения. В общем, смотреть фильмы на X-терминале можно, но категорически не рекомендуется.

Использование сканера

Несмотря на то, что существующая поддержка сканеров в операционной системе Linux оставляет желать лучшего, это не повод, чтобы отказаться от их использования на бездисковых X-терминалах. Для работы со сканером в операционной системе Linux существует проект SANE (Scanner Access Now Easy – "использование сканера теперь проще"), официальный сайт которого http://www.sane-project.org/.

Для того чтобы заставить работать сканер с X-терминалом в рамках проекта LTSP существует специальный пакет: ltsp_sane. Для LTSP версии 3.x можно загрузить файл ltsp_sane-1.0.3.tar.gz, который содержит в себе все необходимые компоненты для локальной работы сканера. Поддержка сканера в проекте LTSP требует частичной функциональности механизма запуска локальных приложений, поэтому перед установкой пакета ltsp_sane следует установить пакет ltsp_local_apps (нужен демон xinetd):

rpm -ihv ltsp_local_apps-3.0.0-0.i386.rpm

Теперь можно приступать к установке пакета ltsp_sane. Для начала извлечем содержимое этого пакета командой:

```
$ tar xvzf ltsp_sane-1.0.3.tar.gz
ltsp_sane-1.0.3/
ltsp_sane-1.0.3/i386/
ltsp_sane-1.0.3/i386/usr/
ltsp_sane-1.0.3/i386/usr/bin/
ltsp_sane-1.0.3/i386/usr/bin/cjpeg
ltsp_sane-1.0.3/i386/usr/bin/djpeg
ltsp_sane-1.0.3/i386/usr/bin/jpegtran
ltsp_sane-1.0.3/i386/usr/bin/rdjpgcom
ltsp_sane-1.0.3/i386/usr/bin/wrjpgcom
ltsp_sane-1.0.3/i386/usr/bin/sane-config
ltsp_sane-1.0.3/i386/usr/bin/scanimage
ltsp_sane-1.0.3/i386/usr/lib/
...
```

Теперь сделаем текущим каталог ltsp_sane-1.0.3:

```
$ cd ltsp_sane-1.0.3
```

И запустим установочный сценарий (обязательно от имени суперпользователя):

```
# ./install
LTSP Sane Scanner access by Robert Stanford (rob@rotapile.com)
Bash Version=2.05b.0(1)-release

Installing Base package.
Installation *Complete*

Have Fun!!!
```

Если вы увидели на экране похожее сообщение, значит установка пакета ltsp_sane прошла успешно. На этом будем считать первый этап настройки сканера на X-терминале успешно пройденным.

Прежде чем переходить ко второму этапу, следует обратить внимание на имеющийся в вашем распоряжении сканер, а также на существующую его поддержку со стороны операционной системы Linux. Я настоятельно рекомендую, прежде чем приступать к попыткам настроить сканер на работу с X-терминалом, сначала попробовать поработать с ним на обычном персональном компьютере с Linux. В противном случае я вам не завидую, так как существует довольно большой шанс, что вы будете пытаться настроить устройство, которое в Linux вообще не работает.

Для активизации сканера на X-терминале необходимо внести изменения в раздел файла /opt/ltsp/i386/etc/lts.conf, который отвечает за X-терминал с подключенным сканером. Если поддержка запуска локальных приложений отсутствует, то добавить нужно такую строчку:

RCFILE_01 = xinetd

Как уже было отмечено раннее, очень важно знать какой сканер настраивать. Так, в моем случае, мне в руки попал сканер BearPaw 1200TA производства компании Mustek (сайт http://www.mustek.com/). Эта модель сканера имеет USB-интерфейс, а значит для ее работы необходимо загрузить еще дополнительные модули ядра. Изменения в файле /opt/ltsp/i386/etc/lts.conf будут такими:

MODULE_01 = uhci

MODULE_02 = scanner

Иногда, после загрузки X-терминала, не появляется символьное устройство /dev/usb/scanner0. Чтобы решить эту проблему, запустите инициализирующий скрипт usbscanner для выбранного X-терминала:

RCFILE_02 = usbscanner

Если вы все сделали правильно, то после перезагрузки X-терминала в текстовом режиме, на экране вы должны увидеть примерно такие строки:

...

Building the XF86Config-4 file

hub.c: new USB device 00:1d.1-2, assigned adress2

scanner.c: USB scanner device (0x55f/0x021e) now attached to scanner0

Building the start_ws script

...

Проверим как сканер был воспринят системой SANE. В ответ на команду:

scanimage -L

На экране X-терминала вы должны будете увидеть ссылку на найденный сканер.

Теперь нужно описать на сервере X-терминалов в файле /etc/sane.d/net.conf, доменное имя X-терминала, к которому подключен сканер. Если вам повезет, то уже прямо сейчас можно будет сканировать на бездисковом X-терминале. Доля везения в деле подключения сканера к X-терминалу безусловно необходима. На практике все может оказаться гораздо печальнее. Так, вышеупомянутый сканер Mustek BearPaw 1200TA использует драйвер gt68xx пакета SANE и для его работы требуется специальный файл a1fw.usb, который нужно поместить в каталог /usr/share/sane/gt68xx/. Иными словами, подводных камней в использовании сканера на X-терминале достаточно много, и только от вашего желания и настойчивости зависит успех в этом нелегком деле.

Доступ пользователей к флоппи дисководам

Лично я противник установки флоппи-дисководов на X-терминалах. Во-первых, они пережиток прошлого, своего рода “компьютерный атавизм”. Во-вторых, дискеты постоянно теряют информацию, так как очень часто выходят из строя, и вообще являются крайне ненадежным способом хранения информации. В-третьих, размер дискеты 1,44 Мбайт чрезвычайно мал по сегодняшним меркам (ни тебе песню записать, ни программу, ни фильм). К тому же для передачи файлов есть ftp, web, электронная почта, и даже USB-flash накопители наконец. В общем, слишком много минусом, и совсем мало плюсов.

Но как оказалось на практике, у пользователей свое мнение на этот счет. Нравятся им дискеты и все тут. У кого дома интернета нет, у кого USB-порта для Flash-накопителя, а кто и просто привык. В общем, без флоппи-дисководов им X-терминалы не понравились.

Для поддержки флоппи-дисководов на X-терминалах предусмотрен специальный пакет ltsp_floppyd. Установка его на сервер особых проблем не вызывает:

rpm -ihv ltsp_floppyd-3.0.0-0.i386.rpm

Фактически, в папку /opt/ltsp/i386 устанавливаются два файла: бинарный демон floppyd, и одноименный сценарий запуска этого демона. Работает поддержка флоппи дисковода не совсем обычно, как это принято в Linux (примонтировал, размонтировал), а через пакет mtools (так что, если в вашей системе он не установлен, то сейчас самое время сделать это). Более того, для доступа к дисководу, кроме команд пакета mtools, нужен еще и специальный файловый менеджер, поддерживающий работу через mtools. Правда у данного метода есть и небольшие ограничения:

X-терминал должен работать в графическом режиме (RUNLEVEL = 5);

Не должен использоваться локальный запуск приложений (LOCAL_APPS = N).

Но обо всем по порядку. Первым делом проверяем, установлен ли пакет mtools:

$ rpm -qi mtools

Name : mtools Relocations: (not relocateable)

Version : 3.9.8 Vendor: ASPLinux

Release : 7 Build Date: Втр 29 Кві 2003 09:31:08

Install Date: Срд 25 Лют 2004 14:16:48 Build Host: arena.asplinux.ru

Group : Applications/System Source RPM: mtools-3.9.8-7.src.rpm

Size : 325481 License: GPL

Signature : DSA/SHA1, Пнд 05 Тра 2003 10:20:01, Key ID c96aa8e69b3c94f4

Packager : ASPLinux Team

URL : http://mtools.linux.lu/

Summary : Programs for accessing MS-DOS disks without mounting the disks.

Description :

Mtools is a collection of utilities for accessing MS-DOS files.

Mtools allow you to read, write and move around MS-DOS filesystem files (normally on MS-DOS floppy disks). Mtools supports Windows95 style long file names, OS/2 XDF disks, and 2m disks.

Mtools should be installed if you need to use MS-DOS disks.

Если получили похожий ответ, то значит пакет mtools уже установлен. Если же на экране написано “package mtools is not installed”, то установите его командой (путь к пакету mtools и его полное название могут отличаться в вашей системе):

rpm -ihv mtools-3.9.8-7.i386.rpm

Теперь в соответствующем разделе файла /opt/ltsp/i386/etc/lts.conf для X-терминала с флоппи-дисководом (например, pashax) добавим такой пункт:

```
[pashax]
...
    RCFILE_01 = floppyd
```

Тем самым мы объяснили системе LTSP, что при запуске X-терминала нужно запускать сценарий floppyd.

Теперь для каждого пользователя, который нуждается в работе с флоппи-дисководом на X-терминалах, в его домашнем каталоге создаем файл .mtoolsrc с таким содержимым:

```
drive a: file="$DISPLAY" remote 1.44m mformat_only
```

Чтобы упростить себе жизнь в будущем, при добавлении новых пользователей можно скопировать этот файл в директорию /etc/skel.

Перегружаем X-терминал и от имени пользователя выполняем команду:

```
$ mdir a:
 Volume in drive A has no label
 Volume Serial Number is 3030-5761
Directory for A:/

AUTOEXEC BAT      1100 05-05-1999  22:22
```

CONFIG SYS 811 01-16-2004 10:57

...

DISPLAY SYS 17239 05-05-1999 22:22

COUNTRY SYS 30742 05-05-1999 22:22

EGA3 CPI 58753 05-05-1999 22:22

MODE COM 29911 05-05-1999 22:22

KEYBRD3 SYS 31633 05-05-1999 22:22

KEYB COM 20135 05-05-1999 22:22

FIX_CIH EXE 21430 05-05-2004 12:34

28 files 1 201 167 bytes

26 624 bytes free

Конечно, работать с дискетой в командной строке неудобно, поэтому воспользуемся файловым менеджером, который поддерживает работу с дискетой через mtools. Довольно успешно с этой задачей справляется файловый менеджер проекта KDE – konqueror. Запускаем его и в поле адреса вводим: floppy:/a . После этого вы должны увидеть в окне файлового менеджера содержимое дискеты, которую, я надеюсь, вы предварительно вставили в флоппи-дисковод X-терминала (см. рис. 9.12):

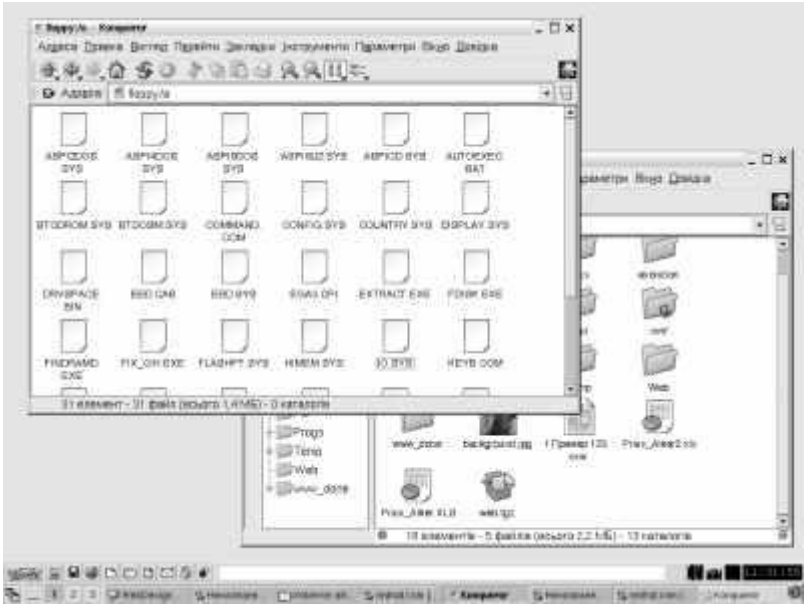

Рис. 9.12. Окно файлового менеджера Konqueror показывает содержимое дискеты X-терминала через пакет mtools

На первый взгляд все работает нормально, очень радует то, что не нужно выполнять монтирования. Правда, есть и небольшие исключения, которые

омрачают радужную картину поддержки дискет в X-терминалах. Во-первых, это отсутствие поддержки в именах файлов символов кириллицы. Более того, если создадите на дискете такой файл, то он будете отображаться абракадаброй. Конечно, все опять упирается в двойные стандарты кодовых страниц для имен файлов и системной кодовой страницы. Если в случае с традиционной дискетой достаточно было указать в параметрах монтирования файла /etc/fstab значение codepage=866, то для пакета mtools такой трюк не проходит, так как для этого пакета не существует аналогичной опции. В общем, от использования файлов, названных буквами кириллицы, на X-терминалах пока придется отказаться.

Не всегда использование файлового менеджера Konqueror оправдано. Как уже было сказано выше, приложения KDE слишком ресурсоемки и изобилуют графическими эффектами, которые увеличивают нагрузку на локальную сеть и сервер. Если вы ищите простой файловый менеджер, который умеет только переписывать файлы с дискеты и на дискету через пакет mtools, то вам должен понравиться MToolsFM, найти который можно на сайте http://mtools.linux.lu.

Мне в руки попал пакет MToolsFM-1.9-3.i386.rpm, установка которого прошла без проблем:

rpm -ihv MToolsFM-1.9-3.i386.rpm

Внешний вид этого файлового менеджера представлен на рисунке 9.13. К его несомненным преимуществам следует отнести также скромный размер в памяти (VmSize менее 8 Мбайт) и просто моментальную скорость запуска. К тому же, он умеет: копировать файлы с дискеты на жесткий диск и обратно, удалять файлы, переименовывать их и даже создавать новые каталоги. Так же MToolsFM имеет довольно примитивную поддержу операций по работе с архивными файлами и поддержку печати на принтере. Работать с файловым менеджером MToolsFM очень просто, простым выделением файлов при помощи “мышки” можно перемещать их как с дискеты так и на дискету.

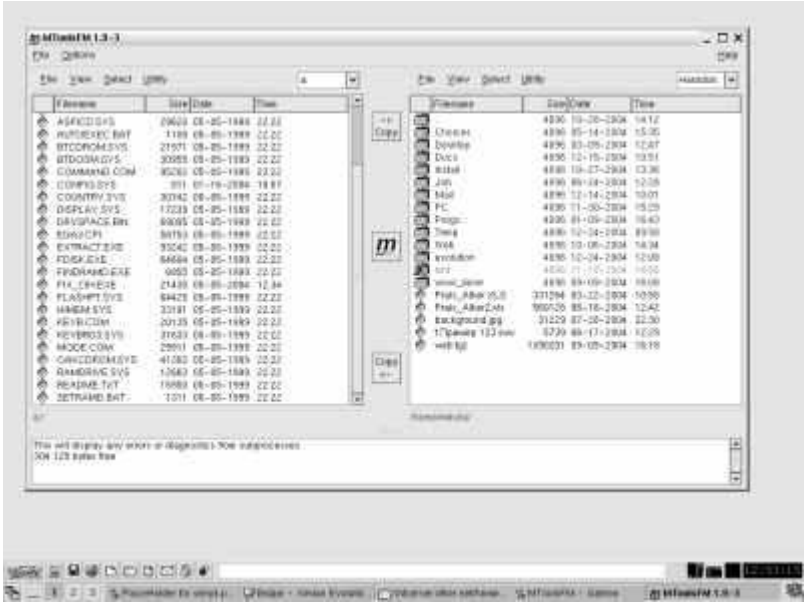

Рис. 9.13. Окно файлового менеджера MToolsFM, позволяющего работать с дискетами пользователю X-терминала через пакет mtools

В заключении рассказа о поддержке флоппи-дисководов на X-терминалах необходимо сказать, что проблемы букв кириллицы присутствуют и в MToolsFM, правда теперь, все “незнакомые” буквы заменяются символами подчеркивания. Большим неудобством является так же то, что при ошибках чтения с дискеты локальный демон floppyd впадает в жуткую “депрессию” и на все запросы пользователя отвечает:

floppyd_io: Input/output error

Permission denied, authentication failed!

Auth failed: Device locked!Drive 'A:' not supported

Cannot initialize 'A:'

Вывести его из этого "ступора" помогает только принудительная перезагрузка X-терминала, что, согласитесь, хоть и не сложно, но все же не очень удобно. Следует также отметить еще один способ обеспечения доступа к локальному флоппи-дисководу X-терминала. Суть его заключается в том, чтобы монтировать его через локальную сеть и для этих целей в рамках проекта LTSP создан специальный демон автоматического монтирования ltsp_autofs. Обязательным условием использования данного пакета является поддержка запуска локальных приложений, но зато в результате вы получаете не только доступ к флоппи-дисководу, а ещё и возможность работы с устройством чтения компакт-дисков. Более детальную информацию ищите в документации к

проекту LTSP. В любом случае пример использования флоппи-дисководов на X-терминалах показывает гибкость настройки и большой потенциал применения этих тонких клиентов, он как бы доказывает, что для X-терминалов нет ничего не возможного.

Локальный запуск приложений

До этого времени все приложения, которые мы запускали на X-терминалах, выполнялись на сервере. Клиент X-терминал только отображал окошки на экране, то есть представлял из себя классический пример терминала в X Window System.Однако, существуют случаи, когда от использования X-терминала может потребоваться большая функциональность. Одним из таких как раз и является локальный запуск приложений на X-терминале. Под локальным запуском будем считать выполнение программы непосредственно в операционной среде X-терминала, а не на сервере, как это было раньше.

Какие же приложения целесообразно запускать локально и в каком случае? Преимуществ локального запуска приложений довольно много. Во-первых, это уменьшает нагрузку на центральный сервер, а следовательно положительно сказывается на снижении ресурсоемкости эксплуатации X-терминальной сети. Во-вторых, работа программы, запущенной локально на терминале, никак не сказывается на работе других процессов и пользователей. В-третьих, программы, использующие звуковые эффекты в своей работе, значительно проще настраивать при локальном запуске, нежели при использовании передачи звука через локальную сеть. В-четвертых, программы, интенсивно использующие графику, также намного лучше работают при локальном запуске.

Несмотря на ряд неоспоримых преимуществ, локальный запуск приложений целесообразен только при достаточных вычислительных ресурсах X-терминала. Особо критичны показатели объема оперативной памяти и мощности процессора. Так, экспериментировать с локальным запуском приложений можно, только после того, как вы убедитесь, что X-терминал имеет не менее 64 Мбайт оперативной памяти. Чтобы сразу не разочароваться в достоинствах локального запуска, рекомендуется начать работу на X-

терминале, который имеет центральный процессор уровня Intel Pentium II или выше. Другими словами, “разгружать” центральный X-терминал сервер целесообразно только теми X-терминалами, которые достаточно мощные для этого. Кроме этого, при локальном запуске приложений скорость их работы может увеличиться, но время запуска всегда будет ниже, чем при работе на сервере. Это связано с двумя факторами: работой через сетевую файловую системы NFS, которая снижает быстродействие, а также с невозможностью использования разделяемых библиотек сервера, которые уже загружены в его оперативную память.

Для того, чтобы иметь возможность запускать приложения локально, X-терминал должен знать такую информациею:

Идентификационный номер пользователя (User ID);

Номер первичной группы, к которой принадлежит пользователь (Primary GID);

Домашний каталог пользователя (как правило, /home/<имя пользователя>).

Для получения подобной информации, в проекте LTSP версии 3.0 используется NIS (Network Information Service - сетевая информационная служба) или, как ее называли раньше, Yellow Pages (Желтые страницы). Компания Sun, разработавшая данную технология, вынуждена была изменить название Yellow Pages на NIS после судебного процесса, так как уже существовала одноименная торговая марка, однако до сих пор все команды сетевой информационной службы начинаются на yp.

Итак, для того чтобы начать использование локальных приложений на X-терминалах, необходимо дополнительно сконфигурировать некоторые службы сервера, в частности NIS и NFS, а в разделе файла /opt/ltsp/i386/etc/lts.conf, посвященному выбранному X-терминалу, добавить параметр "LOCAL_APPS = Y".

Настройка NFS уже не должна представлять для вас проблему, так как достаточно раскомментировать строку экспорта домашнего каталога /home и перезапустить демоны сетевой файловой системы:

/sbin/service nfs restart

Shutting down NFS mountd: [OK]

Shutting down NFS daemon: [OK]

Shutting down NFS quotas: [OK]

Shutting down NFS services: [OK]

Starting NFS services: [OK]

Starting NFS quotas: [OK]

Starting NFS daemon: [OK]

Starting NFS mountd: [OK]

Для правильного функционирования локального запуска приложений нам понадобятся дополнительные программы и разделяемые библиотеки, поэтому на сервере нужно установить пакет ltsp_local_apps. Делаем это такой командой:

rpm -ihv ltsp_local_apps-3.0.0-0.i386.rpm

При использовании параметра запуска локальных приложений на X-терминале после выполнения загрузки ядра операционной системы происходит следующее:

Домашний каталог /home монтируется по сети через NFS;

На X-терминале создается файл /var/yp/nicknames;

Запускается служба portmapper на X-терминале;

Также на X-терминале стартует демон xinetd;

Создается файл /etc/yp.conf в среде X-терминала;

Запускается команда domainname с параметром NIS_DOMAIN, который был получен из содержимого файла /opt/ltsp/i386/etc/lts.conf;

На X-терминале стартует ypbind.

Работа службы NIS довольно специфична, а поэтому для ее правильной работы необходимо, чтобы все узлы локальной сети обращались к одному NIS-серверу и были ассоциированы с одним NIS-доменом. Правда, к домену DNS NIS-домен никакого отношения не имеет, - это два совершенно разные понятия.

NIS_DOMAIN – указывает на домен NIS (не путать с DNS-доменом);

NIS_SERVER – указывает адрес сервера, на котором работает служба NIS, позволяет использовать локальный запуск приложений даже тогда, когда служба NIS работает на сервере, отличном от сервера X-терминалов.

Система NIS работает в архитектуре клиент-сервер. Так, на сервере запущен сервис-демон ypserv, а на клиенте (в нашем случае X-терминале) программа ypbind. Когда клиентскому компьютеру нужна информация по проверке аутентификации пользователя или получения его домашнего каталога, то ypbind обращается к ypserv посредством специального протокола. Для того, чтобы эта система работала, необходимо настроить службу NIS на сервере.

Если вы уже используете NIS, тогда все в порядке, в противном случае приступаем к его настройке. В первую очередь, проверяем установлены ли пакеты NIS:

```
$ rpm -qa | grep ypserv
```

При получении неутешительного ответа устанавливаем его на сервер такой командой:

```
# rpm -ihv ypserv-2.6-2.i386.rpm
Preparing...                ##########################################
[100%]
   1:ypserv                 ##########################################
[100%]
```

Главным конфигурационным файлом сетевой информационной службы является файл /etc/ypserv.conf, содержимое которого представлено ниже:

```
#
# ypserv.conf In this file you can set certain options for the NIS server,
#       and you can deny or restrict access to certain maps based
#       on the originating host.
#
#       See ypserv.conf(5) for a description of the syntax.
#

# Some options for ypserv. This things are all not needed, if
# you have a Linux net.

# Should we do DNS lookups for hosts not found in the hosts table ?
```

```
# This option is ignored in the moment.
dns: no

# How many map file handles should be cached ?
files: 30

# xfr requests are only allowed from ports < 1024
xfr_check_port: yes

# The following, when uncommented,  will give you shadow like passwords.
# Note that it will not work if you have slave NIS servers in your
# network that do not run the same server as you.

# Host                  : Domain  : Map            : Security
#
# *                     : *       : passwd.byname    : port
# *                     : *       : passwd.byuid     : port

# Not everybody should see the shadow passwords, not secure, since
# under MSDOG everbody is root and can access ports < 1024 !!!
*                       : *       : shadow.byname    : port
*                       : *       : passwd.adjunct.byname : port

# If you comment out the next rule, ypserv and rpc.ypxfrd will
# look for YP_SECURE and YP_AUTHDES in the maps. This will make
# the security check a little bit slower, but you only have to
# change the keys on the master server, not the configuration files
# on each NIS server.
# If you have maps with YP_SECURE or YP_AUTHDES, you should create
# a rule for them above, that's much faster.
# *                     : *       : *              : none
```

Если в вашей сети не будет использоваться специфическая настройка NIS, то давайте оставим в нем все как есть.

Теперь следует установить имя NIS-домена для нашего сервера. Сделать это можно командой:

domainname <имя домена NIS>

Или, чтобы доменное имя было всегда доступно после загрузки сервера добавить в файл /etc/sysconfig/network такую строчку:

NISDOMAIN="<имя домена NIS>"

Для того, чтобы предоставить некоторые файлы для общего доступа, используется файл /var/yp/Makefile, который и определяет, что экспортировать, а что нет. Приводить все содержимое этого файла в данной книге нет необходимости, но вы должны обратить внимание на такие его строки:

```
# If you don't want some of these maps built, feel free to comment
# them out from this list.

all:  passwd group hosts rpc services netid protocols mail \
  # netgrp shadow publickey networks ethers bootparams printcap \
  # amd.home auto.master auto.home auto.local passwd.adjunct \
  # timezone locale netmasks
```

Они означают какие сервисы будут доступны удаленным NIS-клиентам от вашего сервера. В данном примере будут экспортироваться для совместного использования: passwd group hosts rpc services netid protocols mail. Если вы считаете, что некоторые из них вам не нужны, то просто закомментируйте их.

Настало время запустить сервер NIS. Делаем это командой:

```
# /sbin/service ypserv start
Starting YP server services:                    [    OK    ]
```

Если запуск сервера NIS прошел успешно, самое время удостовериться, что он будет запускаться автоматически при включении сервера. Для этого достаточно выполнить команду:

```
$ /sbin/chkconfig --list  ypserv
ypserv          0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

Как видите, в данном примере сервер NIS не настроен на автоматический старт. Исправить положение поможет команда:

/sbin/chkconfig ypserv --level 5 on

После успешного выполнения запуска демона ypserv, необходимо инициализировать первичный NIS-сервер. Для этого используется такая команда:

/usr/lib/yp/ypinit -m

Далее на запрос программы следует перечислить имена хостов, на которых будут работать NIS-серверы. Закончив ввод этого списка, необходимо нажать комбинацию клавиш Ctrl+D:

At this point, we have to construct a list of the hosts which will run NIS servers. admin.alker.net is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type a .

next host to add: admin.alker.net

next host to add:

А также подтвердить правильность указания списка узлов:

The current list of NIS servers looks like this:

admin.alker.net

Is this correct? [y/n: y] y

Через несколько секунд формирование базы данных NIS будет окончено, о чем свидетельствуют такие строки на экране:

We need a few minutes to build the databases...

Building /var/yp/alker.trast.net/ypservers...

Running /var/yp/Makefile...

gmake[1]: Entering directory `/var/yp/alker.trast.net'

Updating passwd.byname...

Updating passwd.byuid...

Updating group.byname...

Updating group.bygid...

Updating hosts.byname...

Updating hosts.byaddr...

Updating rpc.byname...

Updating rpc.bynumber...

Updating services.byname...

Updating services.byservicename...

Updating netid.byname...

Updating protocols.bynumber...

Updating protocols.byname...

gmake[1]: Leaving directory `/var/yp/alker.trast.net'

admin.alker.net has been set up as a NIS master server.

Now you can run ypinit -s admin.alker.net on all slave server.

Если в процессе выполнения команды ypinit произошли ошибки, то это указывает либо на неточности в файле /var/yp/Makefile, либо на неправильную настройку сервисов, которые экспортируются. После исправления допущенных ошибок, необходимо повторно произвести запуск утилиты ypinit.

Когда настройка сервера NIS успешно закончена, можно непосредственно приступать к запуску локальных приложений на терминале. Но перед этим, необходимо установить все их компоненты в дерево каталогов /opt/ltsp/i386/. На официальном сайте проекта LTSP (http://www.ltsp.org) можно найти специально собранный пакет с браузером Netscape Navigator (ltsp_local_netscape-3.0.0-0.i386.rpm) для локального запуска.

Если вам нужно запустить локально совсем другую программу, необходимо скопировать принадлежащие ей файлы в директорию /opt/ltsp/i386. Кроме этого, вашей программе могут потребоваться дополнительные разделяемые библиотеки, поэтому необходимо найти и также скопировать их в этот каталог. Понять вышесказанное лучше на конкретном примере. Предположим, что нам захотелось локально запустить медиа-плеер xmms.

Перед определением файлов, из которых состоит программный продукт, важно помнить, что они должны быть скомпилированы для архитектуры X-терминала. Например, если на сервере установлен пакет с архитектурой i686, то он не сможет запуститься на X-терминалах собранных на базе Intel Pentium и Intel 486. Поэтому целесообразно запускать локально, только пакеты для архитектуры i386.

Вернемся к xmms. Для того, чтобы узнать, какие файлы входят в его состав, выполним команду с такими параметрами (вывод приведен с сокращениями):

```
$ rpm -q --filesbypkg xmms
xmms            /usr/bin/wmxmms
xmms            /usr/bin/xmms
xmms            /usr/lib/libxmms.so.1
xmms            /usr/lib/libxmms.so.1.2.1
xmms            /usr/lib/xmms
xmms            /usr/lib/xmms/Effect
xmms            /usr/lib/xmms/Effect/libecho.so
xmms            /usr/lib/xmms/Effect/libstereo.so
xmms            /usr/lib/xmms/Effect/libvoice.so
xmms            /usr/lib/xmms/General
xmms            /usr/lib/xmms/General/libir.so
xmms            /usr/lib/xmms/General/libjoy.so
xmms            /usr/lib/xmms/General/libsong_change.so
xmms            /usr/lib/xmms/Input
xmms            /usr/lib/xmms/Input/libcdaudio.so
xmms            /usr/lib/xmms/Input/libidcin.so
xmms            /usr/lib/xmms/Input/libmikmod.so
xmms            /usr/lib/xmms/Input/libmpg123.so
xmms            /usr/lib/xmms/Input/libtonegen.so
xmms            /usr/lib/xmms/Input/libvorbis.so
xmms            /usr/lib/xmms/Input/libwav.so
```

xmms /usr/lib/xmms/Output
xmms /usr/lib/xmms/Output/libOSS.so
xmms /usr/lib/xmms/Output/libartsout.a
xmms /usr/lib/xmms/Output/libartsout.so
xmms /usr/lib/xmms/Output/libartsout.so.0
xmms /usr/lib/xmms/Output/libartsout.so.0.0.0
xmms /usr/lib/xmms/Output/libdisk_writer.so
xmms /usr/lib/xmms/Output/libesdout.so
xmms /usr/lib/xmms/Visualization
xmms /usr/lib/xmms/Visualization/libbscope.so
xmms /usr/lib/xmms/Visualization/libogl_spectrum.so
xmms /usr/lib/xmms/Visualization/libsanalyzer.so
xmms /usr/share/applications/redhat-audio-player.desktop
xmms /usr/share/doc/xmms-1.2.7
xmms /usr/share/doc/xmms-1.2.7/AUTHORS
xmms /usr/share/doc/xmms-1.2.7/COPYING
xmms /usr/share/doc/xmms-1.2.7/ChangeLog
xmms /usr/share/doc/xmms-1.2.7/FAQ
xmms /usr/share/doc/xmms-1.2.7/INSTALL
xmms /usr/share/doc/xmms-1.2.7/NEWS
xmms /usr/share/doc/xmms-1.2.7/README
xmms /usr/share/doc/xmms-1.2.7/README.ESD
xmms /usr/share/doc/xmms-1.2.7/TODO
xmms /usr/share/locale/af/LC_MESSAGES/xmms.mo
...
xmms /usr/share/locale/wa/LC_MESSAGES/xmms.mo
xmms
/usr/share/locale/zh_CN.GB2312/LC_MESSAGES/xmms.mo
xmms /usr/share/locale/zh_TW/LC_MESSAGES/xmms.mo
xmms /usr/share/man/man1/wmxmms.1.gz
xmms /usr/share/man/man1/xmms.1.gz

```
xmms            /usr/share/pixmaps/mini/xmms_mini.xpm
xmms            /usr/share/pixmaps/xmms.xpm
xmms            /usr/share/pixmaps/xmms_logo.xpm
xmms            /usr/share/xmms
xmms            /usr/share/xmms/wmxmms.xpm
```

Если рассмотреть этот список повнимательней, то можно заметить, что скопировать нужно только файлы:

```
/usr/bin/wmxmms
/usr/bin/xmms
/usr/lib/libxmms.so.1
/usr/lib/libxmms.so.1.2.1
/usr/share/pixmaps/mini/xmms_mini.xpm
/usr/share/pixmaps/xmms.xpm
/usr/share/xmms/wmxmms.xpm
```

А также каталог /usr/lib/xmms. Не следует забывать и про файл локализации данного медиа-проигрывателя. Лично для себя я скопировал файл /usr/share/locale/uk/LC_MESSAGES/xmms.mo.

Теперь настала очередь определить, какими разделяемыми библиотеками пользуется xmms. В этом деле нам поможет такая команда:

```
$ ldd /usr/bin/xmms
    libdl.so.2 => /lib/libdl.so.2 (0x40027000)
    libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4002b000)
    libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40039000)
    libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x40043000)
    libxmms.so.1 => /usr/lib/libxmms.so.1 (0x4005a000)
    libgtk-1.2.so.0 => /usr/lib/libgtk-1.2.so.0 (0x40067000)
    libgdk-1.2.so.0 => /usr/lib/libgdk-1.2.so.0 (0x401af000)
    libgmodule-1.2.so.0 => /usr/lib/libgmodule-1.2.so.0 (0x401e7000)
    libgthread-1.2.so.0 => /usr/lib/libgthread-1.2.so.0 (0x401ea000)
    libglib-1.2.so.0 => /usr/lib/libglib-1.2.so.0 (0x401ee000)
    libXi.so.6 => /usr/X11R6/lib/libXi.so.6 (0x40213000)
```

libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x4021b000)

libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40229000)

libm.so.6 => /lib/tls/libm.so.6 (0x40308000)

libc.so.6 => /lib/tls/libc.so.6 (0x42000000)

/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

Как видно из ответа команды ldd, необходимо проверить дополнительно наличие 15 файлов (файл libxmms.so.1 был нами уже успешно скопирован).

После копирования всех необходимых файлов для локального запуска, на последней стадии подготовки нужно создать специальный сценарий запуска локального приложения. Необходимость в написании такого сценария вызвана тем, что пользователь X-терминала работает в среде сервера, который ничего не знает о локальной среде X-терминала. Поэтому, в сценарии для удаленного запуска используется команда rsh. Например, для локального запуска xmms необходимо написать сценарий следующего содержания:

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} /usr/bin/xmms -display ${DISPLAY}
```

В первой строке сценария запуска локального приложения мы получаем адрес X-терминала, в во второй строке используем команду rsh для непосредственно удаленного запуска.

Некоторые программы требуют перед запуском установки определенных значений переменных окружения. Например, для локального запуска браузера Netscape Navigator применяется такой сценарий:

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} MOZILLA_HOME=/usr/local/netscape \
    /usr/local/netscape/netscape -display ${DISPLAY}
```

Конечно, для того, чтобы воспользоваться командой rsh, необходимо установить на сервере содержащий ее пакет. Сделать это можно при помощи такой команды:

```
# rpm -ihv /mnt/cdrom/ASPLinux/RPMS/rsh-0.17-14.i386.rpm
Preparing...                ##########################################
[100%]
```

1:rsh ##
[100%]

Вообще, локальный запуск приложений – это довольно интересная возможность более полно использовать ресурсы X-терминалов. Но пользователей также следует предостеречь от чрезмерного увлечения такими приложениями. Причин этому довольно много. Во-первых, у X-терминала довольно слабый контроль над операционной системой, которая на нем запущена. Все команды выполняются в среде сервера, а следовательно, если с выполнением локального приложения будет что-то не так, то пользователь вряд ли сможет хоть чем-то помочь. Во-вторых, не все программы удастся запустить локально, а некоторые потребуют от вас выполнить слишком много дополнительных настроек. В-третьих, при построении локальной сети X-терминалов, рационально исходить из предположения, что их аппаратные ресурсы ограничены, а следовательно далеко не все их них подходят для запуска локальных приложений. В-четвертых, при использовании локального запуска приложений на ограниченном числе X-терминалов, мы тем самым теряем их универсальность, то есть в сети появляются терминалы с поддержкой локальных приложений и без, что автоматически делает их неодинаковыми для пользователей. И наконец, пятой причиной, осторожного отношения к X-терминалам следует назвать сложность настройки этого сервиса и его потенциальные проблемы с безопасностью (команда rsh всегда грешила брешами в системе безопасности операционной системы Linux).

Не случайно описание метода запуска локальных приложений было приведено в самом конце этой книги. Лично я посоветовал бы пользователям X-терминальной среды оставить эти эксперименты на потом. Сначала настроить систему X-терминалов в обычном режиме, а после, когда все будет работать исправно, попробовать локальный запуск приложений. Как знать, может к тому времени надобность в нем сама отпадет.

Приложение А. Реальный пример внедрения X-терминалов

К идее использования X-терминалов на рабочем месте я пришел не сразу. Скорее всего, это был вынужденный шаг. В организации, в которой я работаю,

было принято решение легализовать используемое программное обеспечение, а чтобы сделать это при минимуме финансовых затрат, было принято решение перевести по возможности компьютеры с операционных систем Microsoft Windows на Linux. К этому моменту парк компьютерной техники (больше, конечно, он напоминал музей) был таким как показывает Таблица 1.

Таблица 1. Имеющийся парк компьютерной техники

Процессор (CPU)	Чипсет	ОЗУ	Видео	Винчестер	Дополнительноеоборудование	Операционная система	Предназначениекомпьютера, выполняемые задачи
Intel Celeron 1.8 ГГц	i845	256 Мб	AGP GeForce 2 MX/400,32 Мб	6,4 Гб	флоппи, CD-ROM, модем	Windows 98	Бухгалтерия,клиент-банк, программы для бухгалтерской и статистическойотчетности, электронная почта
Intel Pentium II 233МГц	i440 BX	256 Мб	AGP ATI Rage IIC, 4Мб	3,2 Гб	флоппи, CD-ROM	Windows 98	Бухгалтерия,программа для бухгалтерской отчетности
Intel Celeron 433 МГц	i810	128 Мб	Интегрированное	8,4 Гб	флоппи, CD-ROM, модем	Windows 98	Бухгалтерия,клиент-банк, информационно-справочная система по текущемузаконодательству
Intel Pentium MMX 166МГц	i430	64 Мб	PCI s3 trio, 2 Мб	3,2 Гб	флоппи, CD-ROM	Windows 98	Бухгалтерия,информационно-справочная система по текущему законодательству
Intel Celeron 566 МГц	i430	96 Мб	AGP ATI Rage Pro, 8Мб	10 Гб	флоппи, CD-ROM,модем, звук	Windows Me	Бухгалтерия,программа для бухгалтерской отчетности
Intel Pentium MMX 233МГц	i430	48 Мб	PCI s3 trio, 1 Мб	4,3 Гб	флоппи, CD-ROM,модем, звук	Windows 98	Транспортный отдел,офисный пакет, интернет, электронная почта, специализированныепрограммы для определения оптимального маршрута перевозки грузов
Intel Pentium II 350МГц	i440 BX	256 Мб	AG P nVidia	20 Гб	флоппи, CD-ROM,модем, звук	Windows Me	Отдел сбыта, офисныйпакет, интернет,

			Riva TNT,16 Мб				электронная почта, справочники предприятий
Intel Celeron 466 МГц	i430	128 Мб	AGP ATI Rage, 4 Мб	3,2 Гб	флоппи, CD-ROM	Windows Me	Отдел сбыта, офисныйпакет, интернет, электронная почта, программа для ведениясклад ского учета
Intel Celeron 800 МГц	i815	128 Мб	Интегрированное	20 Гб	флоппи, CD-ROM, модем	Windows Me	Отдел сбыта, офисныйпакет, интернет, электронная почта, справочники предприятий
Intel Celeron 1 ГГц	i815	256 Мб	AGP SIS 6325, 4 Мб	40 Гб	флоппи, CD-ROM, модем	Windows 98	Отдел сбыта, офисныйпакет, интернет, электронная почта, справочники предприятий
Intel Pentium MMX 200МГц	i430	64 Мб	PCI s3 trio, 1 Мб	3,2 Гб	флоппи, CD-ROM	Windows 98	Отдел сбыта, офисныйпакет, справочники предприятий
Intel Celeron 466 МГц	i430	64 Мб	AGP MGA G100, 4 Мб	3,2 Гб	флоппи, CD-ROM, модем	Windows 98	Управление, офисныйпакет, интернет, электронная почта
Intel Celeron 500 МГц	i810	256 Мб	Интегрированное	3,2 Гб	флоппи, CD-ROM, модем	ASPLinux 7.3	Системноеадминистрирование, офисный пакет, интернет, электронная почта,программы по управлению сетевой средой
Intel Pentium MMX 200МГц	i430	64 Мб	Интегрированное	3,2 Гб	флоппи	ASPLinux 7.3	Интернет-сервер,брандмауэр, прокси-сервер

Даже примитивный расчет показывает, что для приобретения лицензий на операционную систему Windows и офисный пакет Microsoft Office пришлось бы затратить около 12 * (100 + 500)= 7200 $. Для организации в которой я работаю, сумма практически фантастическая. Если ограничиться только операционной системой Windows, а в качестве офисного пакета использовать

OpenOffice, то сумма единоразовых затрат будет намного скромнее 12 * 100 = 1200 $, но при этом качество администрирования останется на том же низком уровне. Не следует также забывать и о потенциальной угрозе распространения вирусов, от которых использующиеся на предприятии операционные системы Windows 98 и Windows Me практически не защищены. Переход к использованию операционной системы Windows XP также рассматривался, но в виду ее большей стоимости и необходимости существенной модернизации компьютерной техники был отвергнут.

В качестве альтернативы операционным системам от Microsoft был выбран дистрибутив ASPLinux, благодаря тому, что он был неплохо локализирован (поддержка русского и украинского языка), а также системный администратор имел достаточный опыт работы с ним. По сравнению с продуктами от Microsoft, коробочная версия ASPLinux стоила сущие копейки, и руководство компании даже согласилось вложить дополнительные средства в модернизацию парка компьютерной техники для того, чтобы он соответствовал требованиям операционной системы Linux. Здесь не следует забывать о парадоксе человеческого материального мышления. Намного проще объяснить человеку за что он платит деньги, когда тот покупает, например, компьютерную запчасть, чем уговорить его добровольно выложить круглую сумму за компакт-диск с голографической наклейкой. Возможно, в странах с более высокой правовой культурой эти различия не так заметны, но в нашей стране идея платного использования программного обеспечения воспринимается конечными пользователями крайне враждебно. Хотя не стоит забывать, что движение за свободное распространение программ возникло именно в западном мире.

Хорошенько проанализировав ситуацию, стало ясно, что все персональные компьютеры нельзя перевести на Linux, так как часть из них использовала специализированное программное обеспечение, написанное для платформы win32 и заставить его корректно работать под Linux не представлялось возможным. Поэтому было решено использовать компьютеры с операционной сиситемой Linux совместно с компьютерами, оснащенными Windows. Чтобы не

выбрасывать деньги на ветер и обеспечить всех пользователей единой офисной средой в качестве альтернативы пакету Microsoft Office, на всех персональных компьютерах был установлен OpenOffice.

Признаюсь честно, по-началу было довольно тяжело. Пользователи очень негативно, практически в штыки, восприняли идею новой операционной системы. Я был единственным человеком в организации, кто в ней хоть как-то разбирался, и поэтому мне кроме настройки и администрирования компьютеров, пришлось еще и обучать пользователей работе в новых условиях. Спасало ситуацию только то, что перевод пользователей происходил поэтапно, и те, которые уже освоились в Linux давали советы новичкам. Наибольшее количество нареканий вызвал офисный пакет OpenOffice, особенно у тех пользователей, которые были плохо знакомы даже с Microsoft Office. Пользователей смущало непривычное расположение пунктов меню и пиктограмм панели инструментов, разные комбинации горячих клавиш, а главное то, что документы со сложной структурой, сохраненные в форматах Microsoft Office, отображались в OpenOffice неправильно. Ни для кого также не секрет, что OpenOffice довольно ресурсоемкое приложение и по сравнению со своим более именитым конкурентом работает на порядок медленнее. Среднестатистический пользователь не может ждать открытия файла больше 5 секунд, после этого времени в его голову закрадывается мысль: "А не завис ли компьютер?" и он хаотически начинает нажимать все кнопки подряд, а потом звать системного администратора. В результате мне постоянно приходилось слышать жалобы пользователей слабомощных компьютеров на то, что их компьютер тормозит и его нужно срочно модернизировать. Обращение с аналогичным предложением к начальству воспринималось без особого энтузиазма, так как требовало финансовых затрат, и в этом случае самым популярным ответом был: "Денег на модернизацию нет".

К тому времени я уже был знаком с механизмом дистанционной работы в графическом режиме через протокол SSH. Я даже искал способы подключения по этому протоколу слабеньких компьютеров с Linux к более мощным, для использования последних в роли сервера. Мною в тестовых целях был написан

сценарий автоматического подключения к удаленному компьютеру через протокол SSH сразу после загрузки операционной системы. В принципе, это решение было довольно работоспособным, но ему не хватало универсальности, а главное стабильности. По не выясненным причинам некоторые приложения отказывались работать через SSH и зависали сразу после запуска или в середине работы. Очевидно сказывался недостаток знаний в Linux, поэтому я стал искать другие альтернативные решения проблемы, связанной с низкой с производительностью пользовательских компьютеров.

Выход из сложившейся ситуации был найден в разговоре со знакомым системным администратором Владиславом Панченко, который, узнав о моей проблеме, посоветовал обратить внимание на использование X-терминалов в операционной системе Linux и посетить сайт http://www.ltsp.ru/. Я так и сделал и уже через неделю стал яростным поклонником бездисковых компьютеров. В этом проекте меня, в первую очередь, привлекла идея использования централизированных вычислений при минимальных финансовых затратах, максимум, что было нужно - это приобрести высокопроизводительный сервер. Конечно, на словах описано все было очень интересно, а вот как это будет работать на практике, мне предстояло узнать на собственном опыте.

Для своих экспериментов начальство разрешило мне использовать компьютер из отдела сбыта, который был не очень загружен работой. Конфигурация у него была очень скромная (см. номер 7 в Таблице 1), но так как мне требовалась тестовая площадка, то он вполне подходил. Загрузив из сети Интеренет необходимые пакеты проекта LTSP я приступил к настройке своего первого сервера X-терминалов. Не могу сказать, что все прошло гладко с первого раза. Для получения работоспособной конфигурации я провозился с ним неделю (никак не мог подключиться к серверу шрифтов), но конечный результат меня порадовал. Загрузившись с дискеты, которая содержала образ BootROM для моего сетевого адаптера, через несколько секунд я увидел на экране графический экран входа в удаленный компьютер. Я ввел свое имя пользователя и пароль и вошел в свой сеанс, при этом в это же время непосредственно на этом компьютере только в другой комнате работал

совершенно другой человек. Так как конфигурация импровизированного X-терминал сервера была очень скромной (256 Мбайт оперативной памяти для выполнения подобных функций очень мало), то я не могу сказать, что был восхищен скоростными показателями его работы. Но скорость запуска OpenOffice меня приятно удивила. Как я потом узнал, секрет был в том, что другой пользователь также использовал этот офисный пакет, а следовательно часть разделяемых библиотек уже была загружена в оперативную память компьютера, что и объясняет его более быстрый старт. Особо порадовала меня стабильность в работе X-терминала. При тестировании работоспособности в различных прикладных программах не было замечено ошибок и решил постепенно переходить к переводу пользователей на X-терминалы.

Оставались еще опасения по поводу одновременной работы большого числа пользователей на одном сервере, поэтому для того чтобы окончательно убедиться в надежности X-терминального решения, я решил сначала смоделировать реальную работу для 3-х пользователей. В качестве нового сервера X-терминалов был выбран компьютер отдела сбыта №10 из Таблицы 1. Для того, чтобы хоть как-то приблизить его конфигурацию к реальному серверу, был наращен объем оперативной памяти до 512 Мбайт. К нему подключались два X-терминала, переделанные из компьютеров №4 и №11 той же таблицы. Только теперь они были более облегченных конфигураций:

Процессор - Intel Pentium MMX 166 МГц, материнская плата на чипсете i430, оперативная память 16 Мб (SIMM FPM), видеоадаптер PCI s3 trio, 2 Мб, сетевая карта PCI rtl8029 10 Мбит/с, флоппи-дисковод, корпус AT 200 Вт;

Процессор - Intel Pentium MMX 200 МГц, материнская плата на чипсете i430, оперативная память 32 Мб (DIMM SDRAM), видеоадаптер PCI s3 trio, 1 Мб (интегрированный на материнской плате), сетевая карта PCI rtl8029 10 Мбит/с, флоппи-дисковод, корпус ATX 200 Вт.

Фактически из них были извлечены винчестеры, приводы CD-ROM, а также избыток оперативной памяти. Загружались X-терминалы пока с дискет, поэтому дисководы были оставлены в их конфигурации.

Тестовый X-терминал сервер (Процессор - Celeron 1 ГГц, ОЗУ – 512 Мбайт) также оснащен был самой обыкновенной сетевой картой на 10 Мбит/с, что несколько замедляло дистанционную работу пользователей, но ввиду их ограниченного числа не было большой проблемой. На X-терминал сервере также непосредственно работал один пользователь и, как я осознал позже, – это была большая ошибка. В один из рабочих дней этому пользователю нужно было раньше уйти домой, и он, отпросившись у начальства, встал из-за своего рабочего места и выключил компьютер. В ту же секунду погасли экраны 2-х соседних X-терминалов. Вообще, мне кажется, что среди пользователей есть люди, которые не могут понять, что с компьютерной техникой нужно обращаться хотя бы осторожно. В продолжение предыдущего примера хочу привести еще один случай с первым сервером X-терминалов, который мне доводилось обслуживать. В моменты моего отсутствия пользователи регулярно ставили на его системный блок цветок в горшке и даже додумались поливать его в таком положении!

Если смотреть на X-терминал глазами пользователей, то им безусловно понравилась скорость его работы. До этого они имели дело с не очень производительными компьютерами и им приходилось долго ждать запуска OpenOffice и открытия документов. Теперь ситуация изменилась в лучшую сторону, так как 512 Мбайт оперативной памяти вполне хватает для одновременной работы 3-х пользователей этого офисного пакета. К недостаткам, конечно, относится невозможность пользоваться дискетами, так как дисководы предназначены для загрузки X-терминалов, но в последнее время необходимость в этом возникает все реже.

Несмотря на то, что сложность внедрения X-терминалов, в первую очередь, связана с новой операционной системой и незнакомым программным обеспечением, у пользователей может сложиться чисто психологическое неприятие данного вида компьютеров. Основывается оно, как правило, на неспособности обособленной работы X-терминала без постоянной связи с сервером. Лично мне поначалу приходилось выслушивать замечания на подобие: "Что это ты за компьютер нам такой древний поставил, что без

локальной сети даже не включается?". Принцип собственности довольно сильно развит у человека, и помню, поначалу мне долго приходилось рассказывать о преимуществах дистанционной работы.

X-терминалы зарекомендовали себя как достойная альтернатива обычным персональным компьютерам с операционной системой Linux, и поэтому, когда встал вопрос о переезде офиса в другое здание, решено было постепенно все компьютеры перевести в терминальный режим. Учитывая эти особенности, на новом офисе прокладывалась локальная сеть в расчете на ее интенсивное использование. Так, вместо локальных концентраторов (один на комнату) от каждого рабочего места шел сетевой кабель в специальную комнату – серверную. Выделение изолированного помещения специально для работы оборудования - это вообще хорошая идея, так как тем самым вы сводите к минимуму негативное вмешательство окружающих и предотвращаете несанкционированный доступ к нему. В этом же помещении было решено разместить офисную АТС, Интернет-сервер, X-терминал сервер, источники бесперебойного питания, SHDSL-модем и модем для радио-интернета.

При выборе аппаратной базы для сервера X-терминалов за основу была положена идея экономической целесообразности, то есть сервер должен обладать нужными характеристиками для обслуживания до 20 X-терминалов, но при этом иметь разумную стоимость. Окончательной конфигурация сервера была такой:

Процессор: Intel Pentium 4 – 2,6 ГГц, кеш L2 – 512 Мб, FSB – 800 МГ, технология Hyper-Threading;

Материнская плата: ASUS P4800 Deluxe, чипсет i865PE, южный мост ICH5R с поддержкой RAID уровня 0;

Оперативная память: 1 Гб двухканальной DDR PC3200 (2 x 512 Мбайт);

Жесткий диск: WD 120 Гб, кеш 8 Мб, SATA;

CD-ROM: Asus 52x;

Флоппи-дисковод: 3,5"

Сетевые карты: интегрированная 3Com 3C940 1000 Мбит/с, PCI rtl8139C Fast Ethernet 100 Мбит/с – 2 шт;

Корпус: 300 Вт.

Видео-картой компьютер не был укомплектован, так как в наличии уже имелась подходящая, с чипсетом ATI Radeon 8500, которая и была использована в новом X-терминал сервере. В дополнение к серверу был куплен источник бесперебойного питания производства Mustek – PowerMust 1000VA Plus. В такой конфигурации, на момент покупки, стоимость сервера X-терминалов составила чуть больше 800 $, что, согласитесь, совсем не много для компьютера, рассчитанного на одновременную работу 20 пользователей. Данная модель компьютера имела довольно неплохой запас возможности будущей модернизации, так как всегда можно нарастить оперативную память (до 4 Гбайт), увеличить емкость дисковой подсистемы и поставить более мощный процессор (до 3,6 ГГц).

К настройке сервера X-терминалов я приступил сразу после того, как он попал ко мне на стол. Устанавливал я на него операционную систему ASPLinux 9.0 Express, с которой был неплохо знаком. После проверки параметров BIOS, я начал инсталляцию системы. Первым разочарованием, которое меня постигло при этом, была невозможность продолжать установку на диск с интерфейсом Serial ATA: ядро ASPLinux 9.0 попросту не видело его. Переключение в совместимый режим использования SATA лишало возможности работать с одним из IDE-контроллеров, но похоже, что это был единственный выход. Я делал даже попытку установки ядра версии 2.6, но после таких экспериментов пришел к выводу, что не смогу добиться стабильной работы системы, которая так важна для операционной системы сервера X-терминалов. Поэтому я решил остановиться на стабильной сборке от ASPLinux.

Установка пакетов, выбранных для X-терминал сервера, прошла без особых проблем. Система с первого раза успешно загрузилась и на экране я увидел графическое приглашение ввода имени пользователя и пароля. Из оборудования определилось практически все, кроме интегрированной сетевой карты от 3COM, но это и не удивительно, так как это относительно новое устройство, а исходный код его драйвера всегда можно взять с официального сайта производителя. Визуально оценив производительность системы, я

обнаружил очень медленное выполнение дисковых операций. Утилита hdparm показывала что-то в районе 3,5 Мбайт/с. Я сначала подумал, что это из-за работы SATA-винчестера в совместимом режиме, но после обновления ядра до текущей версии, доступной с сайта ASPLinux (ftp://ftp.asplinux.ru/pub/), эти проблемы исчезли. Еще при установке операционной системы, 120 Гбайтный жесткий диск был разбит на 13 разделов, все из которых были отформатированы в формате ext3. Ниже представлен их список:

```
# /sbin/fdisk -l /dev/hda

Disk /dev/hda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *        1       638   5124703+  83  Linux
/dev/hda2          639     14593 112093537+   5  Extended
/dev/hda5          639       893   2048256   82  Linux swap
/dev/hda6          894      3443  20482843+  83  Linux
/dev/hda7         3444      3826   3076416   83  Linux
/dev/hda8         3827      4464   5124703+  83  Linux
/dev/hda9         4465      5102   5124703+  83  Linux
/dev/hda10        5103      6377  10241406   83  Linux
/dev/hda11        6378      9565  25607578+  83  Linux
/dev/hda12        9566     10075   4096543+  83  Linux
/dev/hda13       10076     10585   4096543+  83  Linux
/dev/hda14       10586     10713   1028128+  83  Linux
/dev/hda15       10714     11988  10241406   83  Linux
/dev/hda16       11989     14593  20924631   83  Linux
```

Что-бы лучше понять их предназначение, приведу фрагмент файла /etc/fstab:

```
# /etc/fstab: static file system information.
```

```
#
#
/dev/hda5      none   swap  sw     0       0
/dev/hda1      /      ext3  defaults,errors=remount-ro      0       1
/dev/hda7      /tmp   ext3  defaults      0       1
/dev/hda8      /var   ext3  defaults      0       1
/dev/hda9      /usr   ext3  defaults      0       1
/dev/hda6      /home  ext3  defaults      0       1
proc     /proc  proc   defaults      0       0
none                /dev/shm            tmpfs   defaults       0 0
none     /dev/pts/     devpts        gid=5,mode=620   0       0
/dev/cdrom                          /mnt/cdrom                          udf,iso9660
noauto,owner,kudzu,ro,iocharset=cp1251 0 0
/dev/fd0                                  /mnt/floppy                          auto
noauto,user,owner,kudzu,iocharset=cp1251,codepage=866 0 0
/dev/sda1                                  /mnt/usb_manli                          auto
noauto,user,iocharset=cp1251,codepage=866 0 0
/dev/sdb1                                  /mnt/usb_flash                          auto
noauto,user,iocharset=cp1251,codepage=866 0 0
# Alker common for DOCs
/dev/hda12     /mnt/alker_common       ext3
  rw,nosuid,dev,noexec,auto,nouser,async      0       1
# Common for all win PC
/dev/hda13     /mnt/win_access    ext3   rw,nosuid,dev,noexec,auto,nouser,sync
  0       1
# Backup partitions
/dev/hda11     /mnt/backup        ext3   defaults      0       1
```

Теперь лучше видно для каких целей используются разделы жесткого диска (см. Таблицу 2):

Таблица 2. Распределение разделов жесткого диска X-терминал сервера

Имя раздела	Точка монтирования	Размер, Гб	Предназначение

/dev/hda1	/	5	Корневая файловая система
/dev/hda5		2	Файл подкачки (swap) виртуальнойпамяти
/dev/hda6	/home	20	Домашний каталог пользователей
/dev/hda7	/tmp	3	Каталог для хранения временныхфайлов
/dev/hda8	/var	5	Каталог для хранения изменяемыхфайлов, таких как журналы, почтовые базы и т.п.
/dev/hda9	/usr	5	Каталог для программногообеспечения
/dev/hda10	не примонтирован	10	Резерв
/dev/hda11	/mnt/backup	25	Для резервного копирования
/dev/hda12	/mnt/alker_common	4	Общий доступ для пользователейX-терминалов
/dev/hda13	/mnt/win_access	4	Общий доступ для Windowsкомпьютеров
/dev/hda14	не примонтирован	1	Резерв
/dev/hda15	не примонтирован	10	Резерв
/dev/hda16	не примонтирован	20	Резерв

Более 40 Гбайт на диске было оставлено для будущего использования, а схема расположения различных областей жесткого диска на разных разделах позволяет лучше контролировать использование дискового пространства. После года эксплуатации сервера X-терминалов его жесткий диск заполнен менее чем наполовину:

```
$ df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/hda1            5044156    203544   4584380   5% /
/dev/hda7            3028080     68608   2805652   3% /tmp
/dev/hda8            5044156   1037540   3750384  22% /var
/dev/hda9            5044156   2198856   2589068  46% /usr
/dev/hda6           20161172   8995744  10141288  48% /home
none                 515052         0    515052   0% /dev/shm
/dev/hda12            4030684     44400   3784752   2% /mnt/alker_common
/dev/hda13            4030684   1473964   2355188  39% /mnt/win_access
/dev/hda11           25204468     32820  23911428   1% /mnt/backup
```

Для удобства пользователей на сервере были установлены шрифты TrueType: Arial, Courier New и Times New Roman. Выбор именно этих шрифтов не случаен, так как при обмене данными с пользователями Windows (а таких, как вы знаете, подавляющее большинство в мире) в документах используются именно они.

Когда проверка работоспособности будущего сервера X-терминалов была закончена и проведено тестирование использующегося программного обеспечения, я непосредственно приступил к установке пакетов LTSP. На сервере, в частности, были установлены такие пакеты:

```
$ rpm -qa | grep ltsp
ltsp_core-3.0.9-0
ltsp_x336_svga-3.0.0-0
ltsp_x336_s3-3.0.0-0
ltsp_kernel-3.0.13-0
ltsp_x336_s3v-3.0.0-0
ltsp_x_core-3.0.4-0
ltsp_floppyd-3.0.0-0
```

И пакет из тарбола ltsp_sound-3.0.1-i386.tgz для поддержки звука на X-терминалах. Так как опыт настройки сервера X-терминалов уже имелся, то особых проблем замечено не было. Правда, для работы украинской раскладки клавиатуры был использован файл /opt/ltsp/i386/usr/X11R6/lib/X11/xkb/symbols/ru_UA из пакета русской редакции LTSP (сайт http://www.ltsp.ru). Кроме этого пришлось вручную настроить автоматический запуск демона для работы по протоколу TFTP.

Когда была проверена работа одного тестового X-терминала, я приступил к более тонкой настройки сервера X-терминалов. В частности, была выбрана графическая заставка с желтым цветком, так как стандартный экран входа LTSP показался мне очень унылым. Далее, чтобы пользователи не экспериментировали с различными оконными менеджерами, из каталога /etc/X11/gdm/Sessions были удалены все ссылки на не использующиеся:

```
$ ls /etc/X11/gdm/Sessions/
```

default Failsafe IceWM

Как сами можете убедиться, оставлен был только менеджер окон IceWM, и он же был сделан используемым по умолчанию (символическая связь с файлом default). В параметрах конфигурации оконного менеджера IceWM были сделаны значительные изменения. Во-первых, он был обновлен до самой новой на тот период версии 1.2.13 (с поддержкой системного трея). Во-вторых, специально было отредактировано содержимое панели быстрого запуска для доступа к наиболее популярным прикладным программам, которыми стали: Xterm, MToolsFM, Galeon, OpenOffice, ROX-Filer, Kedit, Evolution и сценарий для принудительного завершения "повисшего" в памяти OpenOffice. В-третьих, для некоторых программ были прописаны "горячие" клавиши доступа. Например, для запуска браузера достаточно было нажать Ctrl+Alt+I, а для открытия окна файлового менеджера – Ctrl+Alt+E. В четвертых, чтобы сохранить однообразное представление рабочих столов всех пользователей X-терминалов, из главного меню IceWM был изъят пункт, отвечающий за выбор темы оформления. В пятых, на панели задач были оставлены часы и три индикатора, отвечающие за загрузку центрального процессора и двух сетевых карт сервера (eth0 и eth1). Также было ограничено количество виртуальных рабочих столов до трех, так как среднестатистическому пользователю этого числа вполне достаточно. Внешний вид рабочего стола X-терминала с открытым документом в OpenOffice вы можете увидеть на рис. А.1:

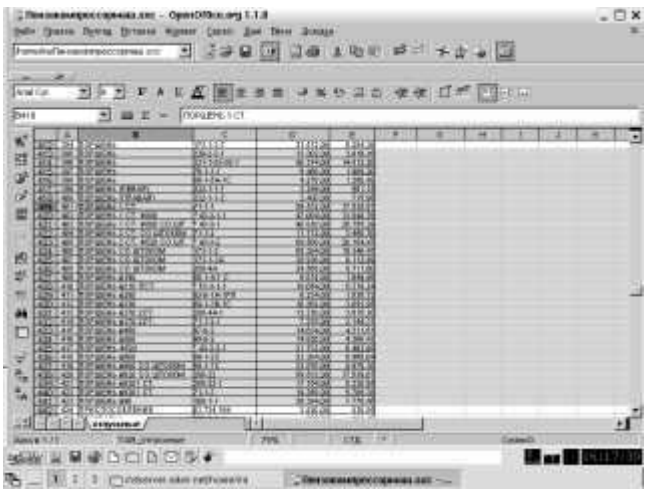

Рис. А.1. Внешний вид экрана работающего X-терминала с открытым документом в OpenOffice

Конфигурационных изменений досталось и другим программным продуктам. Больше всех в этом плане "пострадал" файловый менеджер ROX-Filer, для которого, практически с нуля, была создана система ассоциаций документов с программами, и были частично переделаны иконки к файлам.

В этом приложении я рассказываю о реальном внедрении X-терминалов в офисе предприятия. Поэтому в подтверждение моих слов хочу привести реальные распечатки конфигурационных файлов работающего сервера X-терминалов.

Файл соответствия доменных имен IP-адресам - /etc/hosts:

```
127.0.0.1       localhost.localdomain localhost

192.168.0.26 xtserver.alker.net   xtserver      alker  phones ATS senyk elektro
192.168.1.1   xtserver.alker.net   xtserver

192.168.0.3   account.alker.net   account
192.168.0.5   tanya.alker.net              tanya
192.168.0.20  pasha.alker.net              managers
192.168.0.22  inet.alker.net               inet

192.168.1.13  sergx.alker.net              sergx
192.168.1.14  pxeadmin.alker.net           pxeadmin
192.168.1.15  manalx.alker.net   manalx
192.168.1.16  mihmurx.alker.net mihmurx
192.168.1.18  irax.alker.net               irax
192.168.1.23  natax.alker.net              natax
192.168.1.3   ulax.alker.net               ulax
192.168.1.8   disp.alker.net               disp
192.168.1.6   yanax.alker.net              yanax
```

Файл конфигурации протокола DHCP - /etc/dhcpd.conf:

```
ddns-update-style        none;
```

```
default-lease-time          21600;
max-lease-time              21600;

option subnet-mask          255.255.255.0;
option domain-name          "alker.net";

option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.1.0 netmask 255.255.255.0 {
   }
}

group   {
   use-host-decl-names       on;
   option log-servers        192.168.1.1;
   option broadcast-address      192.168.1.255;
   option routers                192.168.1.254;
   option domain-name-servers    192.168.1.254;
   option root-path              "192.168.1.1:/opt/ltsp/i386";

   host natax {
      hardware ethernet     00:80:48:29:3B:53;
      fixed-address         192.168.1.23;
      filename              "/lts/vmlinuz-2.4.24-ltsp-1";
   }
   host manalx {
      hardware ethernet     00:80:48:29:67:42;
      fixed-address         192.168.1.15;
      filename              "/lts/vmlinuz-2.4.24-ltsp-1";
```

```
  }
  host sergx {
     hardware ethernet     00:80:48:24:80:63;
     fixed-address         192.168.1.13;
     filename              "/lts/vmlinuz-2.4.24-ltsp-1";
  }
  host mihmurx {
     hardware ethernet     00:50:FC:FE:A8:F6;
     fixed-address         192.168.1.16;
     filename              "/lts/vmlinuz-2.4.24-ltsp-1";
  }
  host irax {
     hardware ethernet     00:50:FC:FE:4F:BF;
     fixed-address         192.168.1.18;
     filename              "/lts/vmlinuz-2.4.24-ltsp-1";
  }
  host ulax {
     hardware ethernet     00:80:48:1F:33:99;
     fixed-address         192.168.1.3;
     filename              "/lts/vmlinuz-2.4.24-ltsp-1";
  }
  host pxeadmin {
hardware ethernet     00:11:2F:CC:69:89;
fixed-address              192.168.1.14;
filename              "/lts/2.4.24-ltsp-1/pxelinux.0";
  }
  host pashax {
     hardware ethernet     00:30:84:0F:D8:CA;
     fixed-address         192.168.1.8;
     filename              "/lts/vmlinuz-2.4.24-ltsp-1";
  }
```

```
    host yanax {
        hardware ethernet     00:50:FC:FF:C8:DF;
        fixed-address         192.168.1.6;
        filename              "/lts/vmlinuz-2.4.24-ltsp-1";
    }
}
```

Файл экспортируемых файловых систем NFS - /etc/export:

```
## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Срд Лют 25 12:43:52 EET 2004 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/opt/ltsp/i386                  192.168.1.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles
192.168.1.0/255.255.255.0(rw,no_root_squash,async)

#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home                 192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

## LTS-end ##

# Для пользователей Алкер
/mnt/alker_common 192.168.0.0/255.255.255.0(rw,no_root_squash,sync)
# Для пользователей win-ПК
```

/mnt/win_access 192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

И наконец главный конфигурационный файл проекта LTSP - /opt/ltsp/i386/etc/lts.conf:

```
#
# Config file for the Linux Terminal Server Project (www.ltsp.org)
#

[Default]
    SERVER          = 192.168.1.1
  DISABLE_ACCESS_CONTROL = Y
    XSERVER         = auto
  X_MOUSE_PROTOCOL   = "PS/2"
  X_MOUSE_DEVICE     = "/dev/psaux"
  X_MOUSE_RESOLUTION = 400
  X_MOUSE_BUTTONS    = 3
  XkbLayout  = "ru_UA"
  X_COLOR_DEPTH      = 16
  X_MODE_0        = 800x600
  USE_XFS         = Y
  LOCAL_APPS      = N
    RUNLEVEL        = 5

#-------------------------------------------------------------------------------

[sergx]
    XSERVER         = auto
  RCFILE_01 = floppyd
  XkbLayout  = "ru"
  X_MOUSE_PROTOCOL   = "IntelliMouse"
  X_MOUSE_DEVICE     = "/dev/ttyS0"
  X_MOUSE_EMULATE3BTN = Y
```

```
  X_HORZSYNC       = "30-60"
  X_VERTREFRESH          = "56-60"
      USE_NFS_SWAP       = Y
      SWAPFILE_SIZE      = 48m
  X_MODE_0         = 1024x768
  # Sound ES1688F ISA not tested

[natax]
  RCFILE_01 = floppyd
      XSERVER            = auto
  X_MOUSE_PROTOCOL   = "IMPS/2"
      USE_NFS_SWAP       = N
      SWAPFILE_SIZE      = 8m
  X_VERTREFRESH          = "50-85"
  MODULE_01 = usb-uhci
  MODULE_02 = printer
  PRINTER_0_DEVICE = /dev/usb/lp0
  PRINTER_0_TYPE = S

[manalx]
  RCFILE_01 = floppyd
      XSERVER            = XF86_S3
  X_MOUSE_PROTOCOL   = "IntelliMouse"
  X_MOUSE_DEVICE     = "/dev/ttyS0"
  X_MOUSE_EMULATE3BTN = Y
      USE_NFS_SWAP       = Y
      SWAPFILE_SIZE      = 64m
  PRINTER_0_DEVICE   = /dev/lp0
      PRINTER_0_TYPE     = P           # P-Parallel, S-Serial

[mihmurx]
```

```
  RCFILE_01 = floppyd
      XSERVER          = XF86_S3
  X_MOUSE_EMULATE3BTN = Y
      USE_NFS_SWAP      = Y
      SWAPFILE_SIZE     = 64m
  PRINTER_0_DEVICE   = /dev/lp0
      PRINTER_0_TYPE     = P          # P-Parallel, S-Serial

[irax]
  RCFILE_01 = floppyd
      XSERVER          = XF86_S3
  X_MOUSE_PROTOCOL   = "IntelliMouse"
  X_MOUSE_DEVICE     = "/dev/ttyS0"
  X_MOUSE_EMULATE3BTN = Y
  X_VERTREFRESH          = "50-84"
      USE_NFS_SWAP      = Y
      SWAPFILE_SIZE     = 64m

[adminx]
      RUNLEVEL          = 3
  RCFILE_01 = floppyd
  X_MOUSE_PROTOCOL   = "IMPS/2"
  X_MOUSE_RESOLUTION = 250
  X_MOUSE_BAUD = 9600
      USE_NFS_SWAP      = Y
      SWAPFILE_SIZE     = 64m
  X_MODE_0         = 1024x768
  X_HORZSYNC       = "30-96"
  X_VERTREFRESH          = "50-160"
  X_DisplaySize = "320 240"
  SOUND            = Y
```

```
        SOUND_DAEMON       = esd   # This can be 'nasd', or 'esd' at this time
        VOLUME             = 100    # Speaker & WAVE volume pecentage
    SMODULE_01         = soundcore
    SMODULE_02         = sound
        SMODULE_03         = ad1848
    SMODULE_04         = uart401
    SMODULE_05         = opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5
dma=1 dma2=0

  [pxeadmin]
        RUNLEVEL           = 3
    XSERVER            = vesa
  #RCFILE_01 = floppyd
    X_MOUSE_PROTOCOL   = "IMPS/2"
    X_MOUSE_BAUD = 9600
    X_MODE_0           = 1024x768
    X_HORZSYNC         = "30-96"
    X_VERTREFRESH          = "50-160"

  [ulax]
    RCFILE_01 = floppyd
    X_MOUSE_PROTOCOL   = "IMPS/2"
        USE_NFS_SWAP       = Y
        SWAPFILE_SIZE      = 92m
    X_HORZSYNC         = "30-96"
    X_VERTREFRESH          = "50-160"
  #SOUND             = N
  #       SOUND_DAEMON       = esd   # This can be 'nasd', or 'esd' at this time
  #       VOLUME             = 50    # Speaker & WAVE volume pecentage
  #SMODULE_01         = soundcore
  #SMODULE_02         = sound
```

```
#SMODULE_03        = ad1848
#SMODULE_04        = uart401
#SMODULE_05        = cs4232 irq=5 io=0x220 dma=1
  PRINTER_0_DEVICE   = /dev/lp0
      PRINTER_0_TYPE     = P          # P-Parallel, S-Serial

[pashax]
      XSERVER           = XF86_S3
  X_MOUSE_EMULATE3BTN = Y
  X_VERTREFRESH          = "60-84"
  X_MOUSE_PROTOCOL   = "IntelliMouse"
  X_MOUSE_DEVICE     = "/dev/ttyS0"
      USE_NFS_SWAP       = Y
      SWAPFILE_SIZE      = 92m

[overx]
  X_MODE_0         = 1024x768
  X_MOUSE_EMULATE3BTN = Y
      USE_NFS_SWAP       = N


[yanax]
      USE_NFS_SWAP       = Y
      SWAPFILE_SIZE      = 32m
  RCFILE_01 = floppyd
  X_MOUSE_PROTOCOL   = "IMPS/2"
```

Интересно, что работающий сервер X-терминалов регистрирует в системном журнале сообщения от всех X-терминалов локальной сети. Для примера приведу результат выполнения команды на реальной системе:

```
# tail /var/log/messages
Mar 29 16:44:19 irax dhclient: DHCPACK from 192.168.1.1
```

Mar 29 16:44:19 irax dhclient: bound to 192.168.1.18 -- renewal in 10800 seconds.

Mar 29 16:44:24 irax -- MARK --

Mar 29 16:45:43 xtserver su(pam_unix)[6743]: session opened for user root by mikola(uid=501)

Mar 29 16:52:17 pxeadmin -- MARK --

Mar 29 16:52:19 xtserver gconfd (ira-6498): GConf server is not in use, shutting down.

Mar 29 16:52:19 xtserver gconfd (ira-6498): Exiting

Mar 29 16:52:22 natax -- MARK --

Mar 29 16:54:14 xtserver su(pam_unix)[6848]: authentication failure; logname=mikola uid=501 euid=0 tty= ruser=mikola rhost= user=root

Mar 29 16:54:21 xtserver su(pam_unix)[6849]: session opened for user root by mikola(uid=501)

Теперь разрешите перейти непосредственно к описанию того, как X-терминалы работают на практике. Чтобы лучше понять это, я кратко опишу организацию, в которой работаю. Ее офис находится на 2-м этаже 3-х этажного здания, занимает он 8 комнат. Коллектив небольшой – 40 человек, из которых с компьютерами работает только 20. Перед "заселением" офиса для работы традиционных персональных компьютеров и X-терминалов была проложена локальная сеть. На рис. А.2 вы можете увидеть ее структуру.

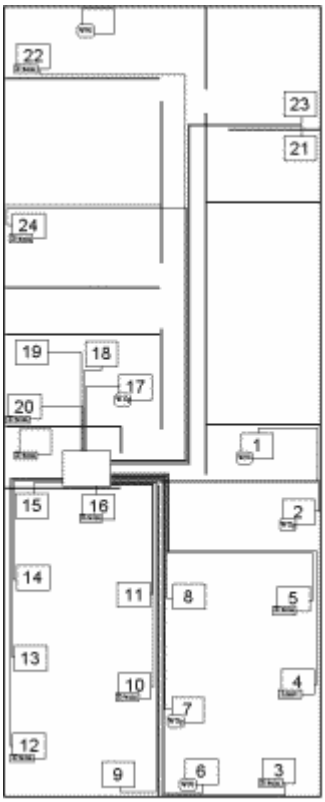

Рис. А.2. Структура локальной сети на плане офиса

К местам предполагаемого размещения пользователей был проведен сетевой кабель и размещен на стенах в виде сетевых розеток (см. рис. А.3). По мере распределения рабочих мест в офисе от сетевых розеток протягивался сетевой кабель к компьютерам пользователей.

Рис. А.3. Сетевая розетка для Х-терминала или персонального компьютера

Так как все сетевые кабели сходятся в одной комнате – серверной, то в ней и расположено практически все оборудование. Внешний вид рабочего стола системного администратора вы можете лицезреть на рис. А.4. Правда, это скорее всего иллюстрация того, как не должен выглядеть рабочий стол системного администратора, но, как говорится, не будем украшать действительность.

Рис. А.4. Рабочее место системного администратора

Внешний вид X-терминалов ничем не отличается от обычных персональных компьютеров. Тот же системный блок, те же монитор, клавиатура и "мышь". На рис. А.5 вы можете увидеть фотографию задней стенки системного блока реального X-терминала.

Рис. А.5. X-терминал (вид сзади)

На экране монитора X-терминала данные отображаются также как и на обычном персональном компьютере (см. рис. А.6). Единственное отличие во

внешнем виде – это использование операционной системы Linux и свойственного ей интерфейса.

Рис. А.6. Х-терминал за работой

Очень удачно подходят на роль будущих Х-терминалов старые персональные компьютеры в фирменных корпусах типа desktop от ведущих иностранных производителей компьютерной техники (IBM, NEC, Compaq, Dell, Siemens-Nixford и т.д.). Главные их преимущества это компактность (системный блок можно поставить под монитор), совместимость с современными устройствами ввода (как правило, "мышь" и клавиатура у таких изделий имеют интерфейс PS/2) и довольно высокое качество. Пример смотрите на рис. А.7.

Рис. А.7. Х-терминал, переделанный из корпуса типа desktop

Чтобы убедиться в том, что это действительно Х-терминал, можете также заглянуть в его "внутренности" (см. рис. А.8).

Рис. А.8. "Внутренности" Х-терминала

А так выглядит на Х-терминале приглашение входа в систему (см. рис. А.9).

Рис. А.9. Приглашение входа в систему на Х-терминале

Переделка части парка компьютерной техники в Х-терминалы помогает лучше использовать возможности персональных компьютеров. Практически все самые слабые компьютеры становятся терминалами, а более современные аппараты используются в качестве полноценных персональных компьютеров. Так, для нужд бухгалтерии были оставлены 4 компьютера с операционной системой Microsoft Windows и только один Х-терминал для работы с первичной

документацией. Их конфигурация теперь стала выглядеть следующим образом (см. Таблицу 3):

Таблица 3. Новая конфигурация компьютеров для бухгалтерии

п/п	Процессор (CPU)	Чипсет	ОЗУ	Видео	Винчестер	Дополнительноеоборудование	Операционная система	Предназначениекомпьютера выполняемые задачи
	Intel Celeron 1.8 ГГц	i845	12 Мб	AGP GeForce 2 MX/400,32 Мб	6,4 Гб	флоппи, CD-ROM, модем	Windows 98	Бухгалтерия,клиент-банк, программы для бухгалтерскойи статистическойотчетности, электронная почта
	Intel Celeron 1 ГГц	i815	256 Мб	AGP MGA G100, 4 Мб	40 Гб	флоппи, CD-ROM, модем	Windows XP HomeEdition	Бухгалтерия,клиент-банк
	Intel Celeron 800 МГц	i815	256 Мб	Интегрированное	10 Гб	флоппи, CD-ROM, модем	Windows XP HomeEdition	Бухгалтерия,клиент-банк
	Intel Celeron 433 МГц	i810	256 Мб	Интегрированное	8,4 Гб	флоппи, CD-ROM	Windows 98	Бухгалтерия, информационно-справочная система по текущему законодательству,программы для бухгалтерскойи статистическойотчетности
	Intel Celeron 500 МГц	i810	128 Мб	Интегрированное	-	флоппи	X-терминал	Выписка первичной документациии ее обработк

Как видно из Таблицы 3, персональные компьютеры бухгалтерии стали более мощными, что положительно сказалось на производительности труда сотрудников этого производственного подразделения.

Для полной легализации программного обеспечения пришлось приобрести только 5 лицензий на операционные системы Microsoft ориентировочной стоимостью 500 $ (примерно по 100 $ за OEM-версию), из которых 4 были использованы для нужд бухгалтерии, а одна для работы специализированной программы транспортного отдела. Хоть это и звучит кощунственно, но в распространении операционной системы Linux в офисах нашей страны самое активное участие приняла компания Microsoft. Ее законные действия по пресечению нелегального использования программ совместно с другими ведущими производителями коммерческого программного обеспечения, а также мероприятия местных правоохранительных органов создали практически идеальную среду для развития свободного программного обеспечения в целом и Linux в частности. Интересно, что когда руководители предприятий начинают понимать, что за программные продукты нужно платить, то альтернатива использования свободных программ, как никогда, кажется заманчивой. А если учесть, что в некоторых аспектах операционная система Linux даже превосходит аналогичные коммерческие продукты, то для многих правильный выбор становится просто очевиден.

Переделка обычных персональных компьютеров в терминалы шла по простой схеме. Выбирался, как правило, самый слабый свободный системный блок (см. рис. А.10). После снятия кожуха корпуса (см. рис. А.11) из него извлекались все лишние детали: винчестер, CD-ROM, звуковая карта, лишние модули памяти, не использующиеся интерфейсные кабели, а также иногда флоппи-дисковод.

Рис. А.10. Типичный кандидат на роль X-терминала

Рис. А.11. Будущий X-терминал в открытом виде

Таким образом получался бездисковый компьютер (см. рис. А.12), для работы которого требовалась установка сетевой карты с функциями загрузки по сети (BootROM или PXE), ну или в редких случаях обычная загрузочная дискета.

Рис. А.12. Х-терминал = ПК без всего лишнего

Если посмотреть на содержимое корпуса Х-терминала повнимательнее, то легко можно заметить, что от обычного персонального компьютера он мало чем отличается (см. рис. А.13). Просто комплектующих поменьше и из-за этого в корпусе попросторнее.

Рис. А_13. Внутри будущего Х-терминала

Интересной получилась история приобретения и прошивки микросхем EPROM для использования их в качестве загрузки сетевыми картами. Для того, чтобы обеспечить на предприятии нужную производительность локальной сети Х-терминалов было принято решение приобрести новые сетевые адаптеры со скоростью 100 Мбит/с. Выбор был сделан в пользу сетевых карт на чипсете rtl8139. Всего было куплено 8 таких устройств. Разъемы BootROM в них были пусты, и в поисках нужных микросхем, я отправился по магазинам

компьютерной техники. К великому моему сожалению, ни в одной компьютерной конторе мне так и не предложили нужных микросхем, более того никто даже не согласился привести их под заказ. Но так как использовать загрузочные функции сетевых карт было необходимо, я обратился в ближайший магазин радио товаров. К моему удивлению, моя просьба их ни сколько не удивила, и мне на выбор был предложен целый список совместимых микросхем. Через неделю, я уже умел в своем распоряжении все необходимо для сетевой загрузки X-терминалов. Дело оставалось за малым: найти программатор и прошить в эти микросхемы загрузочный код. Получив в сайта http://www.rom-o-matic.net нужный образ (благо, чипсеты на сетевых картах были одинаковые), я отправился к знакомому специалисту, и он за чисто символическое вознаграждения прошил все 8 микросхем. Правда, у меня в начаое были опасения по поводу правильности работу этого метода загрузки по сети, и так как микросхемы BootROM были одноразовые, то сначала мне прошили только один образец, а уже после того, как я удостоверился в его 100% работоспособности, прошили и остальные.

После окончания процесса "перераспределения комплектующих" между традиционными персональными компьютерами и X-терминалами получился такой парк компьютерной техники (см. Таблицу 4):

Таблица 4. Парк компьютеров после внедрения X-терминалов

Процессор (CPU)	Чипсет	ОЗУ	Видео	Винчестер	Дополнительное оборудование	Операционная система	Предназначениекомпьютера, выполняемые задачи
Intel Celeron 1.8 ГГц	i845	512 Мб	AGP GeForce 2 MX/400,32 Мб	6,4 Гб	флоппи, CD-ROM, модем	Windows 98	Бухгалтерия,клиент-банк, программы для бухгалтерской и статистическойотчетности, электронная почта
Intel Celeron 1 ГГц	i815	256 Мб	AGP MGA G100, 4 Мб	40 Гб	флоппи, CD-ROM, модем	Windows XP HomeEdition	Бухгалтерия,клиент-банк
Intel Celeron 800	i815	256 Мб	Интегрированное	10 Гб	флоппи, CD-ROM, модем	Windows XP HomeEdition	Бухгалтерия,клиент-банк

МГц							
Intel Celeron 433 МГц	i810	256 Мб	Интегрированное	8,4 Гб	флоппи, CD-ROM	Windows 98	Бухгалтерия,информационно-справочная система по текущему законодательству,программы для бухгалтерской и статистической отчетности
Intel Celeron 566 МГц	i430	256 Мб	AGP ATI Rage Pro, 8Мб	20 Гб, 3,2 Гб	флоппи, CD-ROM, звук	ASPLinux 9.0	Отдел сбыта, офисныйпакет, интернет, электронная почта
Intel Pentium II 350МГц	i440 BX	192 Мб	AGP nVidia Riva TNT,16 Мб	4,3 Гб, 3,2 Гб	флоппи, CD-ROM,модем, звук	Windows 98	Транспортный отдел,офисный пакет, интернет, электронная почта, специализированныепрограммы для определения оптимального маршрута перевозки грузов
Intel Pentium MMX 233МГц	i430	16 Мб	PCI s3 trio, 1 Мб	-	флоппи	X-терминал	Транспортный отдел,офисный пакет, интернет, электронная почта
Intel Celeron 500 МГц	i810	128 Мб	Интегрированное	-	флоппи	X-терминал	Бухгалтерия, выпискапервичной документации и ее обработка
Intel Pentium II 233МГц	i440 BX	32 Мб	AGP SIS 6325, 4 Мб	-	флоппи, звук	X-терминал	Отдел сбыта, офисныйпакет, интернет, электронная почта
Intel Pentium MMX 166МГ	i430	16 Мб	PCI s3 trio, 1 Мб	-	флоппи	X-терминал	Отдел сбыта, офисныйпакет, интернет

ц							
Intel Celeron 466 МГц	i430	32 Мб	AGP ATI Rage, 4 Мб	-	флоппи, звук	X-терминал	Управление, офисныйпакет, интернет, электронная почта
Intel Pentium MMX 200МГц	i430	32 Мб	PCI s3 trio, 2 Мб	-	флоппи	X-терминал	Отдел сбыта, офисныйпакет, интернет, электронная почта
Intel Celeron 466 МГц	i430	256 Мб	AGP ATI Rage IIC, 4 Мб	3,2 Гб, 3,2 Гб	флоппи, CD-ROM, модем	ASPLinux 9.0	Системноеадминистрирование, офисный пакет, интернет, электронная почта,программы по управлению сетевой средой
Intel Pentium MMX 200МГц	i430	96 Мб	Интегрированное	3,2 Гб	флоппи	ASPLinux 7.3	Интернет-сервер,брандмауэр, прокси-сервер
Intel Pentium IV 2,6 ГГц	i865	1 Гб	AGP ATI Radeon 8500,64 Мб	120 Гб	флоппи, CD-ROM	ASPLinux 9.0	Сервер X-терминалов,файловый сервер, система учета телефонных переговоров офисной АТС

Как видно из итоговой таблицы, суммарное количество компьютеров увеличилось на один (сервер X-терминалов). После распределения комплектующих осталось несколько лишних ISA звуковых карт, приводов для чтения компакт-дисков, модемов, два винчестера емкостью 3,2 Гбайт и один 20 Гбайт (правда, последний через месяц вышел из строя), а модули памяти (256 Мбайт) даже пришлось докупить. Несмотря на то, что "лишних" комплектующих осталось не много, значительно улучшились характеристики традиционных персональных компьютеров за счет использования деталей, которые были извлечены из кандидатов в X-терминалы.

После того, как X-терминалы хорошо себя зарекомендовали на рабочих местах, было принято решение увеличить их количество еще на 3 компьютера (см. Таблицу 5). В результате терминальный парк пополнился еще 3-мя "развалинами" (это название они заслужили, так как на рабочих местах они появились в результате похода по магазинам б/у компьютерной техники):

Таблица 5. "Новые" X-терминалы

Процессор (CPU)	Чипсет	ОЗУ	Видео	Дополнительно еоборудование
Intel Pentium MMX 166МГц	i430	32 Мб	PCI Matrox Mystique,2 Мб	флоппи
Intel Pentium 120 МГц	i430	32 Мб	PCI s3 trio, 1 Мб	-
Intel Pentium MMX 200МГц	i430	32 Мб	Интегрированн ое	флоппи

После переезда в новый офис наша организация столкнулась с еще одной проблемой: нехваткой принтеров. Точнее принтеров в наличии было достаточно, но группировка пользователей по комнатам привела к тому, что, в частности, в бухгалтерии не оказалось принтера, который поддерживался операционной системой Linux. Точнее, в наличии были два, так называемых Win-принтера: Canon LBP-810. Некоторые люди думают, что Win-принтеры – это принтеры, сертифицированные корпорацией Microsoft. На самом деле, это устройства без “мозгов”, для печати на которых требуется специальный драйвер доступный, как правило, только для операционной системы Windows. На то время драйвера под операционную систему Linux эти принтеры не имели, и даже несмотря на то, что они были подключены к персональным компьютерам с Windows, печать на них со стороны X-терминалов была под большим вопросом. Порывшись в Интернете и проанализировав имеющиеся решения, я выбрал обходной путь, в решении которого мне помогли две программы:

Интерпретатор Ghostscript (сайт http://www.ghostscript.com/)

RedMon (Redirect monitor) (сайт http://www.ghostgum.com.au/)

К счастью, эти программы бесплатны и свободно доступны в Интернете. Чтобы очень не утомлять вас рассказом о настройке этого специфического способа печати, кратко опишу его алгоритм:

Устанавливаем на Windows компьютер интерпретатор Ghostscript;

Устанавливаем также программу Redmon, которая добавляет новый порт RPT1: ;

Через диалоговое окно управления принтерами Windows, добавляем фиктивный PostScript принтер (например, HP LaserJet 4/4MP PostScript), и указываем в его параметрах порт RPT1: ;

В параметрах порта RPT1: указываем команду вызова интерпретатора Ghostscript со специальным драйвером -sDEVICE=mswinpr2 и реальным win-принтером -sOutputFile="//spool/<Имя принтера>";

Фиктивный PostScript принтер делаем доступным для локальной сети Microsoft и назначаем ему понятное сетевое имя, например, WinPS;

На сервере X-терминалов добавляем новый сетевой принтер, в параметрах которого указываем сетевое имя “расшаренного” фиктивного PostScript принтера (что-то похожее на smb://<адрес узла>/WinPS);

В качестве драйвера-фильтра для нового принтера обязательно указываем драйвер типа-PostScript.

Как видите, не так и сложно. Хотя первый раз, когда я настраивал таким образом печать на Canon LBP-810, провозился с ним целую неделю.

После того, как X-терминалы были настроены и исправно работали, меня посетила идея загрузить сервер X-терминалов дополнительной работой: приспособить его под файловый сервер для сети Microsoft. Благо, под операционную систему Linux написана прекрасная программа, которая помогла мне в этом - сервер Samba. В организации, в которой я работаю, до этого не было выделенного сервера для хранения документов с общим доступом. Если у пользователей возникала потребность в совместном использовании одних и тех же файлов, то на одном из персональных компьютеров к папке прописывался общий доступ к файлам которой пользователи и обращались совместно. У данного подхода было 2 главных недостатка:

Очень сложно было контролировать доступ к файлам на уровне пользователей;

Компьютер с общими сетевыми папками должен был быть постоянно включенным.

У сервера X-терминалов, который стал выступать в роли файлового сервера для сети Microsoft, этих недостатков не было, так как он включен практически круглосуточно и при получении доступа к нему пользователи обязаны пройти аутентификацию.

При настройке Samba-сервера использовалась его версия 2.2.7. В главный конфигурационный файл /etc/samba/smb.conf были внесены такие изменения, для правильной работы с именами файлов, которые содержат буквы кириллицы:

```
character set = 1251
client code page = 866
preserve case = yes
short preserve case = yes
```

Так же была заблокирована возможность экспорта домашних каталогов пользователей Linux-сервера и для всех принтеров, что описаны на сервере X-терминалов, был разрешен доступ со стороны Windows компьютеров локальной сети:

```
[printers]
  comment = All Printers
  path = /var/spool/samba
  browseable = no
  guest ok = no
  writable = no
  printable = yes
```

В эксплуатации сервера Samba показал себя очень надежным программным продуктом и за один год его использования мною не было зафиксировано ни одного случая его неправильной работы. К тому же на

загрузке центрального процессора X-терминал сервера работа демонов smbd и nmbd никак не отразилась.

Частично окрыленный успехом эксплуатации сервера X-терминалов как файлового сервера для сети Microsoft, я решил воспользоваться его круглосуточной работой для выполнения еще одной производственной задачи. Нужно было настроить систему получения данных от офисной АТС (сбор информации о входящих и исходящих звонках и их тарификация). В продаже были программные пакеты тарификации телефонных переговоров, но рассчитаны они были на работу в операционной системе Windows, а подключать к АТС еще один компьютер совсем не хотелось. Ознакомившись с документацией к офисной АТС (Samsung DCS Compact II), я понял, что вся информация от нее поступает в компьютер по кабелю RS232C на последовательный порт. А получить в операционной системе Linux данные с последовательного порта проще простого, достаточно ввести команду (пример для COM1:), похожую на:

```
# cat /dev/ttyS0 > /var/log/smdr.log
```

Теперь все данные, поступающие от АТС, будут записываться в файл /var/log/smdr.log. Осталось немного поднастроить систему автоматического перехода на новый месяц и написать сценарий извлечения данных из файла отчета и представления их в нужной для человека форме. Для таких целей прекрасно подойдет язык программирования Perl, так как его функции обработки текста всегда были на высоте.

В результате сервер X-терминалов стал успешно выполнять не свойственную ему работу: вести статистику использования офисной АТС (см. рис. А.14).

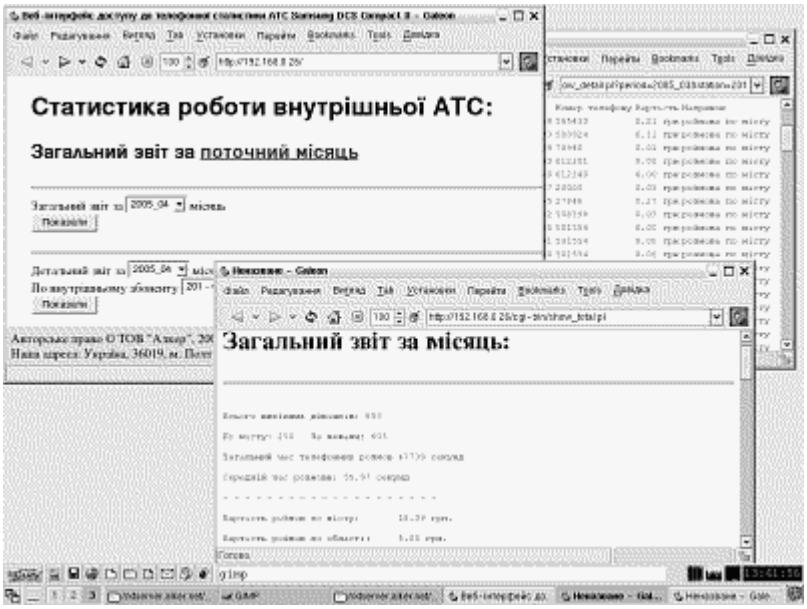

Рис. А.14. Пример отчета офисной АТС, подключенной к серверу X-терминалов

В заключении рассказа о внедрении X-терминалов на предприятии хочу привести данные о расчете экономической целесообразности такого шага. Ведь недаром все познается в сравнении. Для примера будем считать, что имеем дело не с работающим предприятием, а только с его проектом. То есть у нас еще нет ни компьютерной техники, ни программного обеспечения, а есть только потребность в них. Просчитаем стоимость 3-х вариантов построения информационной структуры предприятия:

Персональные компьютеры с операционной системой Windows;

Персональные компьютеры с операционной системой Linux;

X-терминалы под управлением операционной системы Linux.

Будем считать, что приобретаться будут новые персональные компьютеры. Базовая стоимость которых: 500$ - системный блок плюс 150$ - монитор. OEM-версия операционной системы Windows ориентировочно 100$. Для X-терминалов за ориентировочную стоимость бездискового системного блока возьмем также сумму в 100$ (за эти деньги можно купить вполне приличный X-терминал). Сервер X-терминалов начального уровня оценим в 1200$. В случае использования традиционных персональных компьютеров ситуация вполне понятна: стоимость одного компьютера с Windows и Linux отличаются только стоимостью лицензии на продукт Microsoft и составляют соответственно 750$ и 650$ (см. Таблицу 6).

Таблица 6. Анализ экономической целесообразности использования X-терминалов

Кол-во компьютеров	Windows		Linux		X-терминалы	
	Суммарные	На 1 ПК	Суммарные	На 1 ПК	Суммарные	На 1ПК
1	750	750	650	650	1450	1450,00
2	1500	750	1300	650	1700	850,00
3	2250	750	1950	650	1950	650,00
4	3000	750	2600	650	2200	550,00
5	3750	750	3250	650	2450	490,00
6	4500	750	3900	650	2700	450,00
7	5250	750	4550	650	2950	421,43
8	6000	750	5200	650	3200	400,00
9	6750	750	5850	650	3450	383,33
10	7500	750	6500	650	3700	370,00
11	8250	750	7150	650	3950	359,09
12	9000	750	7800	650	4200	350,00
13	9750	750	8450	650	4450	342,31
14	10500	750	9100	650	4700	335,71
15	11250	750	9750	650	4950	330,00
16	12000	750	10400	650	5200	325,00
17	12750	750	11050	650	5450	320,59
18	13500	750	11700	650	5700	316,67
19	14250	750	12350	650	5950	313,16
20	15000	750	13000	650	6200	310,00

Суммарные затраты, как видно из Таблицы 6, также прямо пропорциональны количеству персональных компьютеров. В X-терминальном варианте все намного интереснее. Начальные расходы связаны с приобретением дорогостоящего сервера, а затем удельная стоимость одного X-терминала уменьшается с увеличением их количества. Так, уже при 3-х компьютерах она находится на одном уровне даже с персональными

компьютерами под управлением Linux, а от 4-х и выше имеет уже значительное преимущество. Согласно данным, полученным в Таблице 6, при расчете на 20 компьютеров экономия равна 13000-6200=6800$ по сравнению с Linux-компьютерами и 15000-6200=8800$ по сравнению с компьютерами под управлением лицензионной операционной системы от Microsoft. Интересные результаты получаются от сравнения количества компьютеров при одинаковой сумме затрат. Так, если 15000$ хватает на закупку только 20 персональных компьютеров с Windows, то этой же суммы хватит на один сервер X-терминалов и 55 X-терминальных рабочих станций.

В заключение хотелось бы добавить, что, анализируя свой практический опыт эксплуатации X-терминальной локальной сети, работу системного администратора можно разделить по таким функциям:

настройка сервера X-терминалов;

настройка конечных X-терминалов;

настройка сопутствующих служб (Интернет-сервер, файл-сервер и т.п.);

работа с пользователями.

Как известно, работа с пользователями отнимает больше всего рабочего времени, причем от количества пользователей она зависит прямо пропорционально. Но в сети X-терминалов это правило не действует, так как, с одной стороны, пользователи работают в максимально единообразной среде, несмотря даже на значительные отличия в аппаратной конфигурации, а, с другой стороны, программные настройки для одного пользователя, как правило, становятся доступные и всем остальным пользователям X-терминалов. Поэтому выбор X-терминалов в качестве альтернативы традиционным персональным компьютерам не только позволит уменьшить затраты и даст возможность использовать морально устаревшее оборудование, но еще и сократит удельные расходы на обслуживание одного пользователя.

Приложение B. Отличия в проекте LTSP 4.x

Как вы знаете, все примеры данной книги ориентированы на LTSP версии 3.x. На момент написания книги самой последней версией проекта LTSP была

версия 4.1. Поэтому целью данного приложения является попытка описать наиболее важные отличия новой редакции данного программного продукта.

Итак, что же нового появилось в LTSP кроме цифры в названии? Изменений действительно много, так что обо всем по-порядку.

Установка

Во-первых, был кардинально изменен подход к установке программного продукта на сервер X-терминалов, а также способ его распространения. Вместо привычных программных пакетов для установки (или на худой конец тарболов) теперь предлагается универсальный сценарий, который загружает нужные программные компоненты прямо с официального сайта проекта (http://www.ltsp.org). Теперь пользователь уже не увидет на экране печальной надписи: "Sorry This distro is not supported by LTSP", так как с помощью пошагового интерфейса инсталлятора можно настроить работу сервера LTSP на любом дистрибутиве операционной системы Linux. Предложенный в версии 4.0 способ установки LTSP путем запуска вот такой команды:

wget -q -O - http://www.ltsp.org/ltsp_installer | sh

вряд ли мог удовлетворить всех пользователей, так как он подразумевал стабильное подключение к Интернету на довольно большой скорости. Без выделенной линии системный администратор мог попросту заснуть в процессе установки пакетов, входящих в состав LTSP. Наверное поэтому, вскоре на сайте появился ISO-образ, который требуется программе установки, а уже в версии LTSP 4.1 появился специальный пакет ltsp-utils, который необходимо установить перед началом настройки сервера X-терминалов. Доступен данный пакет в форматах RPM и TGZ и устанавливается командой, похожей на:

rpm -ihv ltsp-utils-0.10-0.noarch.rpm

После его успешной установки следует запустить утилиту ltspadmin. Для ускорения процесса копирования файлов можно вместо стандартного их расположения (http://www.ltsp.org/ltsp-4.1) указать локальную ссылку, например, file:///mnt/cdrom. Конечно, перед этим необходимо смонтировать компакт-диск с записанным ISO-образом, содержащим соответствующие пакеты LTSP.

Конфигурирование и настройка

По идеологическим соображениям от установочного сценария была отделена функция настройки работы сервера и X-терминалов. Более того, теперь в руках системного администратора появилось сразу два инструмента для администрирования и настройки сервера X-терминалов: ltspadmin и ltspcfg (обе эти утилиты являются частями пакета ltsp-utils). С помощью ltspcfg теперь можно настраивать:

Уровень выполнения (runlevel), который соответствует графическому режиму работы сервера;

Сетевые интерфейсы, которые используются для обмена данными с X-терминалами (особенно полезно для сервера с большим количеством сетевых адаптеров);

Работу демона dhcpd путем редактирования таких параметров как fixed-address, filename, subnet-mask, broadcast-address и root-path;

Работу по протоколу TFTP;

Параметры запуска portmapper, который необходим для работы RPC-сервисов;

Конфигурацию сетевой файловой системы NFS. Эти настройки не относятся к параметрам файла /etc/exports, так как за его настройку отвечает пункт номер 10;

Дистанционный доступ к менеджеру дисплея через протокол XDMCP. Поддерживается работа с такими менеджерами дисплея как XDM, GDM и KDM;

Содержимое файла /etc/hosts. Конечно, наилучшим способом обеспечения доступа к IP-адресам компьютеров является использование сервера имен (например, BIND), но его настройка и изменение конфигурации не входят в компетенцию утилиты ltspcfg. Она ограничивается простейшей статической схемой из файла /etc/hosts;

Создание файла /etc/hosts.allow, который нужен для работы службы переключателей TCP;

Создание и редактирование файла /etc/exports, который отвечает за то какие каталоги будут экспортироваться для доступа со стороны X-терминалов;

Создание главного конфигурационного файла lts.conf. Несмотря на то, что для современных бездисковых станций с шиной PCI в индивидуальных параметрах практически нет необходимости, без этого файла сложно осуществлять тонкую настройку работы X-терминалов.

Сценарии для виртуальных экранов

В новой версии проекта LTSP была пересмотрена концепция использования режима выполнения X-терминала (параметр RUNLEVEL главного файла конфигурации). Теперь появились новые параметры SCREEN_01, SCREEN_02 и т.д. Значение этих параметров означает режим, который будет использоваться для конкретного виртуального экрана бездисковой станции. Доступными значениями являются:

- startx

Означает использование традиционного для X-терминала графического режима. Данный параметр соответствует RUNLEVEL = 5 для LTSP версий 3.x;

- telnet

Запускает на терминале Telnet-сессию. Для LTSP версий 3.x это аналогично параметру RUNLEVEL = 4;

- shell

Подразумевает текстовый режим работы терминала, который применяется, как правило, только в целях администрирования для проведения диагностики работы X-терминала. Данный параметр соответствует RUNLEVEL = 3 для LTSP версий 3.x;

- rdesktop

Запускает на терминале единственное приложение rdesktop для работы в дистанционном режиме с терминальным сервером Windows. Идея в использовании данного режима заключается в том, что для работы в полно-экранной Windows-сессии пользователю незачем регистрироваться на сервере под управлением Linux.

Сценарии для работы на виртуальных экранах X-терминала располагаются в каталоге /opt/ltsp/i386/etc/screen.d, и пользователю ничего не стоит создать свой собственный сценарий для выполнения специфических функций бездисковыми рабочими станциями. Такую гибкость в настройке, по моему, можно только приветствовать.

Новое в синтаксисе файла /opt/ltsp/i386/etc/lts.conf

Программа читает информацию из главного конфигурационного файла проекта LTSP: /opt/ltsp/i386/etc/lts.conf. В версии 4.x ее поведение было значительно расширено, в основном благодаря новому ключевому слову "LIKE". Использование этого слова означает, что при описании конфигурации X-терминала, теперь можно копировать данные из другого раздела файла /opt/ltsp/i386/etc/lts.conf. Например, представьте, что у вас есть 3 совершенно одинаковых компьютера, которые вы используете как X-терминалы. Описываем один из них (например, с именем xt_first), а для других попросту копируем конфигурацию:

```
[xt_first]
        X_MOUSE_DEVICE   = /dev/ttyS0
        X_MOUSE_PROTOCOL = Microsoft

[xt_second] LIKE = xt_first

[xt_third]
        LIKE = xt_first
```

Если усложнить пример, и представить, что к одному из X-терминалов подключен принтер к параллельному порту (например, к терминалу с именем tx_second), то теперь фрагмент файла /opt/ltsp/i386/etc/lts.conf будет выглядеть так:

```
[xt_first]
        X_MOUSE_DEVICE   = /dev/ttyS0
        X_MOUSE_PROTOCOL = Microsoft
```

```
[xt_second]
        LIKE = xt_first
        PRINTER_0_DEVICE   = /dev/lp0
        PRINTER_0_TYPE     = P

[xt_third]
        LIKE = xt_first
```

С появлением этого нововведения файл /opt/ltsp/i386/etc/lts.conf стал гораздо проще для чтения и компактнее в размерах.

Новый демон ltspinfod

В четвертой версии проекта LTSP в рабочей среде X-терминалов появился новый демон: ltspinfod. Теперь процесс на сервере может обратиться к этому демону, работающему на X-терминале, и получить в ответ нужную информацию о состоянии бездискового компьютера. Используется этот демон такими сервисными вспомогательными службами как, например, поддержка звука. Теперь сервер может обратиться к этому демону, чтобы определить включена ли для X-терминал поддержка звука и какой именно звуковой демон используется.

Локальный запуск приложений

Теперь в LTSP, начиная с версии 4.0, значительно больше внимания уделяется использованию локального запуска приложений. Для запуска приложений на X-терминале используется утилита ssh, вместо rsh, как это было раньше. Кроме более простого использования, утилиту ssh отличает еще и на порядок более высокий уровень безопасности. Хотя, если присмотреться повнимательнее, то в локальном запуске приложений есть потенциальные бреши в системе безопасности. Так, при использовании ssh используются два ключа: личный (private key) и общий (public key), и чтобы сделать доступ действительно защищенным, необходимо, чтобы личный ключ хранился на локальном устройстве X-терминала, таком как флоппи-дискета, жесткий диск или USB-накопитель. Сейчас такой возможности не предусмотрено и личный ключ как и общий хранится на сервере, доступ к которому пользователи

получают через NFS. Правда, авторы LTSP обещали усилить методы обеспечения безопасности в будущих версиях данного проекта.

Для построения пар ключей для X-терминалов необходимо выполнить такие команды:

ssh-keygen -q -t rsa1 -f /opt/ltsp/i386/etc/ssh/ssh_host_key -C " -N "

ssh-keygen -q -t rsa -f /opt/ltsp/i386/etc/ssh/ssh_host_rsa_key -C " -N "

ssh-keygen -q -t dsa -f /opt/ltsp/i386/etc/ssh/ssh_host_dsa_key -C " -N "

После этого необходимо убедиться, что содержимое файла /etc/ssh/ssh_known_hosts примерно такое:

ws001 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAxFCM2eZU7P3HvEOMYhAFUiwE...

ws002 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAxFCM2eZU7P3HvEOMYhAFUiwE...

ws003 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAxFCM2eZU7P3HvEOMYhAFUiwE...

...

После этого, если вы правильно настроили NIS-сервер, можно добавлять строки поддержки запуска локальных приложений в раздел файла /opt/ltsp/i386/etc/lts.conf:

LOCAL_APPS = Y

NIS_DOMAIN = ltsp

После перезагрузки X-терминала он готов запускать локальные приложения, например, такой командой (пример для X-терминала с доменным именем xtlocal выполняется локальная команда free):

ssh xtlocal free

Для приложений, требующих X-Windows, необходимо несколько усложнить синтаксис команды (добавить переменную окружения DISPLAY=:0.0):

ssh xtlocal env DISPLAY=:0.0 /usr/local/Mozilla/mozilla

Исходный код проекта LTSP

LTSP теперь свободно доступен не только в виде скомпилированных программных компонентов, но и в исходном коде. Наверное, это должно порадовать тех пользователей, которые любят проверять исходный код программного продукта, а также вносить изменения в его поведение. Для параноиков безопасности это еще один шанс убедиться в том, что данная программа не имеет в себе скрытых, недокументированных возможностей, которые могут ослабить информационную защиту.

Среда компиляции программ для LTSP

В проекте LTSP версии 4.x появилось и такое понятие как LBE (The LTSP Build Environment – среда компиляции LTSP). Данная среда позволяет очень просто осуществлять сборку программ для локального запуска на X-терминалах. Правда, пока осуществлять сборку можно только для аппаратной архитектуры x86. Но в планах создателей LTSP расширить возможности данной среды на работу на всех аппаратных платформах, на которых может работать операционная система Linux.

Доступ к локальным устройствам

Теперь доступ к локальным устройствам хранения данных стал значительно проще. Для этих целей есть специальный пакет в составе LTSP: ltsp_localdev. В своей работе данному пакету требуются установленные на сервере утилита монтирования smbmount и демон автоматического монтирования autofs.

Некоторым пользователям придется отредактировать настройки демона автоматического монтирования. В частности, раскомментировать строку файла /etc/auto.master, содержащую "/misc /etc/auto.misc –timeout=60":

```
$ cat /etc/auto.master
# $Id: auto.master,v 1.2 1997/10/06 21:52:03 hpa Exp $
# Sample auto.master file
# Format of this file:
# mountpoint map options
# For details of the format look at autofs(8).
# /misc /etc/auto.misc  --timeout=60
```

Для того, чтобы воспользоваться доступом к компакт-диску X-терминала и его флоппи-дисководу, в параметрах файла /opt/ltsp/i386/etc/lts.conf необходимо добавить такие строки (пример, для X-терминала с именем xtld):

```
[xtld]
        LOCAL_DEVICE_01    = /dev/hdc:cdrom
        LOCAL_DEVICE_02    = /dev/fd0:floppy
```

После этого вы должны добавить такие строки в файл /etc/auto.misc на сервере X-терминалов:

```
xtldcd -fstype=smbfs,workgroup=LTSP,guest ://xtld/cdrom
xtldfl -fstype=smbfs,workgroup=LTSP,fmask=666,dmask=777,guest,username=nobody,rw ://xtld/floppy
```

Теперь, работая в вашем любимом файловом менеджере и обращаясь к каталогам /misc/xtldcd и /misc/xtldfl, вы будете лицезреть содержимое локального компакт-диска и флоппи-дискеты соответственно. Также просто осуществляется доступ к внешним флоппи-дисководам и приводам для чтения компакт-дисков с интерфейсом USB. Для этого достаточно загрузить нужные модули ядра и немного по-другому обратиться к символьным устройствам:

```
[xtld_usb]
        MODULE_01          = usbcore
        MODULE_02          = usb-uhci
        MODULE_03          = usb-storage
        LOCAL_DEVICE_01    = /dev/sr0:cdrom
        LOCAL_DEVICE_02    = /dev/sda1:floppy
```

Анализируя все вышесказанное, можно сделать вывод, что изменений по отношению к версиям 3.x довольно много, но все они только расширяют функциональные возможности проекта LTSP. Следует еще раз поблагодарить его авторов за то, что они смогли создать действительно качественный продукт, который уже не первый год показывает стабильное развитие.

Спасибо за прочтение моей книги. Удачи