

Network Operating Systems-I

DCAP602



L OVELY
P ROFESSIONAL
U NIVERSITY



NETWORK OPERATING SYSTEMS-I

Copyright © 2011 Anindita Hazra
All rights reserved

Produced & Printed by
EXCEL BOOKS PRIVATE LIMITED
A-45, Naraina, Phase-I,
New Delhi-110028
for
Lovely Professional University
Phagwara

SYLLABUS

Network Operating Systems-I

Objectives:

- Understand the concept of Network Operating System.
- Understand Network hardware.
- Understand configuration and security issues of Network Operating System.
- Install and configure network servers like SMTP and FTP servers
- Familiarization with installation and administration of operating system

S. No.	Description
1	Network Operating System: RedHat Linux, Installing RedHat Linux. Preparing for installation. Booting from CD. Graphical Installation Launch. Setting disk partition levels. Setting Boot Loader, First Boot. Creation of User Account.
2	RedHat Linux Basics: Working with Desktop. Using Terminal Emulator. File System Hierarchy. Configuring Desktop: working With Desktop Control Center. Understanding Run Levels. Managing Users.
3	Connecting to Internet: Network Configuration Tool. Connecting to LAN. DNS.
4	Installing Software: RPM. Meaning, RPM Management Tool. Adding & Removing Packages. Querying RPM Packages.
5	Shell: Different types of Shells. Common Shell Commands. File System Commands. Environmental Variables.
6	File System: What is File System. Anatomy of File System. File Permissions and Directories permissions. File Search Utilities.
7	User Accounts: Super User Vs. Normal User. RedHat User Manager. Creating Groups.
8	Server Role: Linux as Web Server. Apache Web Server. Installing Apache. Starting Apache. Configuring Web server. Setting up First Web Page.
9	FTP Server: Meaning, FTP Protocol. Installing vsftpd FTP Server. Starting FTP server. Testing FTP server. Using FTP server. Using FTP Client to Test Anonymous Read Access.
10	File Server: Overview of Samba Server. Installing SAMBA server. Starting and Stopping the SAMBA server. SAMBA configuration with SWAT. Starting SWAT Service. Adding SAMBA User. Creating and Configuring SAMBA Share.

CONTENTS

Unit 1:	Introduction to Network Operating System	1
Unit 2:	RedHat Linux Basics	28
Unit 3:	File System Hierarchy	52
Unit 4:	Configuring Desktop	65
Unit 5:	Connecting to Internet	91
Unit 6:	Domain Name System	99
Unit 7:	Installing Software	128
Unit 8:	Shell	144
Unit 9:	File System Commands	154
Unit 10:	File System	164
Unit 11:	User Accounts	201
Unit 12:	Server Role: Linux as Web Server	222
Unit 13:	FTP Server	241
Unit 14:	File Server	260

Unit 1: Introduction to Network Operating System

Notes

CONTENTS

Objectives

Introduction

- 1.1 RedHat Linux
- 1.2 Installing RedHat Linux
- 1.3 Preparing for Installation
 - 1.3.1 Hardware
 - 1.3.2 Hardware Compatibility Lists
 - 1.3.3 Server Design
 - 1.3.4 Dual-booting Issues
- 1.4 Booting from CD
- 1.5 Graphical Installation Launch
- 1.6 Creating User Accounts
- 1.7 Summary
- 1.8 Keywords
- 1.9 Self Assessment
- 1.10 Review Questions
- 1.11 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss RedHat linux
- Understand installing RedHat linux
- Understand preparing RedHat linux for installation
- Discuss the concept of booting from CD
- Discuss graphical installation launch
- Understand setting disk partition levels
- Discuss setting boot loader
- Understand the concept of the first boot
- Discuss creation of User Account

Introduction

Installation routines and hardware support in Linux at the time were much less advanced than they are today; Red Hat was still at a relatively early stage in its evolution, Mandriva had yet to be created, and SuSE was just coming out from under the shadow of Slackware.

This unit explains how to perform a custom installation of Red Hat Linux from the CD-ROM, using the graphical, mouse-based installation program.

1.1 RedHat Linux

There is quite a variety of Linux distributions from which to choose from. Each distribution offers the same base Linux kernel and system tools, but differ on installation method and bundled applications. Each distribution has its own advantages as well as disadvantages, so it is wise to spend a bit of time researching which features are available in a given distribution before deciding on one.

The installation of a Linux system requires a little more up-front research than does a Windows installation. As many Linux device drivers are created through community-based reverse-engineering, rather than by those devices' manufacturers, it's important to check a number of hardware compatibility lists prior to commencing the installation. This will help you ensure that drivers exist for the devices on your server.

Linux support can take many forms, the most popular being Web-based lists and forums. This approach truly represents the spirit of community in the open source world, where user experience is relied upon to provide solutions to Linux issues. All commercial Linux distributors provide some level of paid support, though the support period may vary widely from one distributor to another.

Linux systems can be installed with a full complement of graphical tools, or as a minimal text-based system. The installers follow suit, providing options to complete an installation from a graphical environment, or from a purely text-based environment.

Unlike Windows systems, the desktop environment is not inextricably bound to the operating system kernel code. Instead, the X Windows and desktop management systems are distinct systems that run in their own space. This feature of Linux allows for the creation of a fully operational, text-based system, which boasts a very small installation code base. However, most users will opt for a graphical system based on X Windows and any of a number of desktop managers.



Did u know? Do Linux support can take many forms, the most popular being Web-based lists and forums?

1.2 Installing RedHat Linux

To begin the installation, put the first installation CD in the CD-ROM drive and reboot the machine. If your machine is configured to boot from the CD-ROM, you'll see the screen shown in Figure 1.1, when the machine starts.

Figure 1.1: The Initial Fedora Installation Screen



The initial installation offers several options. You can choose to install in graphical mode by hitting Enter, or in text mode by typing linux text at the boot: prompt. Either way, the first thing the installer will do is offer to check the installation media for you. This is a good way to determine if your installation CDs have been tampered with, or have become corrupted. The process will take a little while, but I'd recommend that you do run this test.

Like any operating system, Linux requires a minimal set of hardware drivers during the installation. After testing the installation media, you'll see lots of text scrolling down the screen – this is the initial hardware probing process in action. Red Hat helped pioneer the development of graphical Linux installers with Anaconda, Red Hat's installation program. It includes a highly accurate probing and testing mechanism that makes the rest of the installation routine quite painless.

Once all this media testing and hardware probing is done, you'll finally see the Welcome to Fedora Core screen. Click the Next button to get started.



Task

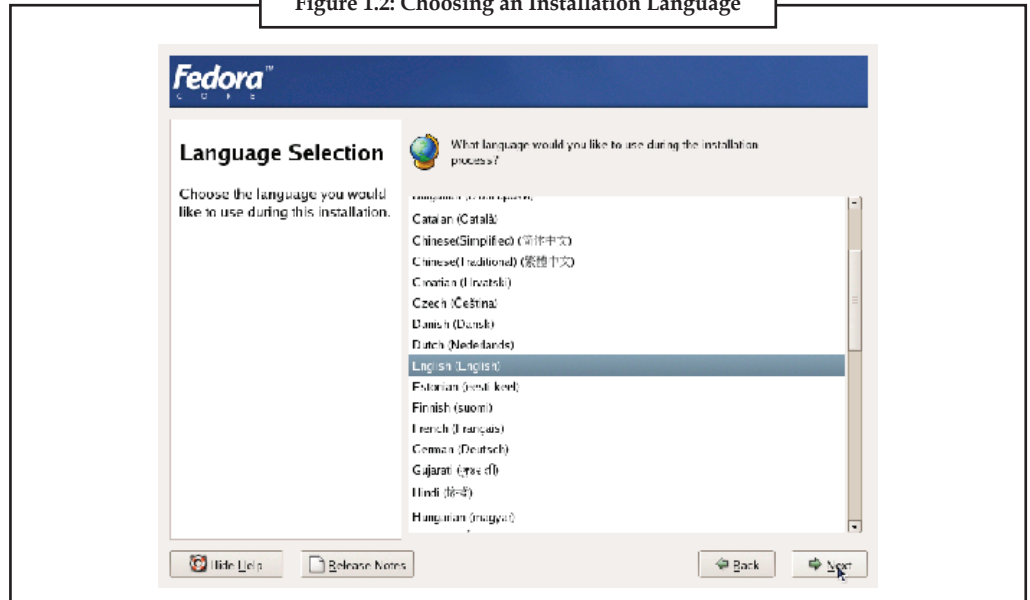
Discuss the first step for installing the RedHat Linux.

Notes

Selecting your Language

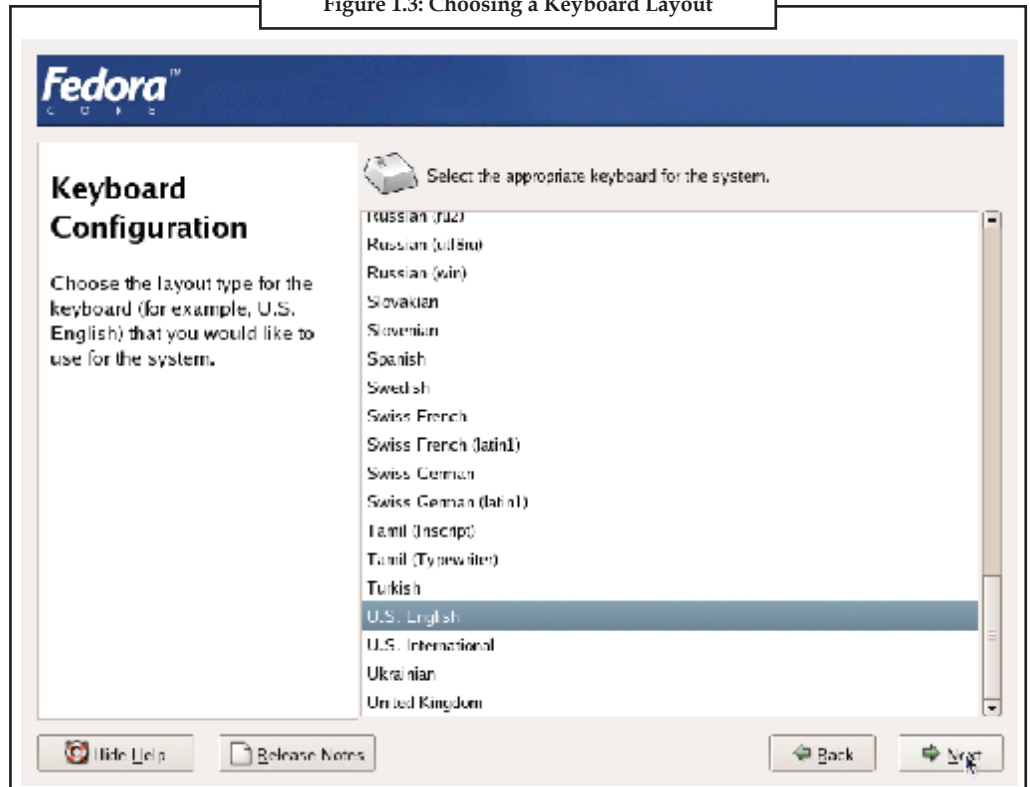
Fedora is truly an international operating system: the installation screens are available in more than 30 languages. Select your native tongue from the Language Selection screen shown in Figure 1.2, and click Next.

Figure 1.2: Choosing an Installation Language



The number of keyboard languages available to Fedora is similar to the number of languages available through the installation screens. Select the language of your keyboard from the screen shown in Figure 1.3.

Figure 1.3: Choosing a Keyboard Layout

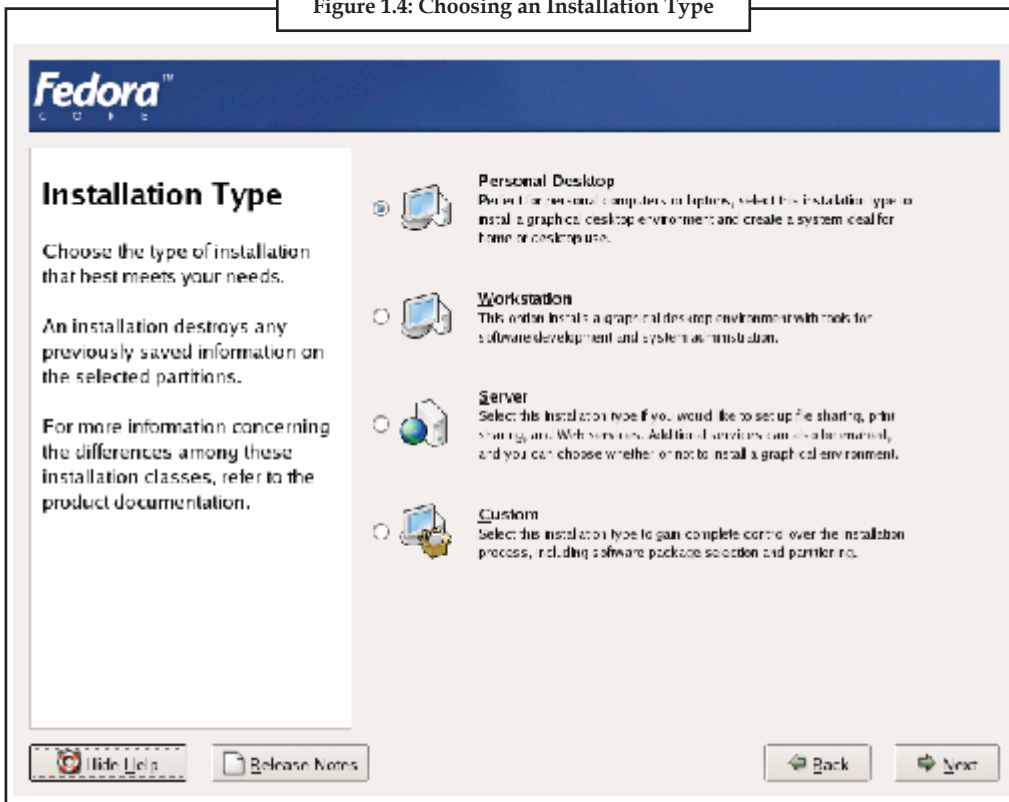


Installation Types

Notes

The Fedora installer offers three specialized installation types: Personal Desktop for home or office use, Workstation for development or system administration work, and Server for file, print and Web server use. There's also a Custom option if you'd like to take complete control over the way your system is configured. As we're setting up a Web server, select the Server option from the Installation Type screen shown in Figure 1.4, before clicking Next.

Figure 1.4: Choosing an Installation Type



Did u know? Is Fedora is truly an international operating system: the installation screens are available in more than 30 languages?

Setting Disk Partitioning Levels

The Fedora installer offers two partitioning methods – automatic and manual – as shown in Figure 1.5.

Notes


Figure 1.5: Selecting a Partitioning Method



Automatic partitioning creates three partitions:

1. The /boot partition is the home of the kernel: the program at the very heart of Linux. Fedora recommends a /boot partition of no less than 100MB, though you'll seldom need this much.
2. The swap partition is used as a fallback for memory when all of the system memory is in use.
3. The / partition contains everything that isn't on its own partition.

Partitions in Linux appear differently than those in Windows. Linux partitions don't use the drive letter designations, such as C:, which you may already be used to. The primary partition on Linux is labeled / (you'll see how this fits into the overall partitioning layout later). Other common partitions on a system include /boot (contains the kernel and boot loader), /home (contains user-specific files), and /var (contains program configuration and variable data). These labels are called mount points. It's possible to organize your system so that it's spread over multiple partitions; for example, it's quite common to put /var (where data, including such things as MySQL databases and Websites, live) on a separate partition. However, automatic partitioning makes things simpler, and spreading your data across different partitions doesn't achieve very much. Some administrators strongly recommend it, but the Fedora rescue CD (also downloadable as an ISO image from the Fedora Website) will help you avoid most problems that might have been aided by splitting the data across different partitions in the past. Therefore, the default partitioning setup is usually sufficient.



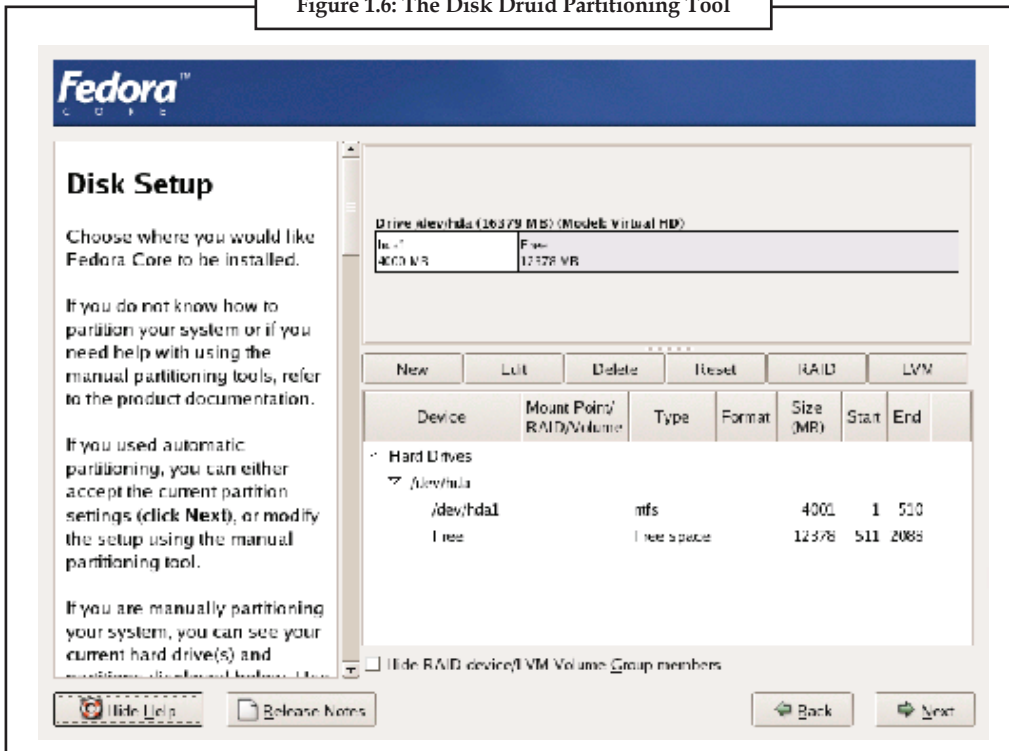
Task Give the three partitions of automatic partitioning.

Using Disk Druid

Notes

Fedora also offers Disk Druid, a graphical partitioning tool. If you'd prefer a scheme other than the default, you'll need to use Disk Druid during the installation process. Disk Druid presents both graphical and textual representations of the partition table on your machine. To select a partition, click on the graphical drive representation (shown in Figure 1.6), or on the textual representation. In either case, you can add, edit, or delete partitions by clicking on the appropriate tool bar buttons.

Figure 1.6: The Disk Druid Partitioning Tool



If the system onto which you're installing Linux has a previous installation of Windows (or some other operating system), you might want to manually delete the partition that contained Windows. Also, if you don't see any space marked as "Free" in the diagram at the top of the screen, you'll need to delete something to make room for Fedora. To do this, select the partition to delete, and click the Delete button.



Note

Disk Druid presents both graphical and textual representations of the partition table on your machine.

Deleting Partitions

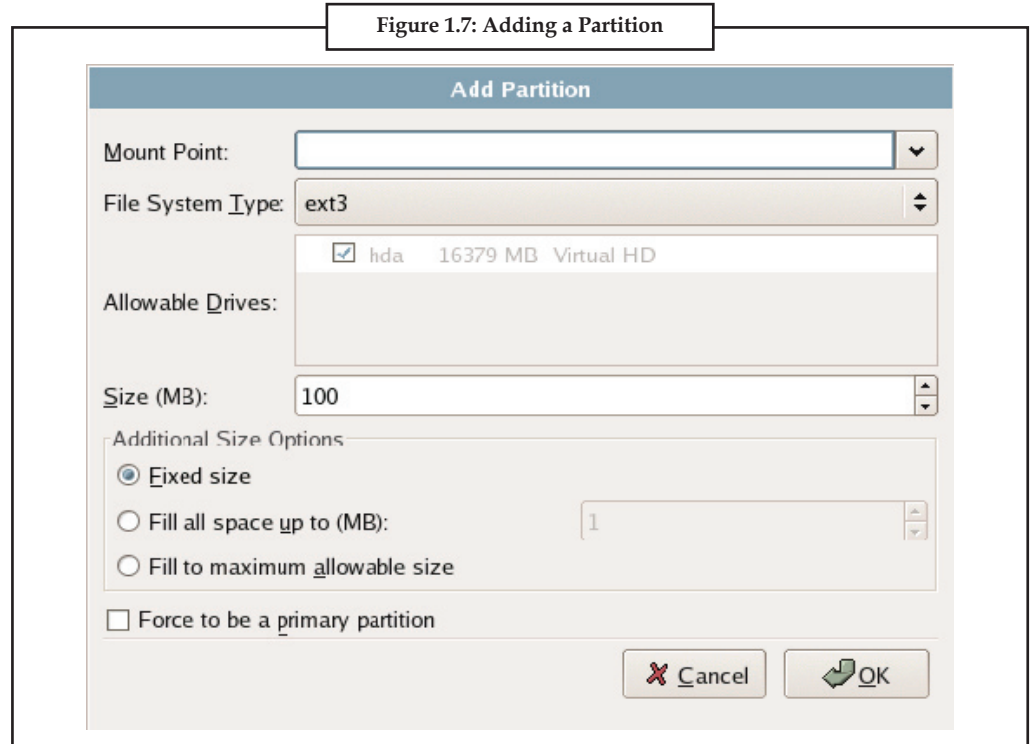
Once you delete a partition, there's no way to get back the data that was on it. (Well, there's no easy way. Advanced recovery tools do exist.) Delete with care.

Notes

Correcting an Accidental Deletion

If you accidentally mark a partition for deletion, or make some other mistake, you can set everything back to its original state by clicking the Reset button. The changes you make to the partitions won't actually take effect until later in the installation procedure.

Click the New button to open the Add Partition dialog shown in Figure 1.7.



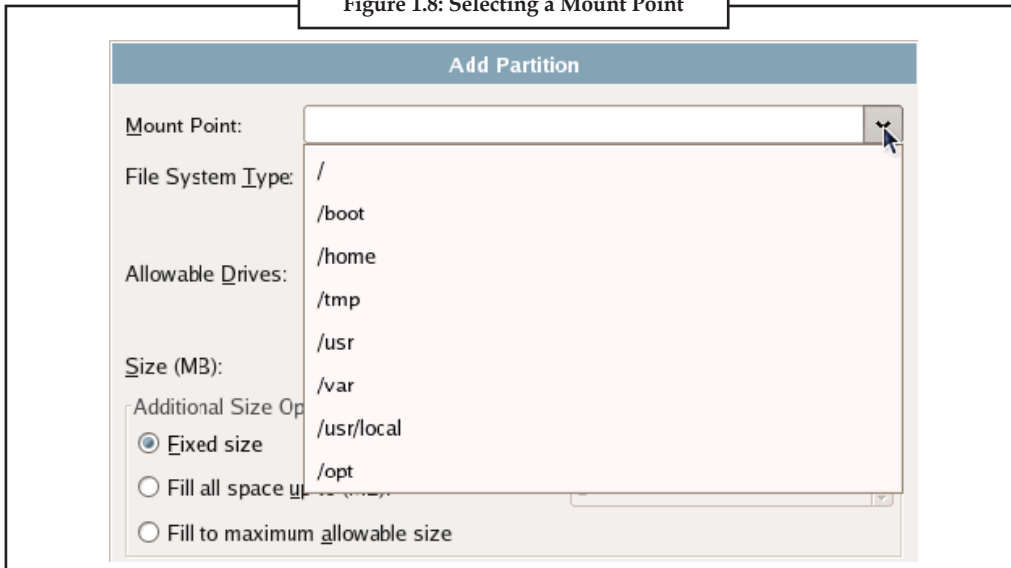
From here, you can designate the mount point, the filesystem type, and the partition's size in megabytes. The window also offers further size options, including the ability to create a partition with all remaining space on the drive.

Selecting the Mount Point drop-down will display all common partition labels (mount points) available for your server, as shown in Figure 1.8, "Selecting a mount point."; alternatively, you can enter the mount point label manually. Bear in mind that these are the most common mount points, and are familiar to all Linux system administrators.



Caution Creating a custom mount point might confuse other administrators of your server.

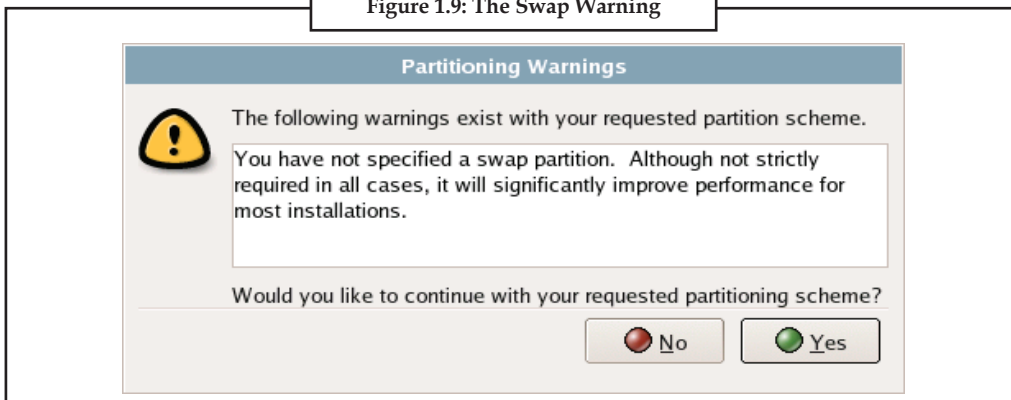
Figure 1.8: Selecting a Mount Point



Once you have created a partition, you can edit it by selecting the partition, then clicking the Edit button, which will give you almost the same options as the Add Partition dialog.

If you try to proceed past the Disk Setup screen without creating a swap partition, you'll receive the warning shown in Figure 1.9. A swap partition in Linux serves much the same purpose as virtual memory in Windows: when the system's memory becomes full, part of the data in memory is written to the swap partition, freeing up that memory space. When the data that was written to the swap partition is needed again, it is read back into memory. To create a swap partition, click the Add button and select swap as the File System Type.

Figure 1.9: The Swap Warning



Note

If you try to proceed past the Disk Setup screen without creating a swap partition, you'll receive the warning shown above.

Swap Space

A good rule of thumb to use when creating swap space on your Linux machine is to create one and a half times the size of the machine's physical memory.

Notes



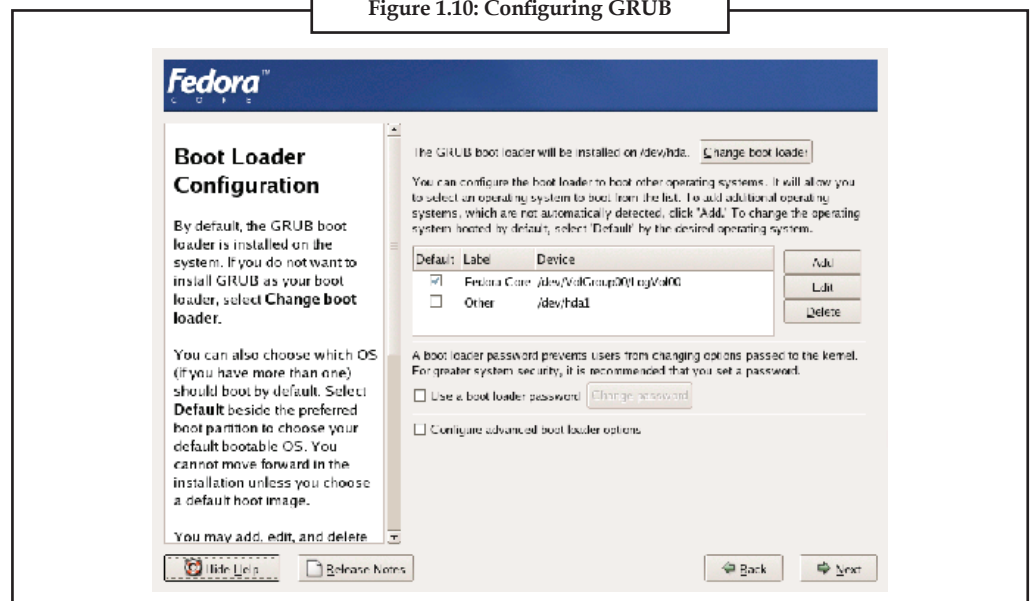
Example: If you have 1GB of physical memory, create a 1.5GB swap partition.

Setting Boot Loader

The GRUB Boot Loader

If you have decided to go with a dual-boot install, you'll need to set up the GRUB boot loader. GRUB is a program that will let you select from a list of installed operating systems, then makes the computer start up the selected OS. As Figure 1.10 shows, it's pretty easy to set up. Note that you should set a boot loader password to prevent unauthorized users from gaining access to the kernel's startup parameters.

Figure 1.10: Configuring GRUB



First Boot

firstboot is the program that runs on the first boot of a Fedora or Red Hat Enterprise Linux system that allows you to configure more things than the installer allows.

It also has a mode known as *reconfig* mode to let you reconfigure user settings that were originally set in the installer. This is generally useful to IHVs shipping hardware to an end user so that the end user can do things such as select their language, set their root password and configure their network. To start this mode, touch `/etc/reconfigSys` on your system.

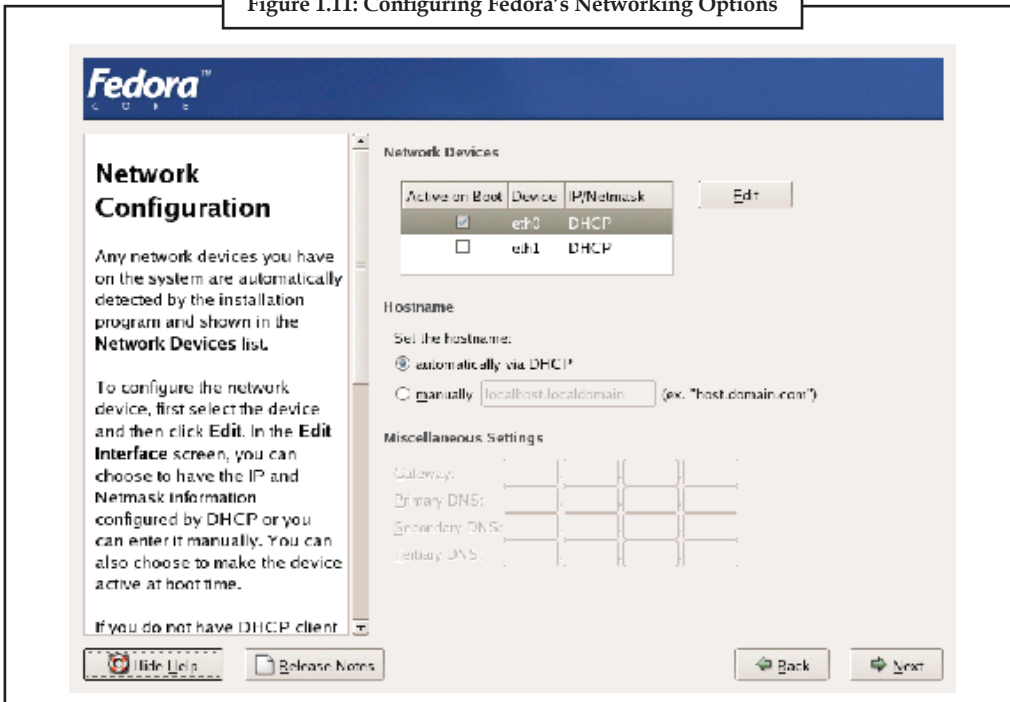
firstboot has a modular architecture such that each screen is a separate python module. This allows firstboot to be a framework that requires no knowledge of the underlying modules. The primary benefit of this is that additional steps can be added without having to modify firstboot itself. The Modules page describes how to create a firstboot module.

Networking

After you've set up all of your partitions, you'll be offered the networking options shown in Figure 1.11, "Configuring Fedora's networking options." Existing Ethernet cards within the machine will be denoted as ethn; if the machine has only one network card, it will be called

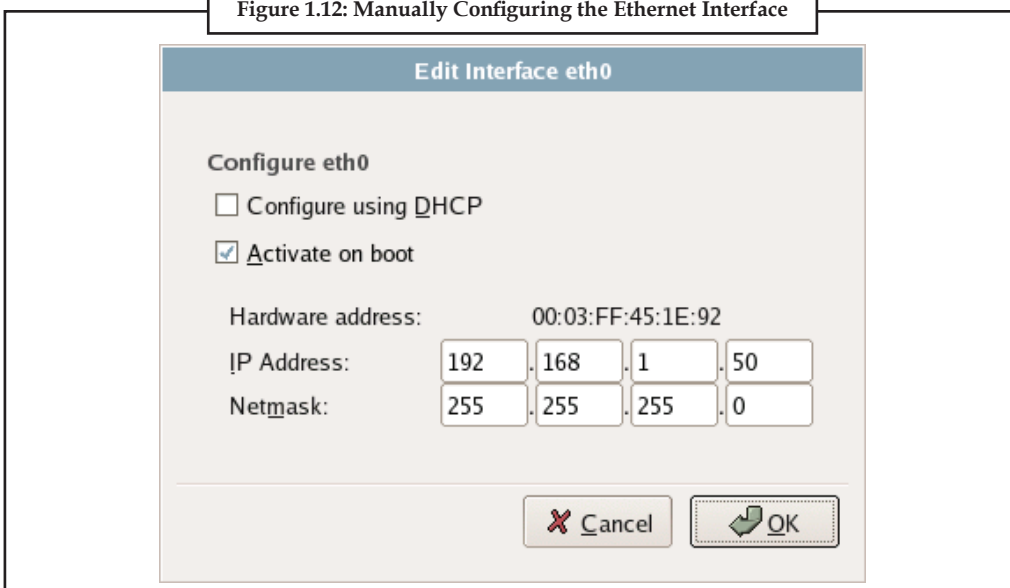
eth0. The default configuration will be something like that displayed in Figure 1.11. The first network connection (usually eth0) will be made active, and will be automatically configured via DHCP. Dynamic Host Configuration Protocol (DHCP) will be used to auto-detect your network settings to enable you to connect to the Internet, or to a private network.) If the machine is on an internal network, you'll probably be able to just leave this as the default. For a Web server that's connected directly to the Internet, you'll need to manually configure your static IP address and manually-configured gateway, DNS, and hostname. In this case, your ISP will be able to provide you with the IP address, gateway, and other details to use.

Figure 1.11: Configuring Fedora's Networking Options



Clicking the Edit button in the Network Configuration screen will display the Edit Interface window shown in Figure 1.12. Here, you can make custom configuration adjustments such as giving the server a static IP address.

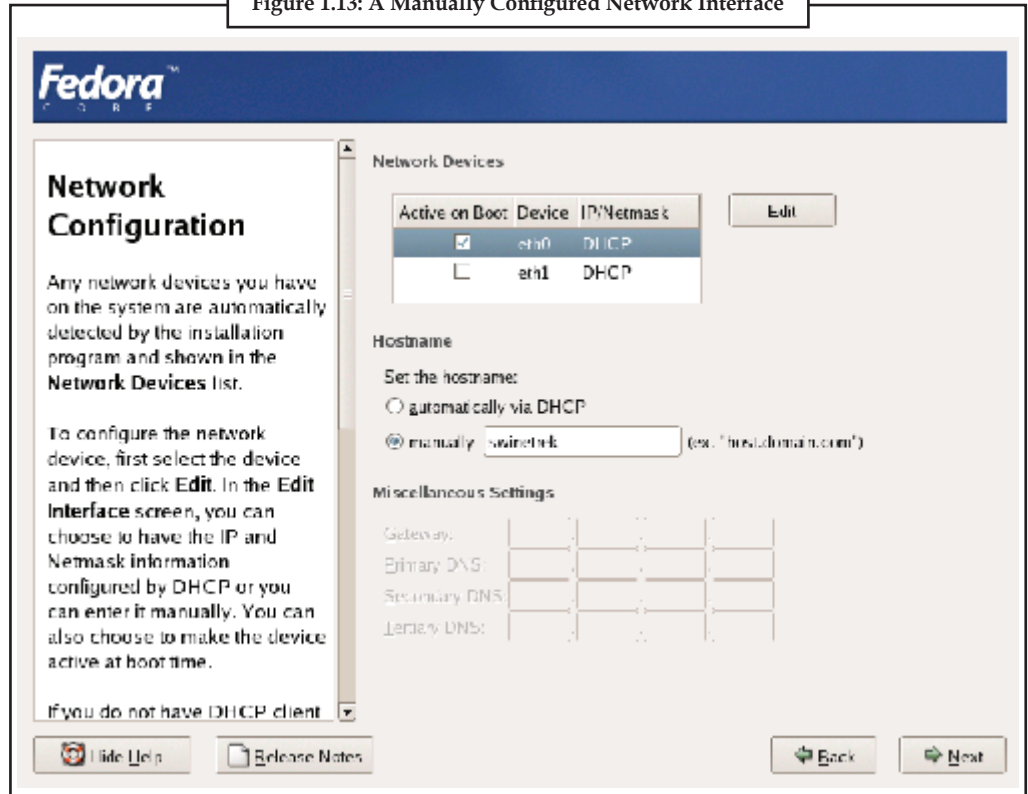
Figure 1.12: Manually Configuring the Ethernet Interface



Notes

When the network device settings have been configured from the previous screen, you're free to configure the hostname, gateway and DNS settings. Figure 1.13 shows a network device configured primarily for internal use.

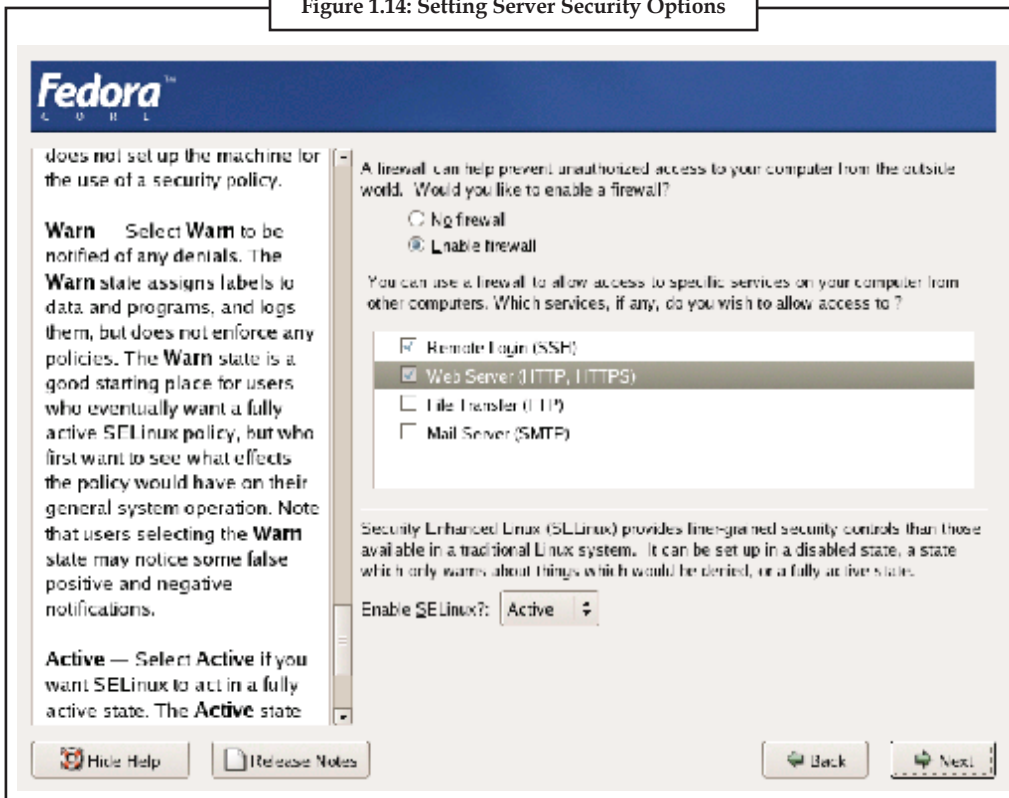
Figure 1.13: A Manually Configured Network Interface



Network Security

The Fedora Core distribution – and many of the other major distributions of Linux –strive to make configuring your network security as easy as possible. By default, Fedora turns on a firewall that blocks all traffic coming in from the network. To customize the firewall, simply select the services you want to run on this machine; alternatively, you can simply disable the firewall, which will leave the machine open and vulnerable to hacker attacks. You can also choose to enable Security Enhanced Linux (SELinux), which can help to minimize any damage caused if hackers gain control of parts of the system. Note that SELinux should not be considered an alternative to a firewall – neither the firewall, nor SELinux, makes your system completely secure, so it's best to enable them both. For our purposes, you should only allow Remote Login and Web Server traffic through the firewall, and set Enable SELinux? to Active, as illustrated in Figure 1.14.

Figure 1.14: Setting Server Security Options



Telnet and FTP Security

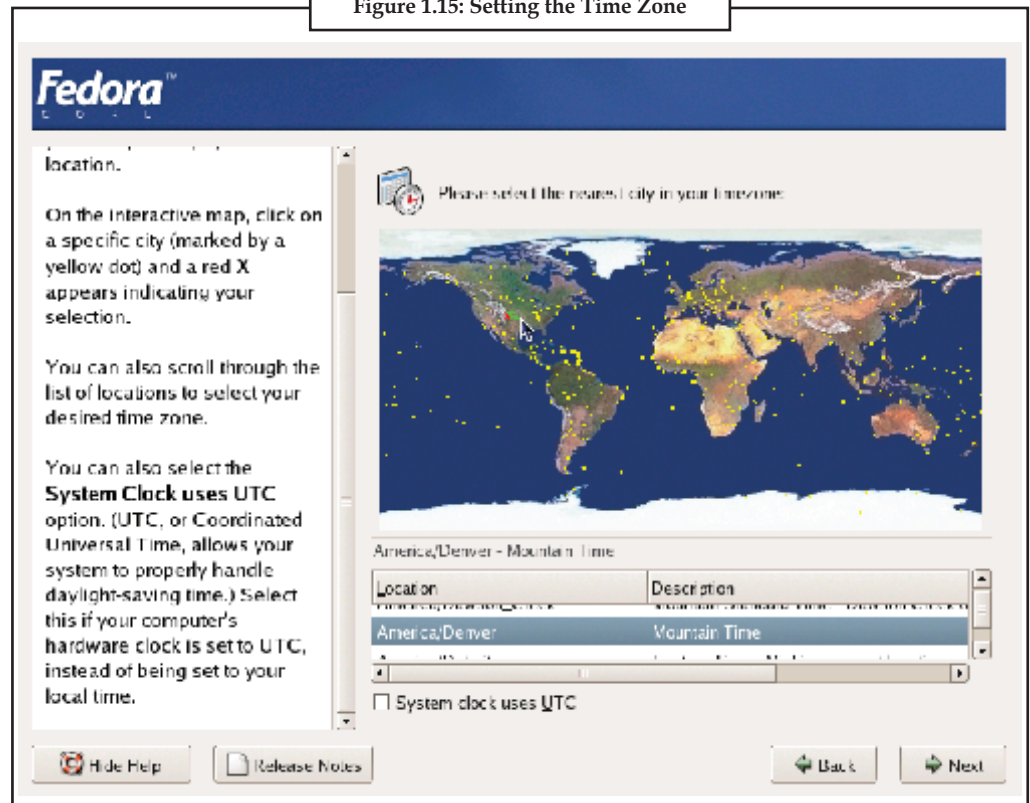
Though they're shown as options in the Fedora security configuration screens, both telnet and FTP are widely recognized as insecure protocols. SSH is a much more secure option than telnet for accessing remote machines, as SFTP is a more secure option than FTP for transferring files. If an FTP capability is required, it's recommended that it be set up on a different server that's isolated as much as possible from the rest of the network.

Setting the Time Zone

Fedora offers two options for setting the time zone for your server. You can roll the mouse over the metropolitan area that's closest to you, or you can select from an exhaustive list of cities. In either case, the chosen city will be highlighted on the map, as shown in Figure 1.15.

Notes

Figure 1.15: Setting the Time Zone



Setting up the Root User

All Linux systems have an administrative account, root. This account has access to everything on the computer; it's similar to the Administrator account in Windows systems. As the power of root in Linux is so broad, it's critical that you make accessing the root account as difficult as possible. Choose a secure password for the root account – one that consists of both upper and lowercase letters, as well as numbers and special characters – and enter it into the fields as shown in Figure 1.16. It is recommended that you record your root password somewhere and keep it safe: if you forget the password, it becomes very difficult to gain access to your machine should things go wrong.

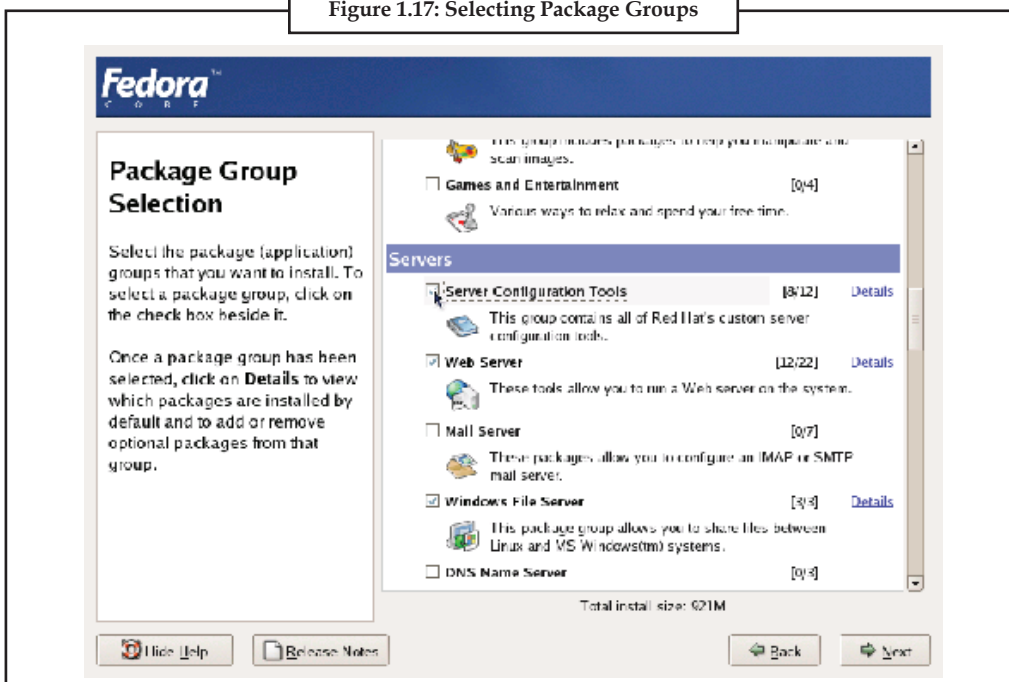
Figure 1.16: Setting the Root Password



Installing Software Packages

Previously, when you were asked to select an installation type (you selected from personal desktop, workstation, server, or custom), your selection determined which software package groups would be made available for selection in this screen. For your server installation, you'll see the full range of server software offered as part of the Fedora distribution, with a few nice extras thrown in. Select each of the package groups you want to install by clicking the appropriate check boxes, as shown in Figure 1.17.

Figure 1.17: Selecting Package Groups



Notes

Each package group contains a number of packages; you can see a list of these (similar to the one shown in Figure 1.18) by clicking the Details link that appears when the package group is checked. This list is made up of base packages – packages that are required for this package group – and optional packages, which you can choose to install as your needs dictate.

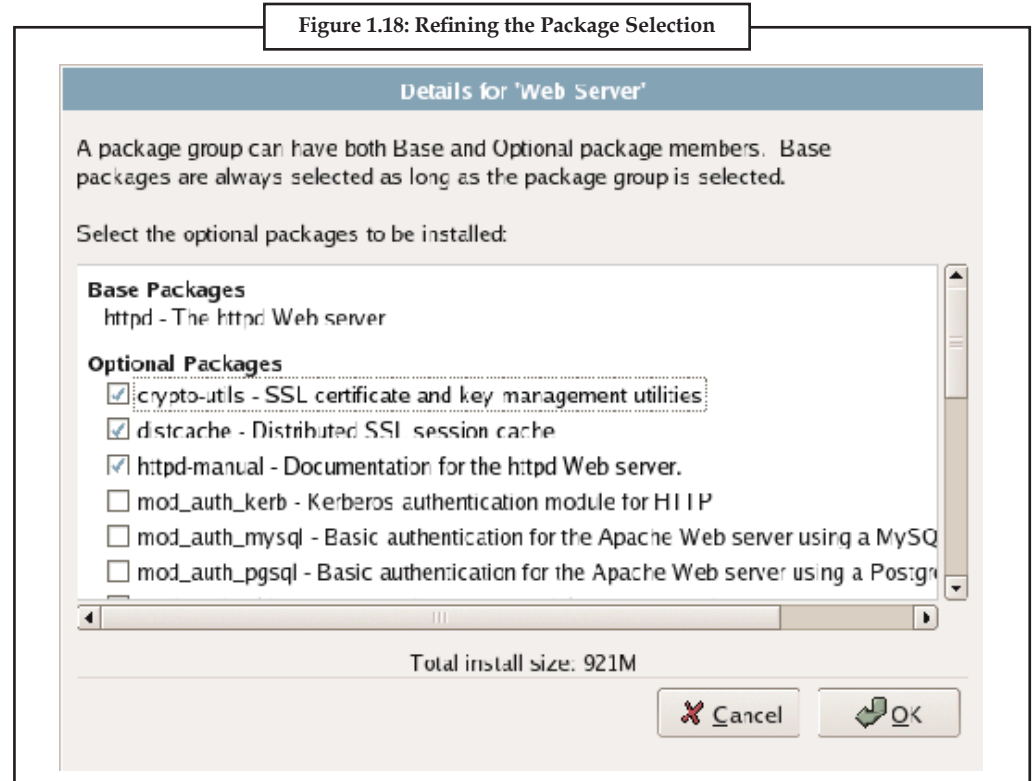


Figure 1.18: Refining the Package Selection

Through a long process of refinement, the Red Hat distributions have come to provide a full range of packages that meet nearly any common computing need. While it's a good goal to keep a server installation to a minimum, you may find that there are some packages you just can't do without. If you're using Linux for the first time, it's perfectly okay to accept the defaults; it's easy to add packages later if you realize that something else is required, and the defaults are carefully chosen by the Fedora team to cover the needs of most people.

1.3 Preparing for Installation

As Linux has gained market share within the server market, Linux driver development has improved markedly. Storage devices, RAID arrays, Ethernet cards—all have enjoyed increasing Linux driver development in the past few years.

In order to avoid the headache of missing drivers, it's important to do a little research before installing your Linux distribution. While it's unlikely that you'll have a problem with modern distributions, you'll still want to do the research just to avoid any hardware issues.

In order to be able to complete the installation procedure smoothly, you should collect certain information about your system before beginning the installation. Often the installation utility will be able to determine your system configuration automatically but when it fails to do so, you must be prepared to supply the needed information. Otherwise, you'll be forced to terminate the installation procedure, obtain the information, and restart the installation.

The following Table 1.1 specifies the configuration information you need. To obtain this information, you can consult your system documentation and the documentation for any devices installed by you. If your documentation is missing or incomplete, you may need to contact your hardware vendor or manufacturer. Alternatively, you may be able to find the needed information on the manufacturer's web site; use a search engine such as Yahoo! or Google to discover the URL of the web site.

Table 1.1: Configuration Information Needed to Install Linux

Device	Information needed
Hard Drive(s)	The number, size, and type of each hard drive Which hard drive is first, second, and so on Which adapter type (IDE or SCSI) is used by each drive For each IDE drive, whether or not the BIOS is set for LBA mode
RAM memory	The amount of installed RAM
CD-ROM Drive(s)	Which adapter type (IDE, SCSI, or other) is used by each drive For each drive using a non-IDE, non-SCSI adapter, the make and model of the drive
SCSI Adapter (if any)	The make and model of the card
Network Adapter (if any)	The make and model of the card
Mouse	The type (serial, PS/2, or bus) The protocol (Microsoft, Logitech, MouseMan, etc.) The number of buttons For a serial mouse, the serial port to which it's connected
Video Adapter	The make and model of the card The amount of video RAM

To obtain the needed information, you may need to examine your system's BIOS settings or open your system's case and examine the installed hardware. Consult your system documentation to learn how to do so.

1.3.1 Hardware

Linux supports a wide range of PC hardware; but not even Linux supports every known device and system. Your PC must meet certain minimum requirements in order to run Linux.

First, determine what kind of hardware you have. Prepare a checklist to assist you. Be as precise as possible, but don't get carried away. For example, if you have an Ethernet card, you need to know what kind (e.g., SMC-Ultra, 3Com 3C509, etc.), base I/O (e.g., io=0x300), interrupt (IRQ 10), but not the hardware address (00 00 a6 27 bf 3c). Not all information will be needed for your hardware. If you have Windows 95 or Windows NT running, you can copy the values from the system hardware device information screen. Otherwise, consult the hardware manuals or the hardware company's Web site.

Linux hardware requirements are modest, but picky. You do not need to have the most advanced and latest model PC to run Linux, but since the development of device drivers is primarily done by volunteers, you need to have devices in your PC for which device drivers have been developed by the Net community.

1.3.2 Hardware Compatibility Lists

Red Hat/Fedora

Red Hat's major product line is Red Hat Enterprise Linux (RHEL), which is mostly based on Red Hat's free software distribution, Fedora. Fedora is not actually maintained by Red Hat; it's maintained by the community of Fedora developers. However, Red Hat does a lot of work on Fedora, because that work flows into RHEL.

Red Hat's Hardware Catalog doesn't extend beyond RHEL to the Fedora releases, which is something that you'll need to remember when looking to the Red Hat site for Fedora support. The list provides information on CPUs, video cards, SCSI controllers, IDE controllers, network cards, modems, and sound cards.

SuSE

SuSE offers two lists: the Express Search and Extended Search. The difference between the two is that the Extended Search offers fields beyond Vendor, Device, and Category. In practice, you're likely only to need the Express Search.

Mandriva Linux

The Mandriva Linux Hardware Compatibility Database is a very comprehensive list of hardware that has been tested by the Mandriva Linux community.

General Linux

The Linux Hardware Compatibility HOWTO is perhaps the most comprehensive of the high-level Linux links. It was begun in 1997 and is updated as often as twice annually. It provides information on all device types and all major manufacturers.

Aside from providing interesting and useful user forums, LinuxQuestions.org also provides an outstanding list of Linux-compatible hardware. This is the most up-to-date of the high-level Linux lists, with updates appearing daily where applicable. While it's not as comprehensive as the HOWTO, the LinuxQuestions list is easily as important because of this timeliness.



Caution Linux Compatible provides both updated lists, and forums in which users can help other users resolve existing hardware issues.

1.3.3 Server Design

A server installation removes all existing partitions on all installed hard drives, so only choose server installation if you're sure you have nothing you want saved. This means that if you have Windows installed in ANY drive it will delete it and install Linux. As in the workstation installation it will partition the hard-drive(s) and install a variety of software packages, but it will not include many of the user-oriented packages present in the workstation installation.

In order to perform a server installation you will need at least 1.8 GBytes of free hard-disk space. No dual-booting will be set up since no other operating system will exist in the machine (remember that a server installation deletes ALL other operating systems). Therefore, unless you are using your machine solely as a server, it is suggested you to do a workstation installation and then add the server software you may need. This also allows preserving a prior Windows installation when you install Linux.

Use RedHat boot diskette(s) and insert the CD-ROM 1 in the drive. A basic Linux kernel will load and run the installation script. Select server as the installation class. The script, like in the workstation case, will try to detect most of your hardware, but will ask at least what monitor you have, mouse, and TCP/IP information to setup networking. Be sure to create a boot diskette for your machine during the installation - the script will prompt you to do so.

1.3.4 Dual-booting Issues

If you are building your dual-boot server on a new computer, be sure to install and configure Windows first. By default, Windows doesn't recognize any of the native Linux filesystems. But, there are third-party utilities that allow Windows to read the drives of a Linux installation on the same machine. If Linux is installed first, the Windows boot loader will take over and load Windows; Linux will be there, but you won't be able to boot into it. A Linux installation will cooperate with Windows and allow you to boot into both.

Linux provides a means to read the FAT32 (typically used by Windows 98 and ME) or NTFS (usually used by Windows NT, 2000, and XP) filesystems. In the case of FAT32, you'll also be able to write to the Windows partitions. If you're using an NTFS-based Windows installation, the files on the Windows partition will be read-only.

If you are installing Linux on a system that already contains a Windows operating system, it may be useful to purchase a nondestructive partition management tool, such as Partition Magic. This will allow you to move the partitions on your Windows system, creating room on the drive for the Linux installation, and preserving the data that already exists on the drive.

With the exception of these important points, the process of installing a dual-boot system is the same as a single OS installation.

1.4 Booting from CD

In order to install Linux, we must begin by booting the Linux kernel. This is accomplished in exactly the same manner as if you wanted to reload MS-DOS: we need a boot disk. But most distributions come only with a CD-ROM, and even if we had a running Linux system, the command to create boot disks for Linux is different than for MS-DOS. If you bought a new computer with a bootable CD-ROM, some distributions allow you to boot in this manner. But we'll go through the process of creating a boot disk for the rest of us.

The first step in getting Red Hat's distribution of Linux onto a system, you need to find a way of starting the installation program. The usual method of doing so is to create an installation disk, although if you are installing from CD-ROM, and your system's BIOS supports it, you should be able to boot directly into the installation program from the CD.

Otherwise, to create an installation diskette, you'll need to copy the "boot.img" (which is simply an image of an ext2-formatted Linux boot diskette with an additional installation program) onto a floppy diskette. The "boot.img" file can be obtained from the /images directory of the Red Hat CD-ROM disk, or downloaded via FTP from ftp://ftp.redhat.com in the /pub/redhat/redhat-6.1/i386/images directory (assuming you are installing Linux on an Intel box).

You can create the boot diskette either from a DOS or Windows system, or from an existing Linux or Unix system. For your destination diskette, you can use either an unformatted or a pre-formatted (for DOS) diskette - it makes no difference.

Under DOS: Assuming your CD-ROM is accessible as drive D:, you can type:

```
d:
cd \images
..\dosutils\rawrite
```

Notes

For the source file, enter “boot.img”. For the destination file, enter “a:” (assuming the diskette you are created is inserted into the A: drive). The “rawrite” program will then copy the “boot.img” file onto diskette.

Under Linux/Unix: Assuming the “boot.img” file is located in the current directory (you may need to mount the CD-ROM under /mnt/cdrom and find the file in /mnt/cdrom/images), you can type:

```
dd if=boot.img of=/dev/fd0
```

The “dd” utility will copy, as its input file (“if”), the “boot.img” file, onto the output file (“of”) /dev/fd0 (assuming your floppy drive is accessible from /dev/fd0).

Unless your Linux or Unix system allows write permissions to the floppy device, you may need to do this command as the superuser. (If you know the root password, type “su” to become the superuser, execute the “dd” command, and then type “exit” to return to normal user status).

With either of the above schemes, you should now have a bootable Red Hat installation diskette that you can use to install your new Red Hat Linux system.



Task

In order to install Linux, we must begin by booting the Linux kernel. Explain

1.5 Graphical Installation Launch

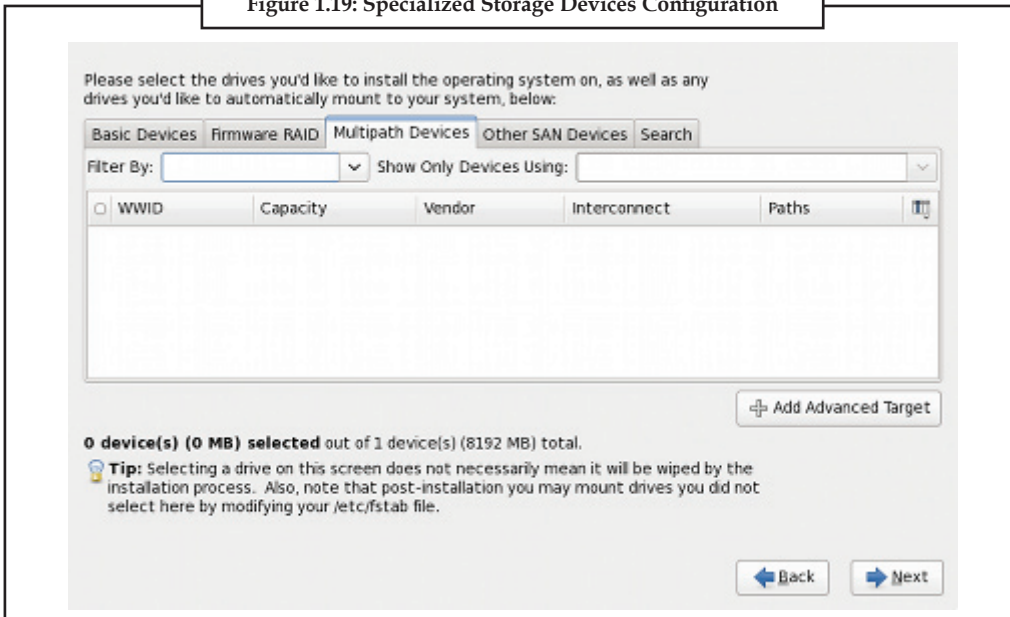
The Red Hat Enterprise Linux graphical installer steps the user through the major steps involved in preparing a system for installation. The Red Hat Enterprise Linux 6 installation graphical installer introduces major usability enhancements for disk partitioning and storage configuration.

The graphical installer now allows a user to choose basic storage devices or specialized storage devices. Basic Storage Devices typically do not need any additional configuration settings before the device is usable. A new interface has been implemented for configuring specialized storage devices. Firmware RAID devices, Fibre Channel over Ethernet (FCoE) devices, multipath devices, and other storage area network (SAN) devices can now be easily configured using the new interface.



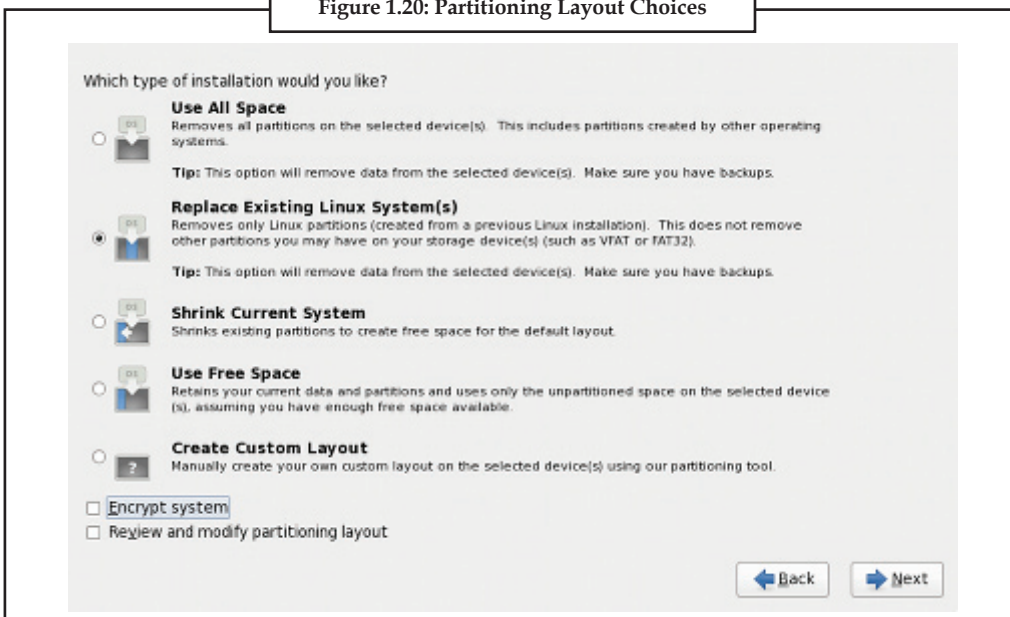
Did u know? Do you think Graphical installer introduces major usability enhancements for disk partitioning and storage configuration?

Figure 1.19: Specialized Storage Devices Configuration



The interface for choosing partitioning layouts has been enhanced, providing detailed descriptions and diagrams for each default partitioning layout.

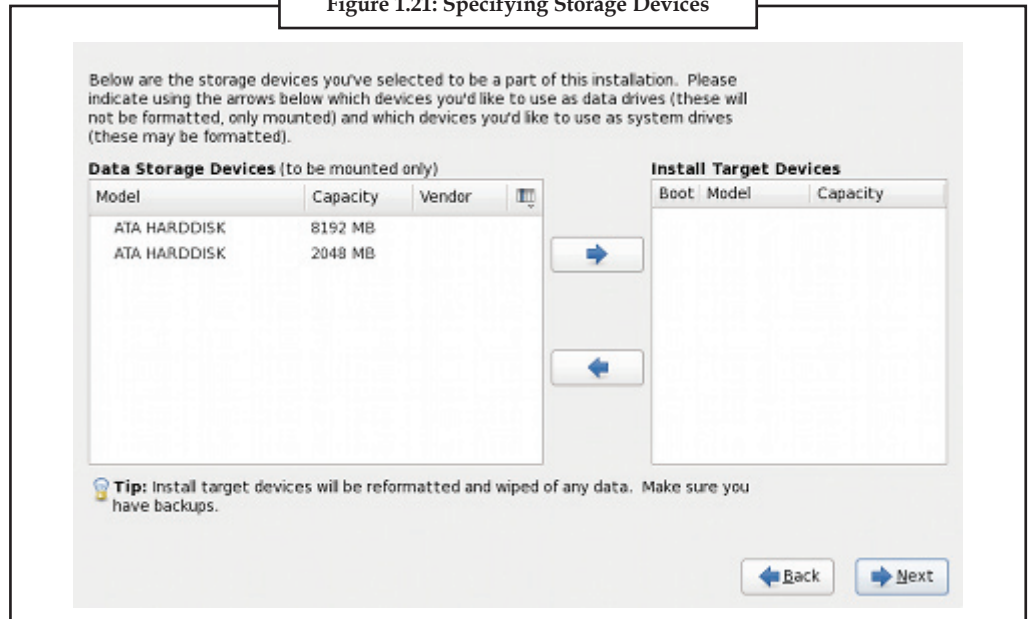
Figure 1.20: Partitioning Layout Choices



The Installer allows storage devices to be specified as either install target devices or data storage devices prior to installation.

Notes

Figure 1.21: Specifying Storage Devices

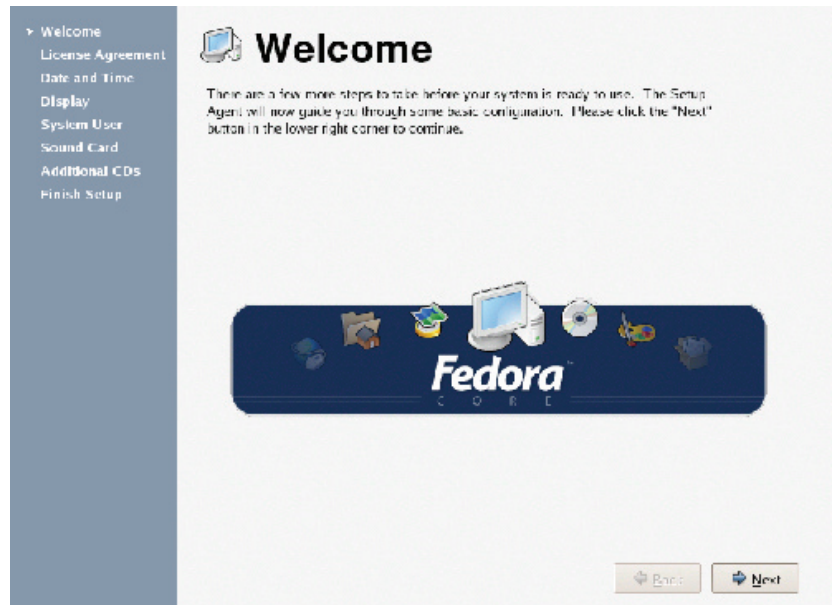


1.6 Creating User Accounts

With the main installation completed, a few housekeeping items are all that remain to be done. Your Fedora server will walk through the process of loading drivers, then present you with the Setup Agent: a set of tools for configuring your system once it has been installed. The use of such tools has become a common approach among Linux distributions, with SuSE providing the YaST2 tool, and Mandriva utilizing SystemDrak. You'll be presented with the Setup Agent's welcome screen, shown in figure, followed by the licence agreement. Once you've indicated that you agree to the license, you'll enter the configuration screens.

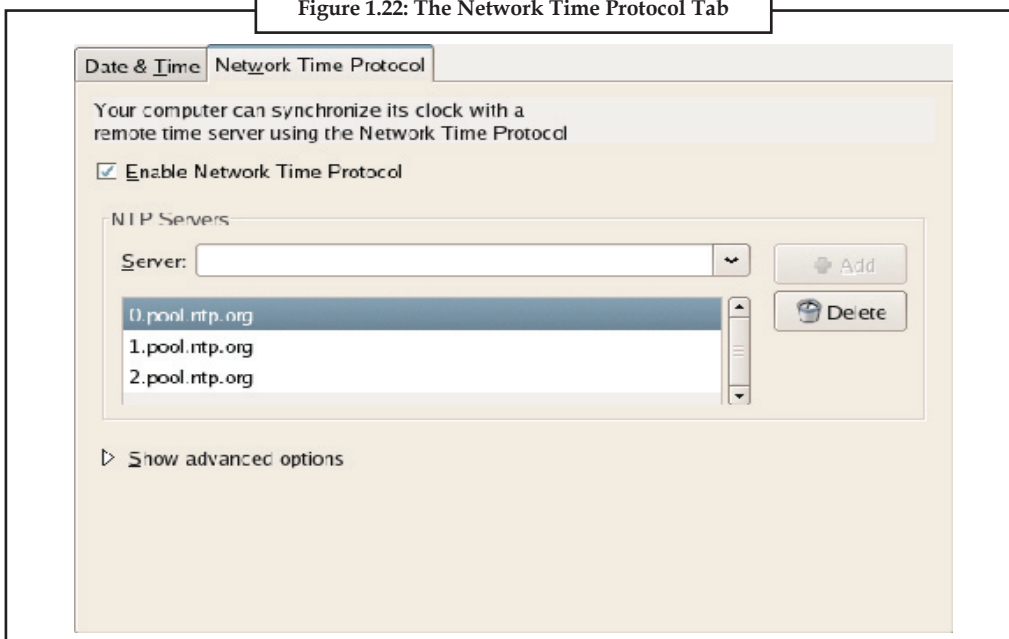


Example: Setup Agent's Welcome Screen



The Date and Time configuration screen provides two tabs: Date & Time and Network Time Protocol. The first tab allows you to confirm that the system clock is accurate. The second tab provides the ability to configure the Network Time Protocol (NTP) software, which can be used to synchronize your system's clock with an authoritative source. Selecting Enable Network Time Protocol in this screen, as illustrated in Figure 1.22, will enable the NTP daemon – a program that runs in the background, periodically checking your system time against the time returned by an NTP server. Several of these servers are listed in the Server drop-down (a good NTP server is pool.ntp.org. This is actually a name shared by many servers, ensuring that it's always available). If NTP is enabled and a server selected, the daemon will start, checking the selected server before moving on to the next Setup Agent screen.

Figure 1.22: The Network Time Protocol Tab

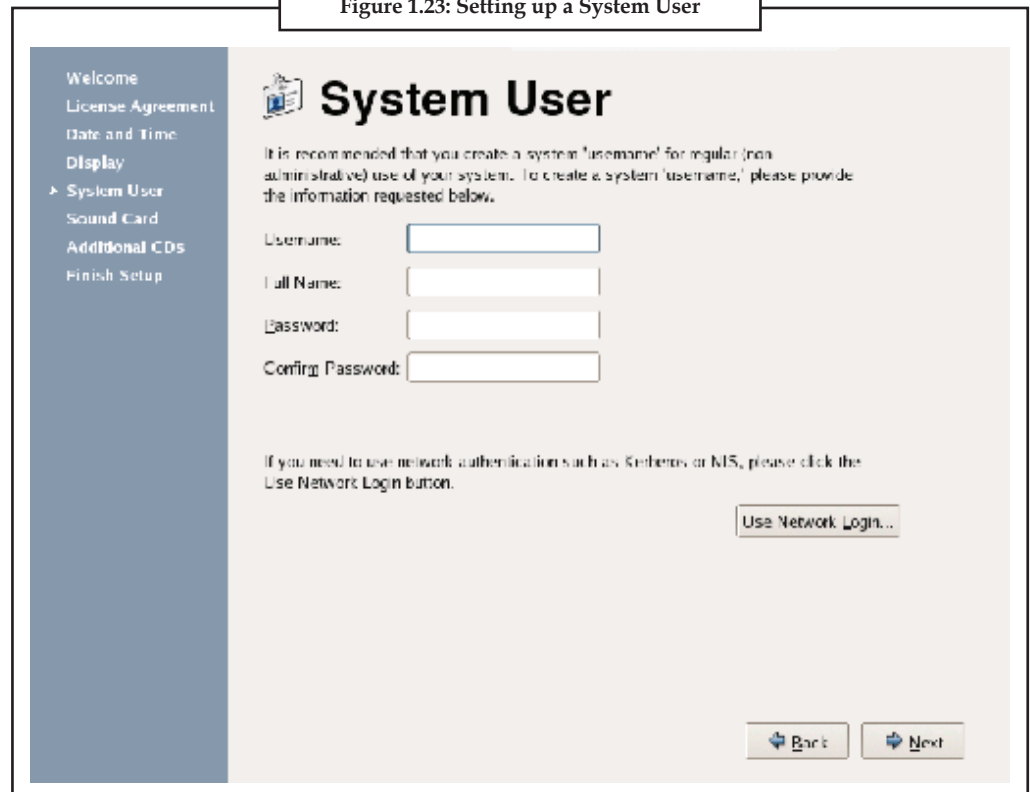


On the Display screen, you can select the type of monitor you're using, the resolution at which you'd like to work, and the color depth. If you can't find your monitor in the list, you can choose Generic CRT Display or Generic LCD Display.

The Setup Agent also provides a screen that allows us to configure an additional user. The user details include a Username, Full Name, and Password, as shown in Figure 1.23. If you decide to allow network logins, you can also select that option from this screen.

Notes

Figure 1.23: Setting up a System User



Create User Accounts

As with Windows, it's highly recommended that you create user accounts in addition to the main administration or root account. The root account is omnipotent; it has permissions to create, modify, and destroy any file on the system. Performing an action as root without careful forethought can have catastrophic consequences for your system. Nearly every Linux user can recount in detail the first (and likely only) time they rendered their system inoperable from the root account.

If the Fedora installer found a sound card on your system, you'll be asked to confirm its details. You'll also see a button with which to test it out, though, on a production Web server, this may not be necessary. There's also an Additional Software screen, which you can use to install any extra software you might need. You can just skip this screen for now.

Congratulations, you've now set up a Linux Web server! The graphical installation provides new Linux users with a manageable set of tools to get the system up and running. However, there are cases in which the text mode installation is a quicker and more efficient means to the same end.



Caselet

Nokia Planning Network Operating Centre: Maran

Our Bureau

Chennai, Aug. 1

NOKIA plans to set up a network operating centre in India, and that could possibly come to Chennai.

The centre would do remote monitoring of its network operating system in the Asian region. "We are expecting a statement from Nokia in this regard," Mr Dayanidhi Maran, Union Minister for Communications and Information Technology, said.

Asked on Intel's plans for manufacturing in India, Mr Maran said, "Hold on. They will have to come here."

It may be recalled that media reports have speculated on Intel's interest in a manufacturing plant in India and the sops it has been expecting from the government.

Source: <http://www.thehindubusinessline.in/2005/08/02/stories/2005080202870100.htm>

1.7 Summary

In order to be able to complete the installation procedure smoothly, you should collect certain information about your system before beginning the installation. First, determine what kind of hardware you have. Prepare a checklist to assist you. A server installation removes all existing partitions on all installed hard drives, so only choose server installation if you're sure you have nothing you want saved. If you are building your dual-boot server on a new computer, be sure to install and configure Windows first. By default, Windows doesn't recognize any of the native Linux filesystems. But, there are third-party utilities that allow Windows to read the drives of a Linux installation on the same machine. Different types of installation methods are available. You can select any one from them. All commercial Linux distributors provide some level of paid support, though the support period may vary widely from one distributor to another. Linux systems can be installed with a full complement of graphical tools, or as a minimal text-based system. The installers follow suit, providing options to complete an installation from a graphical environment, or from a purely text-based environment.

1.8 Keywords

Boot Disk: It is a removable digital data storage medium from which a computer can load and run (boot) an operating system or utility program.

Boot Loader: The program which makes multi booting possible is called a boot loader.

Disk Partitioning: It is the act or practice of dividing the storage space of a hard disk drive into separate data areas known as partitions.

Dual Booting: It is the act of installing multiple operating systems on a computer, and being able to choose which one to boot when switching on the computer power.

Installation: Installation (or setup) of a program (including drivers, plugins, etc.) is the act of putting the program onto a computer system so that it can be executed.

Notes

1.9 Self Assessment

Fill in the blanks:

1. The Fedora installer offers specialized installation types.
2. Automatic disk partitioning creates partitions
3. Fedora also offers Disk Druid which is a
4. Full form of RHEL is
5. SuSE offers two Hardware Compatibility Lists: and
6. In order to perform a server installation you will need at least of free hard-disk space.
7. Linux systems can be installed with a full complement of..... , or as a minimal text-based system.
8. Linux requires a minimal set of during the installation.
9. The now allows a user to choose basic storage devices or specialized storage devices.
10. Linux support can take many forms, the most popular being lists and forums.
11. The Date and Time configuration screen provides two tabs: and Network Time Protocol.
12. The Setup Agent also provides a screen that allows us to configure an
13. Fedora offers two options for setting the for your server.
14. If an capability is required, it's recommended that it be set up on a different server that's isolated as much as possible from the rest of the network.
15. A good rule of thumb to use when creating on your Linux machine is to create one and a half times the size of the machine's physical memory.

1.10 Review Questions

1. What do you think about the significance of RedHat Linux?
2. Make an analysis and write a note on installing the RedHat linux.
3. Examine the preparation for installing Linux in server configuration.
4. How to build dual-boot server on a new computer?
5. Create a Linux boot disk under DOS.
6. What are different types of Linux installation methods?
7. Describe the Disk Partitioning process during Linux installation.
8. How to set up the Root User?
9. Write short notes on the following topics:
 - (a) Disk Druid
 - (b) Swap Space

10. In your opinion is the Mandriva Linux Hardware is the Compatibility Database. Why or why not?

Notes

Answers: Self Assessment

- | | | |
|------------------------------------|-----------------------------|---------------|
| 1. three | 2. three | |
| 3. graphical portioning tool | 4. Red Hat Enterprise Linux | |
| 5. express search, extended search | 6. 1.8 GBytes | |
| 7. graphical tools | 8. hardware drivers | |
| 9. graphical installer | 10. web-based | |
| 11. Date & Time | 12. additional user | 13. time zone |
| 14. FTP | 15. swap space | |

1.11 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

<http://fedoraproject.org/wiki/FirstBoot>

www.google.co.in/search?hl=en&source=hp&q=case+study+on+redhat+linux&aq=f&aqi=&aql=&oq=

Unit 2: RedHat Linux Basics

CONTENTS

Objectives

Introduction

- 2.1 Working with Desktop
- 2.2 Starting X Windows and KDE
- 2.3 About GNOME
- 2.4 Using Terminal Emulator
 - 2.4.1 Don't try to use TERM Variable for Emulation
 - 2.4.2 Communication (Dialing) Programs
- 2.5 Testing Terminal Emulation
- 2.6 The Linux Console
- 2.7 Emulation Software
- 2.8 Summary
- 2.9 Keywords
- 2.10 Self Assessment
- 2.11 Review Questions
- 2.12 Further Readings

Objectives

After studying this unit, you will be able to:

- Know about working with desktop
- Learn using terminal emulator

Introduction

In graphical computing, a Desktop Environment (DE) commonly refers to a style of Graphical User Interface (GUI) that is based on the desktop metaphor which can be seen on most modern personal computers today. Almost universally adopted in modern computers, these graphical interfaces are designed to assist the user in easily accessing and configuring (or modifying) the most important (or frequently accessed) specific OS packed features, yet it is not meant to give access to the whole vast feature set found in an OS, reason for which the traditional, yet more complicated and less intuitive, Command-Line Interface (CLI) is still in use when full control over the OS is required.

A desktop environment typically consists of icons, windows, toolbars, folders, wallpapers, and desktop widgets.

Software which provides a desktop environment might also provide drag and drop functionality and other features which make the desktop metaphor more complete. On the whole, a desktop environment is to be an intuitive way for the user to interact with the computer using concepts which are similar to those used when interacting with the physical world, such as buttons and windows.

While the term desktop environment originally described a style of user interfaces following the desktop metaphor, it has also come to describe the programs that provide the metaphor itself. This usage has been popularized by the Gnome and the K Desktop Environment. Today, GNOME and KDE are the dominant solutions, and often installed by default on Linux systems.

2.1 Working with Desktop

In the past, huge mainframe computers with character-cell terminals used to be the only cost effective solution for institutions requiring operating systems that supported multiple users, number crunching power, and large databases. Personal computers and workstations were either too expensive or lacked the capabilities provided by mainframes. This resulted in the proliferation of mainframes in work settings that needed to share data and resources: colleges, universities, design engineering and research companies, hospitals, banking, and accounting firms.

Due to better technology and decreased costs, however, desktop workstations began to perform the same functions as a mainframe at a fraction of the cost. This provided an attractive alternative for institutions wishing to replace or upgrade their out-of-date mainframes, and created a large workstation market supplied by several companies that designed and developed their own workstation products.



Did u know? What is the role of vendors in supplying a graphical user interface?

Predecessors

Several bitmap display systems preceded X. From Xerox came the Alto (1973) and the Star (1981). From Apple came the Lisa (1983) and the Macintosh (1984). The Unix world had the Andrew Project (1982) and Rob Pike's Blit terminal (1982).

Carnegie-Mellon University produced a remote-access application called Alto Terminal, that displayed overlapping windows on the Xerox Alto, and made remote hosts (typically DEC VAX systems running Unix) responsible for handling window-exposure events and refreshing window contents as necessary.

X derives its name as a successor to a pre-1983 window system called W (the letter preceding X in the English alphabet). W Window System ran under the V operating system. W used a network protocol supporting terminal and graphics windows, the server maintaining display lists.

Origin and Early Development

The original idea of X emerged at MIT in 1984 as a collaboration between Jim Gettys (of Project Athena) and Bob Scheifler (of the MIT Laboratory for Computer Science). Scheifler needed a usable display environment for debugging the Argus system. Project Athena (a joint project between Digital Equipment Corporation (DEC), MIT and IBM to provide easy access to computing resources for all students) needed a platform-independent graphics system to link together its heterogeneous multiple-vendor systems; the window system then under development in Carnegie Mellon University's Andrew Project did not make licenses available, and no alternatives existed.

Notes

The project solved this by creating a protocol that could both run local applications and call on remote resources. In mid-1983 an initial port of W to Unix ran at one-fifth of its speed under V; in May 1984, Scheifler replaced the synchronous protocol of W with an asynchronous protocol and the display lists with immediate mode graphics to make X version 1. X became the first windowing system environment to offer true hardware-independence and vendor-independence.


Scheifler, Gettys and Ron Newman set to work and X progressed rapidly. They released Version 6 in January 1985. DEC, then preparing to release its first Ultrix workstation, judged X the only windowing system likely to become available in time. DEC engineers ported X6 to DEC's QVSS display on MicroVAX.

In the second quarter of 1985 X acquired color support to function in the DEC VAXstation-II/GPX, forming what became version 9.

A group at Brown University ported version 9 to the IBM RT/PC, but problems with reading unaligned data on the RT forced an incompatible protocol change, leading to version 10 in late 1985. By 1986, outside organizations had started asking for X. The release of X10R2 took place in January 1986; that of X10R3 in February 1986. Although MIT had licensed X6 to some outside groups for a fee, it decided at this time to license X10R3 and future versions under what became known as the MIT License, intending to popularize X further and in return, hoping that many more applications would become available. X10R3 became the first version to achieve wide deployment, with both DEC and Hewlett-Packard releasing products based on it. Other groups ported X10 to Apollo and to Sun workstations and even to the IBM PC/AT. Demonstrations of the first commercial application for X (a mechanical computer-aided engineering system from Cognition Inc. that ran on VAXes and displayed on PCs running an X server) took place at the Autofact trade show at that time. The last version of X10, X10R4, appeared in December 1986.

Attempts were made to enable X servers as real-time collaboration devices, much as Virtual Network Computing (VNC) would later allow a desktop to be shared. One such early effort was Philip J. Gust's SharedX tool.

Although X10 offered interesting and powerful functionality, it had become obvious that the X protocol could use a more hardware-neutral redesign before it became too widely deployed; but MIT alone would not have the resources available for such a complete redesign. As it happened, DEC's Western Software Laboratory found itself between projects with an experienced team. Smokey Wallace of DEC WSL and Jim Gettys proposed that DEC WSL build X11 and make it freely available under the same terms as X9 and X10. This process started in May 1986, with the protocol finalized in August. Alpha-testing of the software started in February 1987, beta-testing in May; the release of X11 finally occurred on September 15, 1987.



Note X11 protocol design, led by Scheifler, was extensively discussed on open mailing lists on the nascent Internet that were bridged to USENET newsgroups. Gettys moved to California to help lead the X11 development work at WSL from DEC's Systems Research Center, where Phil Karlton and Susan Angebrandt led the X11 sample server design and implementation. X therefore represents one of the first very large-scale distributed free software projects.

The MIT X Consortium and the X Consortium, Inc.

In 1987, with the success of X11 becoming apparent, MIT wished to relinquish the stewardship of X, but at a June 1987 meeting with nine vendors, the vendors told MIT that they believed in the need for a neutral party to keep X from fragmenting in the marketplace. In January 1988, the MIT X Consortium formed as a non-profit vendor group, with Scheifler as director, to direct the future development of X in a neutral atmosphere inclusive of commercial and educational interests. Jim

Fulton joined in January 1988 and Keith Packard in March 1988 as senior developers, with Jim focusing on Xlib, fonts, window managers, and utilities; and Keith re-implementing the server. Donna Converse, Chris D. Peterson, and Stephen Gildea joined later that year, focusing on toolkits and widget sets, working closely with Ralph Swick of MIT Project Athena. The MIT X Consortium produced several significant revisions to X11, the first (Release 2 - X11R2) in February 1988. Ralph Mor and Jay Hersh joined the staff later. In 1993, as the MIT X Consortium prepared to depart from MIT, the staff were joined by R. Gary Cutbill, Kaleb Keithley, and David Wiggins.

DECwindows CDE on OpenVMS 7.3-1In 1993, the X Consortium, Inc. (a non-profit corporation) formed as the successor to the MIT X Consortium. It released X11R6 on May 16, 1994. In 1995 it took on the development of the Motif toolkit and of the Common Desktop Environment for Unix systems. The X Consortium dissolved at the end of 1996, producing a final revision, X11R6.3, and a legacy of increasing commercial influence in the development.

Disability Action Committee for X (DACX)

In fact, the X Window System has become the de facto standard on computer workstations. Because of this, the X Window System provides an ideal place upon which to build accessibility solutions: any solutions for X.

Windows will apply across many makers and models of workstations. Since the early 1980s, the Trace Center has worked closely with companies that develop personal computers and their operating systems to develop strategies which make them more accessible to individuals with disabilities.

For people with mobility impairments, strategies have been developed to provide basic access in a number of operating systems. Among these strategies are AccessDOS for IBM computers running DOS, Easy Access for the Apple Macintosh, Access Pack for Microsoft Windows, and those being built into IBM OS/2. Likewise, the Trace Center has worked closely with companies such as Apple and Berkeley Systems Inc. to develop strategies to assist in their development of products for people with visual impairments.

Building upon and expanding the model used to provide access in the personal computer market, the Trace Center brought together a group of researchers, consumers and companies interested in developing access solutions for workstations. This group, known as the Disability Action Committee for X (DACX), had an initial meeting in conjunction with the Closing-The-Gap Conference in Minneapolis, in October 1992. Discussion at the kick-off meeting centered around what should be done to provide basic disability access to computer workstations.

To best utilize the limited resources within DACX, the group formed subcommittees to concentrate initial efforts in three specific areas:

1. The definition of hooks and library calls needed to provide screen reader access to applications (called "X clients") running on a workstation;
2. Implementation of screen magnification in X;
3. Implementation of "built-in" features to provide basic access for people who have mobility impairments that prevent them from effectively using the keyboard and mouse.

At the present time, DACX includes members from organizations like Bell Atlantic, Berkeley Systems Inc., Digital Equipment Corp., Georgia Tech, IBM, MIT, SUN Microsystems, and the Trace Center.

Notes



Note Trace Center is acting as the coordinator and secretariat for DACX, as well as assisting in implementation of access strategies in software development and testing. The Trace Center maintains an electronic mail account at the University to allow DACX members to communicate issues quickly.

The Downside

The downside of x-windows is that it requires a lot of bandwidth to operate. You can get it to work over a 14.4 baud modem, but it is slow. Even over Ethernet graphic intensive clients can be sluggish. Also, more than one flavor of x-windows emerged so that the main benefit is somewhat negated. Motif is one of the main flavors of x-windows, but others exist.

Enter KDE and GNOME

X has become the de facto window system in free software.

KDE was founded in 1996 by Matthias Ettrich. At the time, he was troubled by the inconsistencies in UNIX applications. He proposed a new desktop environment. He also wanted to make this desktop easy to use. His initial Usenet post spurred a lot of interest.

Matthias chose to use the Qt toolkit for the KDE project. At the time, Qt did not use a free software licence. Members of the GNU project became concerned with the use of such a toolkit for building a free software desktop environment. In August 1997, two projects were started in response to KDE: the Harmony toolkit (a free replacement for the Qt libraries) and GNOME (a different desktop without Qt and built entirely on top of free software). GTK+ was chosen as the base of GNOME in place of the Qt toolkit.

In November 1998, the Qt toolkit was licensed under the free/open source Q Public License (QPL). But debate continued about compatibility with the GNU General Public License (GPL). In September 2000, Trolltech made the UNIX version of the Qt libraries available under the GPL, in addition to the QPL, which has eliminated the concerns of the Free Software Foundation.

Both KDE and GNOME now participate in freedesktop.org, an effort to standardize UNIX desktop interoperability, although there is still some competition between them.

KDE and Gnome are complete desktop environments that consist of a large number of tightly integrated yet still separate pieces of software. Gnome uses a window manager called metacity, KDE uses kwin. Both these desktops can be used with any other window manager if you do not like the default choice.

Linux is like Lego. You can build your own desktop environment. Both KDE and Gnome are just big packages with software aimed to look and feel the same way, work well together and generally give you a nice experience. If you dislike a component, then replace it with something else. It's that simple.

Application that are "made for gnome" or "made for kde" can be used with any desktop. This only means that that the program use a set of library functions found in their underlying gnomelibs or kdelibs. You do not need to use the actual desktops to use the applications, software made for KDE and Gnome can be used with any window manager / desktop as long as you got the proper libraries installed. There is no reason to use only applications made for the desktop you prefer, the "best software" for one task is made for KDE, the best for another task is made for Gnome. Use the best from both worlds.

Both KDE and Gnome can be customized to behave exactly the way you want. What desktop you prefer is your own choice and preference. When in doubt, try to learn both. Or experiment with other desktops. Remember, *nix applications are not locked to the desktop they are made for, Gnome applications can be used in KDE and vice versa.

There is no “best desktop”, but there is a desktop that’s best for you. It’s a matter of preference, and hardware.



Task

“KDE and Gnome are complete desktop environments.” Explain.

About KDE

KDE or the K Desktop Environment, is a network transparent contemporary desktop environment for UNIX workstations. KDE seeks to fulfill the need for an easy to use desktop for UNIX workstations, similar to desktop environments found on Macintosh and Microsoft Windows operating systems.

The UNIX operating system is according to us the best available today. When it comes to stability, scalability and openness UNIX has no competition. In fact UNIX has been the undisputed choice of information technology professionals for many years. The lack of an easy to use contemporary desktop environment, however, has prevented UNIX from finding its way onto desktops of typical computer users in offices and homes. UNIX today dominates the server market and is the preferred computing platform for computing professionals and scientists alike.

The Internet, a household name traces its heritage to UNIX. In spite of such ubiquitous creations from the UNIX community, average computer users still expect it to be difficult to use and often stay away. This fact is particularly unfortunate since a number of implementations of UNIX, all of which are of exceptional quality and stability (Debian GNU/Linux, FreeBSD, NetBSD etc.) are freely available off the Internet.



Caution Many KDE applications have a K in the name, mostly as an initial letter and capitalized.

However, there are notable exceptions like ksynaptic, whose K is not capitalized, and Amarok (formerly named amaroK). The K in many KDE applications is obtained by spelling a word which originally begins with C or Q differently, for example Konsole and Kuickshow. Also, some just prefix a commonly used word with a K, for instance KOffice. Among KDE 4 applications and technologies, however, the trend is not to have a K in the name at all, such as Plasma, Phonon and Dolphin.

The KDE project’s mascot is a green dragon named Konqi. Gandalf the wizard was the former mascot for the KDE project during its 1.x and 2.x versions, but he was dropped due to copyright issues (his resemblance to Gandalf).

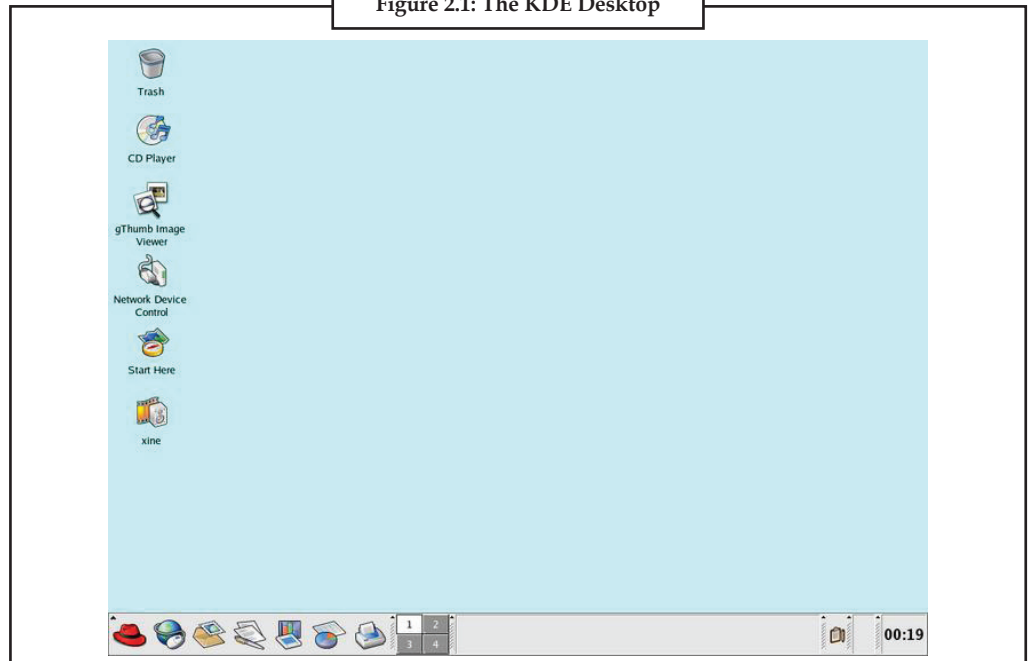
Major applications for KDE include:

1. Amarok - Audio player
2. Dolphin - File manager
3. K3b - Optical disc authoring software
4. Kate - Text editor
5. Kdenlive - Non-linear video editing
6. KDevelop - Integrated development environment
7. Konsole - Terminal emulator

Notes

8. Kontact - Personal information manager featuring an e-mail client, a news client, a feed aggregator, To-do lists and more.
9. Kopete - Instant messaging client
10. Konqueror - Web browser
11. KOffice - Office suite
12. KTorrent - BitTorrent client

Figure 2.1: The KDE Desktop



Licensing Issues

All new source code and related data files in KDE SVN must meet the following requirements:

1. All files must be in their preferred modifiable form, or alongside the file which is the preferred modifiable form. In the 'alongside' case, there must be clear instructions for how to build using the preferred modifiable form.
2. All source files must contain a copyright header which identifies the copyright holder(s) together with an e-mail address that can be used to reach the copyright holder.
3. Each source file must contain one of the licence headers listed below to state under which terms the software may be used, modified and redistributed.
4. Source files that are part of a library with a public API which is part of the KDE Platform (kdelibs, kdepimlibs and kdbase-runtime) must be licensed under one of the following terms:
 - (a) LGPL version 2.1 as listed in kdelibs/COPYING.LIB or later
 - (b) LGPL version 2.1 or version 3 or later versions approved by the membership of KDE e.V.
 - (c) BSD license as listed below.
 - (d) Ensure that the BSD license does not contain the so called 'advertisement clause'.

- (e) MIT license as listed below.
 - (f) X11 license as listed below.
5. Any other source files must be licensed under one of the terms listed under (4) or one of the following terms:
 - (a) GPL version 2 as listed in kdelibs/COPYING or later
 - (b) GPL version 2 or version 3 or later versions approved by the membership of KDE e.V.
 - (c) Code copied from Qt may be licenced under GPL version 2 or version 3 or later versions approved by Trolltech ASA and the KDE Free Qt Foundation.
 6. Translations of text from the files described in section 4 must be licenced under one of the terms in sections 4. Translations of text from other files must be licenced under one of the terms in sections 4 or 5.
 7. Icons in kdelibs, kdeplibs or kdebase runtime must be licenced under the LGPL 3 or later versions approved by KDE e.V. Icons elsewhere can be licenced under any of the terms in section 4 or 5.
 8. Documentation must be licensed under one of the following terms:
 - (a) FDL versions 1.2 as listed in kdelibs/COPYING.DOC or later versions with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
 - (b) FDL versions 1.2 as listed in kdelibs/COPYING.DOC or later versions approved by KDE e.V. with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Documentation may additionally be licensed under any of the terms in section 4 or 5.

Other data included in KDE's codebase must be licenced under one of the terms in sections 4 or 5.
 10. Contributors who have signed the Fiduciary License Agreement with KDE e.V. may licence files described in section 3 under LGPL 2.1 & 3 only and may licence other source files under GPL 2 & 3 only.
 11. As new versions of GNU licenses are published they will be discussed by the membership of KDE e.V. and accepted via an announcement on <http://ev.kde.org> if: not adopting the licence would harm the future distribution of KDE, the licence preserves freedom for our developers & users and a vote by the membership agrees to the acceptance of the new licence version.

2.2 Starting X Windows and KDE

The X Window System provides a way to execute programs (including a graphical environment such as KDE) in a general way when `startx` is executed. This is in the form of two files (`.xserverrc` and `.xinitrc`). Normally these reside in your home directory and are used for starting server and client related applications respectively. The `startkde` script in the `kde/bin` directory executes all the programs required by KDE in one handy script. You can therefore start KDE along with your X server by adding a line to execute `startkde` to the `.xinitrc` file in your home directory.

Example below shows samples for the relevant files.

```
$ cat ~/.xserverrc
exec X -screen 0 1024x768x16 -engine 4 -ac -nowinkill -noreset -emulate3buttons
100
```

Notes

```
$ cat ~/.xinitrc
/usr/local/kde1/bin/startkde
```

There are two ways to install these files:

1. Copy the above samples to the related lines manually, for example
\$ echo /usr/local/kde1/bin/startkde > ~/.xinitrc
2. Install the kdemisc-1.1.2 package, which will install these files in /home/administrator by default. After that you can move/copy the files to your home directory. Before installing this package make sure that you backup your previous files.

Once you have created the two files you can start KDE on the base prompt with: startx &.

KDE Basics

KDE can be viewed either as a user desktop or as a development platform for applications. Here we'll quickly describe the major features of both.

The K Desktop Environment as User Desktop

Konqueror Browser

Konqueror is KDE's next-generation web browser, file manager and document viewer. Widely heralded as a technological break-through for the GNU/Linux desktop, the standards-compliant Konqueror has a component-based architecture which combines the features and functionality of Internet Explorer/Netscape Communicator and Windows Explorer. Konqueror supports the full gamut of current Internet technologies, including JavaScript, Java, HTML 4.0, CSS-1 and -2 (Cascading Style Sheets), SSL (Secure Socket Layer for secure communications) and Netscape Communicator plug-ins (for playing Flash, RealAudio, RealVideo and similar technologies).

KOffice Office Suite

KDE also ships with the highly anticipated release of the KOffice suite. The integrated suite consists of a spreadsheet application (KSpread), a vector drawing application (Karbon), a frame-based word-processing application (KWord), a presentation program (KPresenter), and a chart and diagram application (KChart). Native file formats are XML-based, and work on filters for proprietary binary file formats is progressing. Combined with a powerful scripting language and the ability to embed individual components within each other using KDE's component technology (KParts), the free KOffice suite provides all the necessary functionality to all but the most demanding power users.

KIO Network Transparency

In addition, KIO's network transparency offers seamless support for accessing or browsing files on GNU/Linux, NFS shares, MS Windows SMB shares, HTTP pages, FTP directories and LDAP directories. The modular, plug-in nature of KDE's file architecture makes it simple to add additional protocols (such as IPX or WebDAV) to KDE, which would then automatically be available to all KDE applications.

aRts Multimedia Architecture

Notes

KDE 2.0 introduced a new multimedia architecture based on aRts, the Analog Realtime Synthesizer. aRts enables playing multiple audio or video streams concurrently, whether on the desktop or over a network. ARts is a full-featured sound system, and includes filters, a modular analog synthesizer and a mixer. Its architecture allows developers to create additional filter plugins and users to apply sequences of filters using a graphical drag-n-drop approach. Video support is available for MPEG versions 1, 2 and 4 (experimental), as well as the AVI and DivX formats.

Customizability

KDE's customizability touches every aspect of this next-generation desktop. KDE's sophisticated theme support starts with Qt's style engine, which permits developers and artists to create their own widget designs. KDE ships with over 14 of these styles, some of which emulate the look of various operating systems. Other configuration options permit users to: choose among icon themes and system sounds (using a simple drop-and-replace approach); configure key bindings; select from over 50 languages; customize toolbar layouts and entries and menu composition; employ single-click or double-click to activate desktop items; navigate the desktop using a keyboard instead of a mouse; and much, much more. Moreover, KDE fully supports Unicode and KHTML is the only free HTML rendering engine on GNU/Linux/X11 that features nascent support for BiDi scripts such as Arabic and Hebrew.

Standards Compliance

Besides the exceptional compliance with Internet and file-sharing standards mentioned above, KDE achieves exceptional compliance with the available GNU/Linux desktop standards. KWin, KDE's new re-engineered window manager, complies to the new Window Manager Specification. Konqueror and KDE comply to the Desktop Entry Standard. KDE generally complies with the X Drag-and-Drop (XDND) protocol as well as with the X11R6 session management protocol (XSMP).

The K Development Environment as Development Platform

KDE offers developers a rich set of major technologies. Chief among these are the Desktop COmmunication Protocol (DCOP), the I/O libraries (KIO), the component object model (KParts), an XML-based GUI class, and a standards-compliant HTML rendering engine (KHTML).

DCOP Messaging

DCOP is a client-to-client communications protocol intermediated by a server over the standard X11 ICE library. The protocol supports both message passing and remote procedure calls using an XML-RPC to DCOP "gateway". Bindings for C, C++ and Python, as well as experimental Java bindings, are available.

KIO Network Technology

KIO implements application I/O in a separate process to enable a non-blocking GUI without the use of threads. The class is network transparent and hence can be used seamlessly to access HTTP, FTP, POP, IMAP, NFS, SMB, LDAP and local files. Moreover, its modular and extensible design permits developers to "drop in" additional protocols, such as WebDAV, which will then automatically be available to all KDE applications. KIO also implements a trader which can locate handlers for specified mimetypes; these handlers can then be embedded within the requesting application using the KParts technology.

Notes

KParts Components

KParts, KDE's component object model, allows an application to embed another within itself. The technology handles all aspects of the embedding, such as positioning toolbars and inserting the proper menus when the embedded component is activated or deactivated. KParts can also interface with the KIO trader to locate available handlers for specific mimetypes or services/protocols. This technology is used extensively by the KOffice suite and Konqueror.

XML GUI Builder

The XML GUI employs XML to create and position menus, toolbars and possibly other aspects of the GUI. This technology offers developers and users the advantage of simplified configurability of these user interface elements across applications and automatic compliance with the KDE Standards and Style Guide irrespective of modifications to the standards.

KHTML Rendering Engine


KHTML is an HTML 4.0 compliant rendering and drawing engine. The class supports the full gamut of current Internet technologies, including JavaScript, Java, HTML 4.0, CSS-2 (Cascading Style Sheets), SSL (Secure Socket Layer for secure communications) and Netscape Communicator plugins (for viewing Flash, RealAudio, RealVideo and similar technologies). The KHTML class can easily be used by an application as either a widget (using normal X Window parenting) or as a component (using the KParts technology). KHTML, in turn, has the capacity to embed components within itself using the KParts technology.

The KDE Control Center

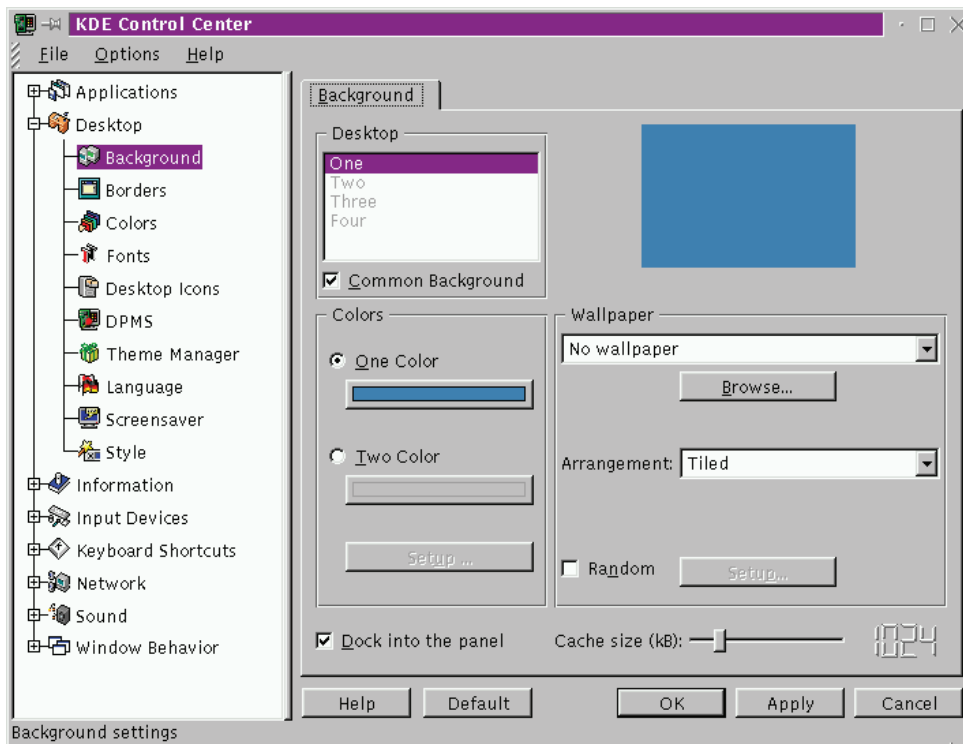
The KDE Control Center also known as KControl, is the centralized settings manager for the KDE desktop environment. It is part of the kadmin package. It can be considered the KDE counterpart of the Windows Control Panel.

KControl has a modular architecture. The window is split into two parts: the left side displays the list of available modules (also known as applets), and the right side displays the current module. Applications can install their own KControl modules (an example being Konqueror), although most applications prefer to use their own, independent settings managers.

Although KControl is installed by default in Kubuntu, that distribution uses a non-standard version of it by default, redesigned to resemble the Mac OS X system settings manager.



Notes "classic" KControl is also available, however, a ported version of the Kubuntu systemsettings application has also replaced KControl in KDE 4. Systemsettings has been accepted as "an improved user interface for configuring the desktop and other aspects of the system".



The KDE Control Center

Here you will find settings that change the way your KDE desktop and applications look.

Background

This section controls the color or image that is set as your desktop background. These settings can be applied to all virtual workspaces, or to only a specific one. There are a range of background wallpapers that come with KDE or you can supply your own.

Colors

This is where you can modify the colors for your KDE applications. There are a variety of color schemes installed with KDE by default, and you can find others at KDE-look.org. You can also create your own. Here you can also modify the contrast and choose whether you want your KDE colors to be applied to non-KDE applications, for a more consistent overall appearance.

Fonts

Here you can control the various font settings for KDE applications. You can also modify here anti-aliasing settings, including what range of fonts to exclude from anti-aliasing settings.

Desktop Icons

This section is where you can manage your icon themes and other settings related to icons. New icon themes can be downloaded from KDE-look.org, and installed here. Conversely, you can remove icon themes by highlighting them in the list and clicking remove. You can also set icon sizes for various uses in KDE and effects to apply to icons.

Notes

Screen Saver

Here you can configure options about your screensaver. You can configure the timeout before it starts, and whether it requires a password to unlock the screen.

Style

This section allows you to modify your widget style. A variety of styles come with KDE. This is also where you would enable or disable interface options such as transparent menus, showing icons on buttons and tooltips. Some styles have more configuration options than others.

Theme Manager

This is where you can create and manage themes that are made up of personalized settings. They are a combination of desktop background, colors, KDE widget styles, icons, fonts and what Screensaver you'd like to display. This allows you to save your favorite "looks" and apply them with the click of a mouse button.

2.3 About GNOME

GNOME is a desktop environment, providing an alternative to the KDE interface, for Linux and other Unix environments that grew out of the GNU Project. It is the graphical user interface which runs on top of a computer operating system—composed entirely of free software. GNU Network Object Modeling Environment is an international project which includes creating software development frameworks, selecting application software for the desktop, and working on the programs which manage application launching, files handling, and window and task management.

GNOME is part of the GNU Project and can be used with various Unix-like operating systems, most notably those built on top of the Linux kernel and the GNU system, and as part of Java Desktop System in Solaris. Companies such as Red Hat, Inc. and Ximian, formerly Helix Code, Inc. and acquired by Novell, support the GNOME environment.

Starting X Windows and GNOME

GDM is the GNOME Display Manager, a graphical login program. The easiest way to start GNOME is with GDM, the GNOME Display Manager. GDM, which is installed as a part of the GNOME desktop (but is disabled by default), can be enabled by adding `gdm_enable="YES"` to `/etc/rc.conf`. Once you have rebooted, GNOME will start automatically once you log in - no further configuration is necessary.

GNOME may also be started from the command-line by properly configuring a file named `.xinitrc`. If a custom `.xinitrc` is already in place, simply replace the line that starts the current window manager with one that starts `/usr/local/bin/gnome-session` instead. If nothing special has been done to the configuration file, then it is enough simply to type:

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

Next, type `startx`, and the GNOME desktop environment will be started.

Figure 2.2: The GNOME Desktop



Task

"GDM is the GNOME Display Manager, a graphical login program." Comment.

GNOME Basics

GNOME is a powerful but simple desktop environment with a strong focus on usability, accessibility, and internationalization. GNOME is designed to be usable by everybody, regardless of technical expertise, disabilities, or native language. GNOME makes it easy for people to use their computers.

GNOME provides a comprehensive developer platform that allows developers to create professional software that is easy to use and aesthetically pleasing. This document provides a high-level overview of the GNOME platform along with links to detailed documentation on each part of the platform.

The GNOME Configuration Tool

GConf is the system for storing and retrieving configuration settings in GNOME. GConf consists of two parts: a client library for accessing settings, and a session daemon which is responsible for the details of storing and retrieving those settings. Using a daemon allows GConf to use different storage backends, validate input, and provide simultaneous access to different applications.

Notes

Settings stored in GConf are stored and retrieved using a unique key, or identifier string. Keys use a simple hierarchical namespace to avoid collision among settings for applications and the desktop. You can provide a schema file to detail your configuration keys. This allows GConf to validate the type of the input, and to show localized documentation about the key. This helps systems administrators, who can set multiple settings at once without having to navigate preference dialogs.

GConf can look up settings from different settings at once, typically from different locations on the file system. By having appropriate system sources configured, GConf enables systems administrators to provide both default and mandatory settings for all users. Tools such as GNOME's Configuration Editor and Sabayon make it easy to deploy fully configured systems using GConf.

The GConf client library provides notifications of changes to settings, making it easy to provide instant-apply settings in your application, regardless if settings are changed from within your application or using another tool. Setting the value of a key will notify all interested applications, allowing desktop-wide and other cross-application settings to work instantly and effortlessly.

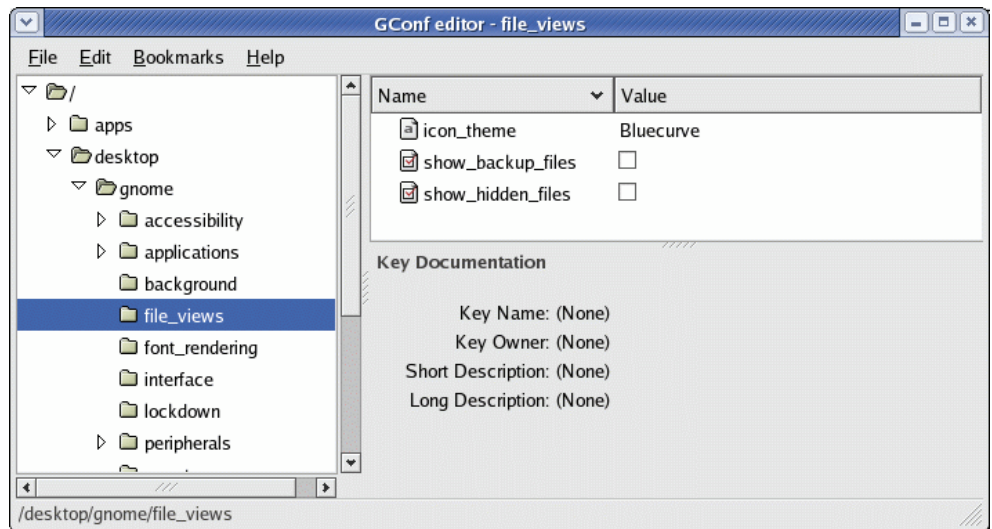
GConf makes it easy to lock down systems by setting particular keys read-only, preventing users from changing their values. In addition, GNOME provides a number of high-level keys that can be used to disable actions such as saving to disk and changing the panel layout. Tools such as Pessulus make it easy for administrators to find and lock down important keys.

You should use GConf to store all user preferences in your application. Using GConf will make it easy to provide instant-apply preferences, and it will make your settings accessible to systems administrators and configuration and backup tools.

GConf is used by GNOME to expose settings to multiple applications. GConf is the GNOME way of persisting application settings and should be used by GNOME application programmers. GConf includes notification service alerts to applications to changes in configuration data, and is used by GNOME itself.



Did u know? A command line tool (`gconftool`, FC3 `/usr/bin/gconftool-2`) and a simple GUI application (`gconf-editor`) are supplied to facilitate administration.



GConf-editor

The GNOME Control Center

Notes

The GNOME Control Center allows you to configure various parts of your GNOME system with different tools called “capplets”. The capplets are associated with the core set of GNOME applications and helps you to manage your system.

You can configure the appearance and operation of GNOME and GNOME-compliant applications by using the GNOME control center, shown in figure. The function of Control Center resembles that of the Windows 9x control panel, though it looks different and works somewhat differently. To launch the control center, select Settings GNOME Control Center from the GNOME main menu.

Like the Windows control panel, which uses small programs called applets to perform its functions, the GNOME control center uses small programs called capplets. However, the control center’s user interface hides this detail from you, so you needn’t normally be aware of what’s happening behind the scenes. The control center user interface resembles that of file manager and menu editor: The left pane of the control center window presents a hierarchically structured set of configuration categories and the right pane displays information pertaining to the current choice.

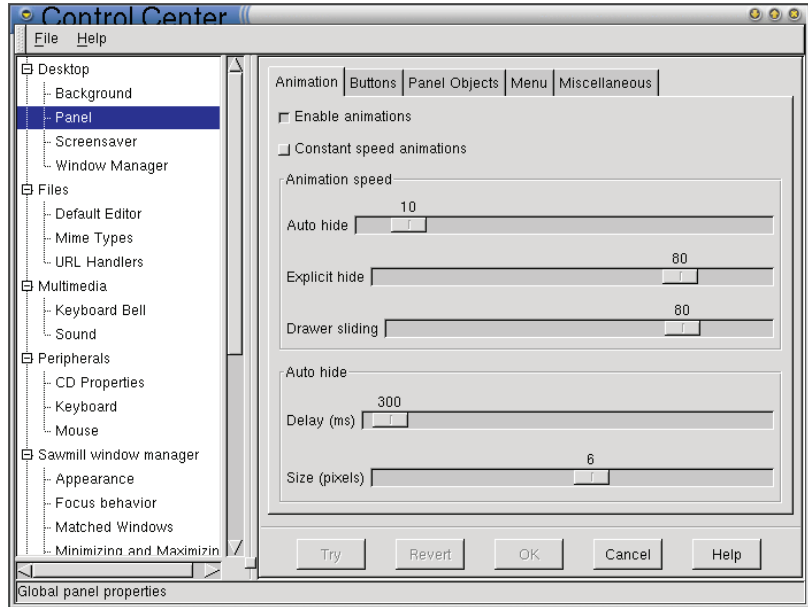
1. Using control center, you can:
2. Select background properties
3. Configure a screen saver
4. Select a desktop theme
5. Select a window manager
6. Configure the default text editor
7. Specify MIME types that control the handling of multimedia files
8. Configure the keyboard bell and sounds
9. Configure keyboard and mouse properties
10. Specify applications that GNOME automatically launches when it starts
11. Specify a variety of options governing the appearance of GNOME-compliant applications

Simply select the configuration category by clicking in the left pane. You can then revise the configuration parameters by specifying the desired values in the right pane. The buttons that appear in the right pane vary from capplet to capplet.

Notes



Notes Try button lets you experiment without permanently altering the GNOME configuration. The OK button permanently updates the GNOME configuration whereas the Cancel button discards your changes.



The GNOME Control Center

Desktop

The Desktop section controls the most visible global settings for your Gnome environment.

1. **The Background Properties Capplet:** The background image can be set here by either selecting a color or an image. If you would like to set the background by any other means you may disable this capplet by unselecting Use Gnome to set background checkbox.
2. **Global Panel Preferences:** This capplet is provided by the Gnome Panel and is documented in the Panel Manual.
3. **The Screensaver Capplet:** In this capplet you can change your screensaver properties. This capplet contains a list of available screen savers you may choose and a demo screen.

In Global Screen Saver settings you can change the time, password, and power management properties. You can decide how long you would like the screen saver to wait before starting by typing the number of minutes in the Start After text box. If you would like a password to return to your desktop click the Require Password button. Your account login password is the password set for the screen saver.

4. **Theme Selector:** The Theme Selector capplet allows you to select which GTK theme you would like to run. GTK themes are coordinated settings that define the look and feel of such elements of graphical user interface as buttons, menus, scrollbars etc. of all Gnome applications (more generally, of all applications using the GIMP Toolkit (GTK), hence the name).

5. **Window Manager Capplet:** Gnome is not dependent on any one window manager; therefore the Window Manager capplet allows you to select which window manager you wish to use. Your current window manager will be labeled Current. It only shows Gnome-compliant window managers; if you want to use other window managers, you have to tell Window Manager capplet about them. If you wish to add a new window manager to the main list you may press the Add button. This will launch the Add New Window Manager dialog.



Did u know? The Document Handlers section of the Gnome Control Center allows you to change the way certain file types and functions are viewed, edited or manipulated.

2.4 Using Terminal Emulator

Since a PC has a screen and keyboard (as does a terminal) but also has much more computing power, it's easy to use some of this computing power to make the PC computer behave like a text terminal. This is one type of terminal emulation. Another type of terminal emulation is where you set up a real terminal to emulate another brand/model of terminal. To do this you select the emulation you want (called "personality" in Wyse jargon) from the terminal's set-up menu. This section is about the first type of emulation: emulating a terminal on a PC.

In emulation, one of the serial ports of the computer will be used to connect the emulated terminal to another computer, either with a direct cable connection from serial port to serial port, or via a modem. Emulation provides more than just a terminal since the PC doing the emulation can also do other tasks at the same time it's emulating a terminal. For example, kermit or zmodem may be run on the PC to enable transfer of files over the serial line (and possibly over the phone line via a modem) to the other computer that you are connected to. The emulation needs only to be run on one of the virtual consoles of the PC, leaving the other virtual consoles available for using the PC in command-line-interface.

Much emulation software is available for use under the MS Windows OS. See Make a non-Linux PC a terminal This can be used to connect a Windows PC to a Linux PC (as a Text-Terminal). Most Linux free software can only emulate a VT100, VT102, or VT100/ANSI. If you find out about any others, let me know. Since most PC's have color monitors while VT100 and VT102 were designed for a monochrome monitor, the emulation usually adds color capabilities (including a choice of colors). Sometimes the emulation is not 100% perfect but this usually causes few problems. For using a Mac computer to emulate a terminal see the mini-howto: Mac-Terminal.

2.4.1 Don't try to use TERM Variable for Emulation

Some have erroneously thought that they could create an emulator at a Linux console (monitor) by setting the environment variable TERM to the type of terminal they would like to emulate. This does not work. The value of TERM only tells an application program what terminal you are using. This way it doesn't need to interactively ask you this question. If you're at a Linux PC monitor (command line interface) it's a terminal of type "Linux" and you can't change this. So you must set TERM to "Linux".

If you set it to something else you are fibbing to application programs. As a result they will incorrectly interpret certain escape sequences from the console resulting in a corrupted interface. Since the Linux console behaves almost like a vt100 terminal, it could still work almost OK if you falsely claimed it was a vt100 (or some other terminal which is something like a vt100). It may seem to work OK most of the time but once in a while will make a mistake when editing or the like.

Notes



Task

Analyse the uses of TERM variable.

2.4.2 Communication (Dialing) Programs

Dialing programs for making a PPP connection to the Internet don't normally include any terminal emulation. But some other modem dialing programs (such as minicom or seyon) do. Using them, one may (for example) dial up some public libraries to use their catalogs and indexes, (or even read magazine articles). They are also useful for testing modems. Seyon is only for use with X Window and can emulate Tektronix 4014 terminals.

The communication program Kermit doesn't do terminal emulation as it is merely a semi-transparent pipe between whatever terminal you are on and the remote site you are connected to. Thus if you use kermit on a Linux PC the terminal type will be "Linux". If you have a Wyse60 connected to your PC and run kermit on that, you will appear as a Wyse60 to the remote computer (which may not be able to handle Wyse60 terminals). Minicom emulates a VT102 and if you use it on Wyse60 terminal VT102 escape sequences coming into your computer's serial port from a remote computer will get translated to the Wyse escape sequences before going out another serial port to your Wyse60 terminal. Kermit can't do this sort of thing.

Emulators exist under DOS such as telix and procomm work just as well. The terminal emulated is often the old VT100, VT102, or ANSI (like VT100).

Emulation under X Window

Xterm (or uxterm which is like xterm except it supports unicode) may be run under X Window. They can emulate a VT102, VT220, or Tektronix 4014. There are also various xterm emulations (although there is no physical terminal named "xterm"). If you want pixmaps but don't need the Tektronix 4014 emulation (a vector graphics terminal; see Graphics Terminals) you may use eterm. Predecessors to eterm are rxvt and xvt. One way to change font size in xterm is to right click the mouse while holding down the Ctrl key.

For non-Latin alphabets, kterm is for Kanji terminal emulation (or for other non-Latin alphabets) while xcin is for Chinese. There is also 9term emulation. This seems to be more than just an emulator as it has a built-in editor and scroll-bars. It was designed for Plan 9, a Unix-like operating system from AT&T.

Real Terminals Better

Unless you are using X Window with a large display, a real terminal is often nicer to use than emulating one. It usually has better resolution for text, and has no disk drives to make annoying noises.

2.5 Testing Terminal Emulation

For the VT series terminals there is a test program: `vttest` to help determine if a terminal behaves correctly like a vt53, vt100, vt102, vt220, vt320, vt420 etc. There is no documentation but it has menus and is easy to use. To compile it run the configure script and then type "make". It may be downloaded from: <http://ibiblio.unc.edu/pub/Linux/utills/console/>

2.6 The Linux Console

The console for a PC Linux system is normally the computer monitor in text mode. It emulates a terminal of type "Linux" and the escape sequences it uses are in the man page: `console_codes`. There is no way (unless you want to spend weeks rewriting the kernel code) to get it to emulate anything else. Setting the TERM environment variable to any type of terminal other than "Linux" will not result in emulating that other terminal. It will only result in a corrupted interface since you have falsely declared (via the TERM variable) that your "terminal" is of a type different from what it actually is. See Don't Use TERM For Emulation.



Note

In some cases, the console for a Linux PC is a text-terminal. One may recompile Linux to make a terminal receive most of the messages which normally go to the console. See `Make a Serial Terminal the Console`.

The "Linux" emulation of the monitor is flexible and has features which go well beyond those of the vt102 terminal which it was intended to emulate. These include the ability to use custom fonts and easily re-map the keyboard. These extra features reside in the console driver software (including the keyboard driver). The console driver only works for the monitor and will not work for a real terminal even if it's being used for the console. Thus the "console driver" is really a "monitor driver". In the early days of Linux one couldn't use a real terminal as the console so "monitor" and "console" were once always the same thing.

The stty commands work for the monitor-console just like it was a real terminal. They are handled by the same terminal driver that is used for real terminals. Bytes headed for the screen first go thru the terminal (tty) driver and then thru the console driver. For the monitor some of the stty commands don't do anything (such as setting the baud rate). You may set the monitor baud rate to any allowed value (such as a slow 300 speed) but the actual speed of putting text on the monitor screen will not actually change. The file `/etc/ioctl.save` stores stty settings for use only when the console is in single user mode (but you are normally in multiuser-user mode). This is explained (a little) in the `init` man page.

Many commands exist to utilize the added features provided by the console-monitor driver. Real terminals, which use neither scan codes nor VGA cards, unfortunately can't use these features. To find out more about the console see the `Keyboard-and-Console-HOWTO`. Also see the various man pages about the console (type "man -k console"). Unfortunately, much of this documentation is outdated.

2.7 Emulation Software

Emulators often don't work quite right so before purchasing software you should try to thoroughly check out what you will get.

Make a Linux PC a Terminal

Unless you want to emulate the standard vt100 (or close to it) or a Wyse 60, there doesn't seem to be much free terminal emulation software available for Linux. The free programs `minicom` and `seyon` (only for X Window) can emulate a vt100 (or close to it). `Seyon` can also emulate a Tektronix 4014 terminal.

`Minicom` may be used to emulate a directly connected terminal by simply starting `minicom` (after configuring it for the serial port used). Of course, you don't dial out and when you want to quit (after you logout from the other PC) you use `minicom's` `q` command to quit without reset since there is no modem to reset. When `minicom` starts, it automatically sends out a modem init string

Notes

to the serial port. But since there's no modem there, the string gets put after the "login:" prompt. If this string is mostly capital letters, the getty program (which runs login) at the other PC may think that your terminal has only capital letters and try to use only capital letters. To avoid this, configure the modem init strings sent by minicom to null (erase the init strings).

The terminal emulator "Procomm" (which is from Dos), can be used on a Linux PC if you run dosemu to emulate Dos. For details see: <http://solarflow.dyndns.org/pcplus/>.

There's a specialized Linux distribution: Serial Terminal Linux. It will turn a PC to into a minicom-like terminal. It's small (fits on a floppy) and will not let you use the PC for any other purpose (when it's running). It will let you have more than one session running (similar to virtual terminals), one for each serial port you have.

TERM (non-free commercial software from Century Software) Terminal Emulator can emulate Wyse60, 50; VT 220, 102, 100, 52; TV950, 925, 912; PCTERM; ANSI; IBM3101; ADM-11; WANG 2110. Block mode is available for IBM and Wyse. It runs on a Linux PC.

Make a non-Linux PC a Terminal

Emulators exist which run on non-Linux PCs. They permit you to use a non-Linux-PC as a terminal connected to a Linux-PC. Under DOS there is telix and procomm. Windows comes with "HyperTerminal" (formerly simply called "Terminal" in Windows 3.x and DOS). Competing with this is "HyperTerminal Private Edition" <http://www.hilgraeve.com/hpte/index.html> which is non-free to business. It can emulate vt-220. The Windows "terminals" are intended for calling out with a modem but they should also work as directly connected terminals?? Turbosoft's TtWin can emulate over 80 different terminals under Windows. See <http://www.ttwin.com/> or <http://www.turbosoft.com.au/> (Australia). See also Reflection

For the Mac Computer there is emulation by Carnation Software <http://www.carnationsoftware.com/carnation/HT.Carn.Home.html>

One place to check terminal emulation products is Shuford's site, but it seems to lists old products (which may still work OK). The fact that most only run under DOS (and not Windows) indicates that this info is dated. See http://www.cs.utk.edu/~shuford/terminal/term_emulator_products.txt.



Caselet

Red Hat on a roll with new Linux Offerings'

V. Rishi Kumar

HYDERABAD, Sept.1

WITH the Central and State Governments stepping up efforts to reach out e-governance applications for the common man to bridge the digital divide, various Linux-based applications are gaining momentum and Red Hat is on the roll, according to Mr Sachin Dabir, Head-Enterprise Sales, Red Hat India.

Mr Dabir told Business Line that the Linux operating system and Advanced Server were being supported by some of the world's top enterprises, software and hardware vendors, offering a range of infrastructure solutions at attractive price points.

With cost of ownership becoming a critical issue, many organisations and corporate users are migrating to these solutions from Red Hat Linux away from expensive operating systems.

Contd...

The launch of Advanced Server had received industry-wide international support from some of the global leading players including Oracle and Veritas Software along with some hardware server platform providers, Mr Dabir said and added that this had begun to make impact in India within several industry segments and the Governments adopting for various applications.

With a growing number of alliance partners and ISVs (independent software vendors) supporting the Red Hat drive, these offerings had become easier to adopt as they came along with integrated services including centralised maintenance through single intuitive Web interface, thereby providing an ideal platform, Mr Dabir said.

Red Hat India, a joint venture between Red Hat Inc, (a premium open source and Linux provider) and Clover Technologies, markets and offers support and services for Red Hat Linux in the sub-continent. It was in the process of building channel partners and was teaming up with OEMs with its products, Mr Dabir said.

“Take the case of educational institutions or for that matter, universities, the best way to go forward is to adopt open source-based products that help them save on investments. Especially in multi-user environments, the cost of ownership, which is a decisive factor in expanding operations, comes to the fore,” he said.

Source: <http://www.thehindubusinessline.in/2002/09/02/stories/2002090201050200.htm>

2.8 Summary

In graphical computing, a Desktop Environment (DE) commonly refers to a style of Graphical User Interface (GUI) that is based on the desktop metaphor which can be seen on most modern personal computers today. A desktop environment typically consists of icons, windows, toolbars, folders, wallpapers, and desktop widgets. KDE or the K Desktop Environment, is a network transparent contemporary desktop environment for UNIX workstations. GNOME is a desktop environment, providing an alternative to the KDE interface, for Linux and other Unix environments that grew out of the GNU Project. While the term desktop environment originally described a style of user interfaces following the desktop metaphor, it has also come to describe the programs that provide the metaphor itself. This usage has been popularized by the Gnome and the K Desktop Environment. Today, GNOME and KDE are the dominant solutions, and often installed by default on Linux systems.

2.9 Keywords

Desktop Environment (DE): In graphical computing, a desktop environment (DE) commonly refers to a style of Graphical User Interface (GUI) that is based on the desktop metaphor which can be seen on most modern personal computers today.

GNOME: It is a desktop environment, providing an alternative to the KDE interface, for Linux and other Unix environments that grew out of the GNU Project.

KDE or the K Desktop Environment: It is a network transparent contemporary desktop environment for UNIX workstations.

X Window System: It is a Graphical User Interface (GUI) of Linux operating system.

Notes

2.10 Self Assessment

Fill in the blanks:

1. Full form of DACX is
2. Full form of CLI is
3. Full form of GUI is
4. KDE was founded in 1996 by
5. is KDE's next-generation web browser, file manager and document viewer.
6. Full form of DCOP is
7. is the system for storing and retrieving configuration settings in GNOME.
8. The GNOME Control Center allows you to configure various parts of your GNOME system with different tools called
9. GDM is the
10. preceded X. From Xerox came the Alto (1973) and the Star (1981).
11. Windows will apply across many makers and models of
12. KIO implements application I/O in a separate process to enable a non-blocking without the use of threads.
13. The X Window System provides a way to execute programs (including a graphical environment such as KDE) in a general way when is executed.
14. GConf makes it easy to lock down systems by setting particular keys , preventing users from changing their values.
15. may be run under X Window.

2.11 Review Questions

1. While impressive theoretically, due to better technology and decreased costs, however, desktop workstations began to perform the same functions as a mainframe at a fraction of the cost. What do you think as the reason behind it?
2. What are the components of desktop environment?
3. What is DACX? What are the best ways to best utilize the limited resources within DACX?
4. What is KDE? KDE can be viewed either as a user desktop or as a development platform for applications. Comment
5. Do you think GNOME is a powerful but simple desktop environment with a strong focus on usability, accessibility, and internationalization?
6. What are the major applications for KDE?
7. Write short notes on the following topics:
 - (a) Konqueror
 - (b) KOffice suite
 - (c) DCOP Messaging

- (d) KControl
- (e) GConf
8. "KDE and Gnome are complete desktop environments that consist of a large number of tightly integrated yet still separate pieces of software." Explain
9. Why Global Screen Saver settings can change the time, password, and power management properties?
10. "The console for a PC Linux system is normally the computer monitor in text mode." Comment

Answers: Self Assessment

- | | |
|--------------------------------------|------------------------------------|
| 1. Disability Action Committee for X | 2. Command-Line Interface |
| 3. Graphical User Interface | 4. Matthias Ettrich |
| 5. Konqueror | 6. Desktop Communication Protocol |
| 7. GConf | 8. Caplets |
| 9. GNOME Display Manager | 10. Several bitmap display systems |
| 11. Workstations | 12. GUI |
| 13. Startx | 14. read-only |
| | 15. Xterm |

2.12 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online link

<http://www.eskimo.com/~johnnyb/computers/stl/>.

Unit 3: File System Hierarchy

CONTENTS

Objectives

Introduction

3.1 File Systems

3.1.1 Types of File Systems

3.1.2 File Systems and Operating Systems

3.2 File Concept

3.3 Access Methods

3.3.1 Sequential Access

3.3.2 Direct Access

3.3.3 Other Access Methods

3.4 Directory Structure

3.5 File System Structure

3.6 Summary

3.7 Keywords

3.8 Self Assessment

3.9 Review Questions

3.10 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss File Systems
- Understand File Concept
- Discuss Access Methods
- Understand Directory Structure
- Understand File System Structure

Introduction

Another part of the operating system is the file manager. While the memory manager is responsible for the maintenance of primary memory, the file manager is responsible for the maintenance of secondary storage (e.g., hard disks).

Each file is a named collection of data stored in a device. The file manager implements this abstraction and provides directories for organizing files. It also provides a spectrum of commands to read and write the contents of a file, to set the file read/write position, to set and use the protection mechanism, to change the ownership, to list files in a directory, and to remove a file. The file manager provides a protection mechanism to allow machine users to administer how

processes executing on behalf of different users can access the information in files. File protection is a fundamental property of files because it allows different people to store their information on a shared computer, with the confidence that the information can be kept confidential.

In addition to these functions, the file manager also provides a logical way for users to organize files in secondary storage.

In this unit we take a brief look at how the operating system deals with files system. We will discuss how an operating system keeps track of files. The hard disk is comprised of a large number of sequentially numbered sectors. As files are created, free sectors are allocated to hold the file contents and marked as allocated. To keep track of the sectors and whether they are allocated or free, and to which file they belong, the operating system maintains a number of tables. In this unit we will learn the how the disk space is allocated and how the efficiency and performance of file system can be increased. Hard drives are not perfect: they develop defects due to magnetic dropout and imperfect manufacturing. On more primitive disks, this is checked when the disk is formatted and these damaged sectors are avoided. If sector becomes damaged under operation, the structure of the disk must be patched up by some repair program. Usually the data are lost. In this unit data recovery is also discussed.

3.1 File Systems

A file system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them. File systems may use a data storage device such as a hard disk or CD-ROM and involve maintaining the physical location of the files, they might provide access to data on a file server by acting as clients for a network protocol (e.g., NFS, SMB, or 9P clients), or they may be virtual and exist only as an access method for virtual data.

More formally, a file system is a set of abstract data types that are implemented for the storage, hierarchical organization, manipulation, navigation, access, and retrieval of data. File systems share much in common with database technology, but it is debatable whether a file system can be classified as a special-purpose database (DBMS).

3.1.1 Types of File Systems

File system types can be classified into disk file systems, network file systems and special purpose file systems.

1. **Disk file systems:** A disk file system is a file system designed for the storage of files on a data storage device, most commonly a disk drive, which might be directly or indirectly connected to the computer.



Example: Disk file systems include FAT, FAT32, NTFS, HFS and HFS+, ext2, ext3, ISO 9660, ODS-5, and UDF. Some disk file systems are journaling file systems or versioning file systems.

2. **Flash file systems:** A flash file system is a file system designed for storing files on flash memory devices. These are becoming more prevalent as the number of mobile devices is increasing, and the capacity of flash memories catches up with hard drives.

While a block device layer can emulate a disk drive so that a disk file system can be used on a flash device, this is suboptimal for several reasons:

Erasing blocks: Flash memory blocks have to be explicitly erased before they can be written to.

Random access: Disk file systems are optimized to avoid disk seeks whenever possible, due to the high cost of seeking. Flash memory devices impose no seek latency.

Notes

Wear leveling: Flash memory devices tend to wear out when a single block is repeatedly overwritten; flash file systems are designed to spread out writes evenly.

Log-structured file systems have all the desirable properties for a flash file system. Such file systems include JFFS2 and YAFFS.

3. **Database file systems:** A new concept for file management is the concept of a database-based file system. Instead of, or in addition to, hierarchical structured management, files are identified by their characteristics, like type of file, topic, author, or similar metadata. Example: dbfs.

4. **Transactional file systems:** Each disk operation may involve changes to a number of different files and disk structures. In many cases, these changes are related, meaning that it is important that they all be executed at the same time. Take for example a bank sending another bank some money electronically. The bank's computer will "send" the transfer instruction to the other bank and also update its own records to indicate the transfer has occurred. If for some reason the computer crashes before it has had a chance to update its own records, then on reset, there will be no record of the transfer but the bank will be missing some money.

Transaction processing introduces the guarantee that at any point while it is running, a transaction can either be finished completely or reverted completely (though not necessarily both at any given point). This means that if there is a crash or power failure, after recovery, the stored state will be consistent. (Either the money will be transferred or it will not be transferred, but it won't ever go missing "in transit".)

This type of file system is designed to be fault tolerant, but may incur additional overhead to do so.

Journaling file systems are one technique used to introduce transaction-level consistency to file system structures.

5. **Network file systems:** A network file system is a file system that acts as a client for a remote file access protocol, providing access to files on a server.



Example: Network file systems include clients for the NFS, SMB protocols, and file-system-like clients for FTP and WebDAV.

6. **Special purpose file systems:** A special purpose file system is basically any file system that is not a disk file system or network file system. This includes systems where the files are arranged dynamically by software, intended for such purposes as communication between computer processes or temporary file space.

Special purpose file systems are most commonly used by file-centric operating systems such as Unix.



Example: The procfs (/proc) file system used by some Unix variants, which grants access to information about processes and other operating system features.

Deep space science exploration craft, like Voyager I & II used digital tape based special file systems. Most modern space exploration craft like Cassini-Huygens used Real-time operating system file systems or RTOS influenced file systems. The Mars Rovers are one such example of an RTOS file system, important in this case because they are implemented in flash memory.

The time taken to erase blocks can be significant, thus it is beneficial to erase unused blocks while the device is idle.



Did u know? What is database file system?

3.1.2 File Systems and Operating Systems

Most operating systems provide a file system, as a file system is an integral part of any modern operating system. Early microcomputer operating systems' only real task was file management - a fact reflected in their names. Some early operating systems had a separate component for handling file systems which was called a disk operating system. On some microcomputers, the disk operating system was loaded separately from the rest of the operating system. On early operating systems, there was usually support for only one, native, unnamed file system.



Example: CP/M supports only its own file system, which might be called "CP/M file system" if needed, but which didn't bear any official name at all.

Because of this, there needs to be an interface provided by the operating system software between the user and the file system. This interface can be textual (such as provided by a command line interface, such as the Unix shell, or OpenVMS DCL) or graphical (such as provided by a graphical user interface, such as file browsers). If graphical, the metaphor of the folder, containing documents, other files, and nested folders is often used.

Flat file systems: In a flat file system, there are no subdirectories-everything is stored at the same (root) level on the media, be it a hard disk, floppy disk, etc. While simple, this system rapidly becomes inefficient as the number of files grows, and makes it difficult for users to organise data into related groups.

Like many small systems before it, the original Apple Macintosh featured a flat file system, called Macintosh File System. Its version of Mac OS was unusual in that the file management software (Macintosh Finder) created the illusion of a partially hierarchical filing system on top of MFS. This structure meant that every file on a disk had to have a unique name, even if it appeared to be in a separate folder. MFS was quickly replaced with Hierarchical File System, which supported real directories.

3.2 File Concept

A file is a collection of letters, numbers and special characters: it may be a program, a database, a dissertation, a reading list, a simple letter etc. Sometimes you may import a file from elsewhere, for example from another computer. If you want to enter your own text or data, you will start by creating a file. Whether you copied a file from elsewhere or created your own, you will need to return to it later in order to edit its contents.

The most familiar file systems make use of an underlying data storage device that offers access to an array of fixed-size blocks, sometimes called sector, generally 512 bytes each. The file system software is responsible for organizing these sectors into files and directories, and keeping track of which sectors belong to which file and which are not being used. Most file systems address data in fixed-sized units called "clusters" or "blocks" which contain a certain number of disk sectors (usually 1-64). This is the smallest logical amount of disk space that can be allocated to hold a file.

However, file systems need not make use of a storage device at all. A file system can be used to organize and represent access to any data, whether it be stored or dynamically generated (e.g, from a network connection).

Notes

Whether the file system has an underlying storage device or not, file systems typically have directories which associate file names with files, usually by connecting the file name to an index into a file allocation table of some sort, such as the FAT in an MS-DOS file system, or an inode in a Unix-like file system. Directory structures may be flat, or allow hierarchies where directories may contain subdirectories. In some file systems, file names are structured, with special syntax for filename extensions and version numbers. In others, file names are simple strings, and per-file metadata is stored elsewhere.

Other bookkeeping information is typically associated with each file within a file system. The length of the data contained in a file may be stored as the number of blocks allocated for the file or as an exact byte count. The time that the file was last modified may be stored as the file's timestamp. Some file systems also store the file creation time, the time it was last accessed, and the time that the file's meta-data was changed. (Note that many early PC operating systems did not keep track of file times.) Other information can include the file's device type (e.g., block, character, socket, subdirectory, etc.), its owner user-ID and group-ID, and its access permission settings (e.g., whether the file is read-only, executable, etc.).

The hierarchical file system was an early research interest of Dennis Ritchie of Unix fame; previous implementations were restricted to only a few levels, notably the IBM implementations, even of their early databases like IMS. After the success of Unix, Ritchie extended the file system concept to every object in his later operating system developments, such as Plan 9 and Inferno.


Traditional file systems offer facilities to create, move and delete both files and directories. They lack facilities to create additional links to a directory (hard links in Unix), rename parent links (".." in Unix-like OS), and create bidirectional links to files.

Traditional file systems also offer facilities to truncate, append to, create, move, delete and in-place modify files. They do not offer facilities to prepend to or truncate from the beginning of a file, let alone arbitrary insertion into or deletion from a file. The operations provided are highly asymmetric and lack the generality to be useful in unexpected contexts.



Example: Interprocess pipes in Unix have to be implemented outside of the file system because the pipes concept does not offer truncation from the beginning of files.

Secure access to basic file system operations can be based on a scheme of access control lists or capabilities. Research has shown access control lists to be difficult to secure properly, which is why research operating systems tend to use capabilities. Commercial file systems still use access control lists.



Task "A file system can be used to organize and represent access to any data, whether it be stored or dynamically generated." Describe

3.3 Access Methods

There are several ways that the information in the file can be accessed. Some systems provide only one access method for files. On other systems, many different access methods are supported.

3.3.1 Sequential Access

Information in the file is processed in order, one record after the other. This is by far the most common mode of access of files. For example, computer editors usually access files in this fashion. A read operation reads the next portion of the file and automatically advances the file pointer. Similarly, a write appends to the end of the file and the file pointer. Similarly, a write appends to

the end of the file and the file pointer. Similarly, a write appends to the end of the end of the file and advances to the end of the newly written material (the new end of file). Such a file can be reset to the beginning, and, on some systems, a program may be able to skip forward or backward n records, for some integer n . This scheme is known as sequential access to a file. Sequential access is based on a tape model of a file.

A sequential file may consist of either formatted or unformatted records. If the records are formatted, you can use formatted I/O statements to operate on them. If the records are unformatted, you must use unformatted I/O statements only. The last record of a sequential file is the end-of-file record.

3.3.2 Direct Access

Direct access is based on a disk model of a file. For direct access, the file is viewed as a numbered sequence of block or records. A direct-access file allows arbitrary blocks to be read or written. Thus, after block 18 has been read, block 57 could be next, and then block 3. There are no restrictions on the order of reading and writing for a direct access file. Direct access files are of great use for intermediate access to large amounts of information.

The file operations must be modified to include the block number as a parameter. Thus, we have "read n ", where n is the block number, rather than "read next", and "write n ", rather than "write next". An alternative approach is to retain "read next" and "write next" and to add an operation; "position file to n " where n is the block number. Then, to effect a "read n ", we would issue the commands "position to n " and then "read next".

Not all OS support both sequential and direct access for files. Some systems allow only sequential file access; others allow only direct access. Some systems require that a file be defined as sequential or direct when it is created; such a file can be accessed only in a manner consistent with its declaration.



Note

Direct-access files support both formatted and unformatted record types. Both formatted and unformatted I/O work exactly as they do for sequential files.

3.3.3 Other Access Methods

Other access methods can be built on top of a direct-access method. These additional methods generally involve the construction of an index for a file. The index contains pointers to the various blocks. To find an entry in the file, the index is searched first and the pointer is then used to access the file directly to find the desired entry. With a large file, the index itself may become too large to be kept in memory. One solution is to create an index for the index file. The primary index file would contain pointers to secondary index files, which would point to the actual data items.



Example: IBM's indexed sequential access method (ISAM) uses a small master index that points to disk blocks of a secondary index. The secondary index blocks point to the actual file blocks. The file is kept sorted on a defined key. To find a particular item, we first make a binary search of the master index, which provides the block number of the secondary index. This block is read in, and again a binary search is used to find the block containing the desired record. Finally, this block is searched sequentially. In this way, any record can be located from its key by at most direct access reads.

Notes

3.4 Directory Structure

The directories themselves are simply files indexing other files, which may in turn be directories if a hierarchical indexing scheme is used. In order to protect the integrity of the file system in spite of user or program error, all modifications to these particular directory files are commonly restricted to the file management system. The typical contents of a directory are:

1. file name (string uniquely identifying the file), type (e.g. text, binary data, executable, library), organization (for systems that support different organizations).
2. device (where the file is physically stored), size (in blocks), starting address on device (to be used by the device I/O subsystem to physically locate the file);
3. creator, owner, access information (who is allowed to access the file, and what they may do with it);
4. date of creation/of last modification;
5. locking information (for the system that provide file/record locking).

As far as organization, by far the most common scheme is the hierarchical one: a multi-level indexing scheme is used, in which a top-level directory indexes both files and other directories, which in turn index files and directories, and so on. Usually this scheme is represented in the form of a tree.

The hierarchical architecture has distinct advantages over a simple, one-level indexing one: the tree structure can be effectively used to reflect a logical organization of the data stored in the files; names can be reused (they must uniquely identify files within each directory, not across the whole file system); in a multi-user system, name conflicts between files owned by different users can be solved by assigning to each user a directory for her own files and sub-directories, the so called user's "home" directory.

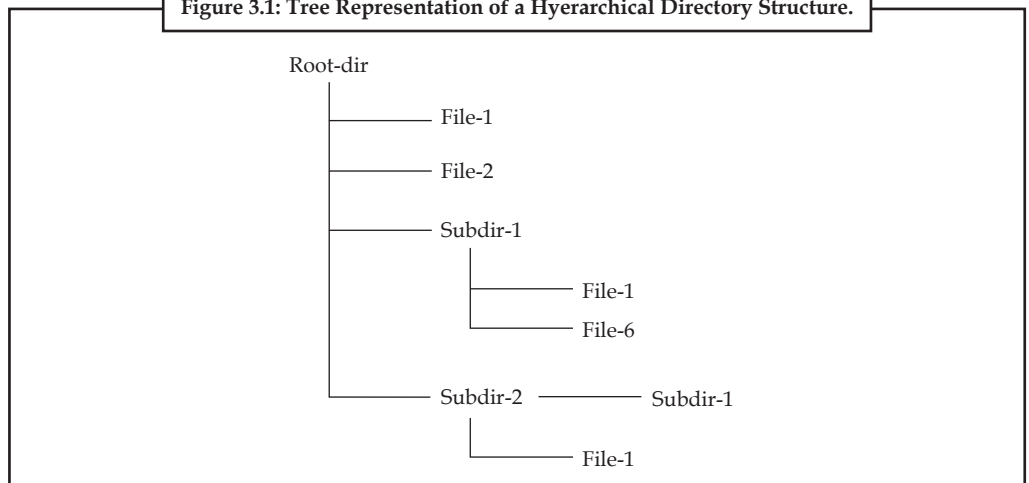
A complete indexing of a file is obtained by navigating the tree starting from the top-level, "root", directory, and walking along a path to the tree leaf corresponding to the file.

A "pathname" is thus obtained, which uniquely identifies the file within the whole file system.



Example: The pathname for file "File-6" in Figure 3.1 is ":Root-dir:Subdir-1:File-6", where a colon is used to separate tree nodes.

Figure 3.1: Tree Representation of a Hierarchical Directory Structure.



A complete pathname is not the only way to identify a file in the directory tree structure: a “relative” pathname, starting from a parent directory is suited just as well, provided that the FMS already knows about that directory. This addressing methods can be usefully exploited by making the FMS assign to all processes a “current working directory” (CWD) attribute, i.e. the complete patname of a directory of interest, and defining a way for the process to identify files by just specifying a “relative” pathname starting from that directory. In the same example, if “:Root-dir:Subdir-1” is the CWD of a process, the above file might be identified simply as “File-6”, using the convention that patnames not starting with a color are relative to the CWD. The advantage is twofold: the entire file system structure up to the CWD need not be known by a program (hence its data can be safely moved in other directories without having to rewrite the program), and file access time is decreased.



Caution Keep in mind it is no longer necessary to navigate the whole tree in order to find the address of a file.

3.5 File System Structure

The file system structure is the most basic level of organization in an operating system. Almost all of the ways an operating system interacts with its users, applications, and security model are dependent upon the way it organizes files on storage devices. Providing a common file system structure ensures users and programs are able to access and write files.

File systems break files down into two logical categories:

1. Shareable vs. unsharable files
2. Variable vs. static files

Shareable files are those that can be accessed locally and by remote hosts; unsharable files are only available locally. Variable files, such as documents, can be changed at any time; static files, such as binaries, do not change without an action from the system administrator.

The reason for looking at files in this manner is to help correlate the function of the file with the permissions assigned to the directories which hold them. The way in which the operating system and its users interact with a given file determines the directory in which it is placed, whether that directory is mounted with read-only or read/write permissions, and the level of access each user has to that file. The top level of this organization is crucial. Access to the underlying directories can be restricted or security problems could manifest themselves if, from the top level down, it does not adhere to a rigid structure.

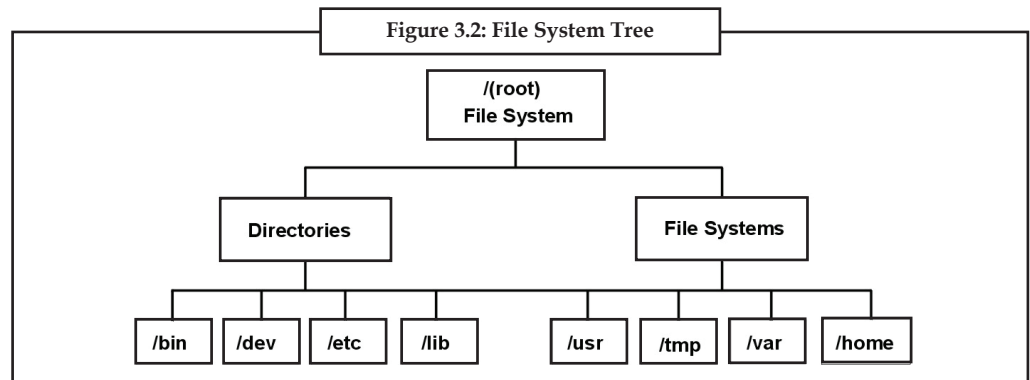
It is important to understand the difference between a file system and a directory. A file system is a section of hard disk that has been allocated to contain files. This section of hard disk is accessed by mounting the file system over a directory. After the file system is mounted, it looks just like any other directory to the end user.

However, because of the structural differences between the file systems and directories, the data within these entities can be managed separately.

When the operating system is installed for the first time, it is loaded into a directory structure, as shown in the following illustration.

Figure 3.2 File System Tree. This tree chart shows a directory structure with the / (root) file system at the top, branching downward to directories and file systems. Directories branch to /bin, /dev, /etc, and /lib. File systems branch to /usr, /tmp, /var, and /home.

Notes



The directories on the right (/usr, /tmp, /var, and /home) are all file systems so they have separate sections of the hard disk allocated for their use. These file systems are mounted automatically when the system is started, so the end user does not see the difference between these file systems and the directories listed on the left (/bin, /dev, /etc, and /lib).

On standalone machines, the following file systems reside on the associated devices by default:

/File System	/Device
/dev/hd1	/home
/dev/hd2	/usr
/dev/hd3	/tmp
/dev/hd4	/(root)
/dev/hd9var	/var
/proc	/proc
/dev/hd10opt	/opt

The file tree has the following characteristics:

1. Files that can be shared by machines of the same hardware architecture are located in the /usr file system.
2. Variable per-client files, for example, spool and mail files, are located in the /var file system.
3. The /(root) file system contains files and directories critical for system operation. For example, it contains
 - (a) A device directory (/dev)
 - (b) Mount points where file systems can be mounted onto the root file system, for example, /mnt
4. The /home file system is the mount point for users' home directories.
5. For servers, the /export directory contains paging-space files, per-client (unshared) root file systems, dump, home, and /usr/share directories for diskless clients, as well as exported /usr directories.
6. The /proc file system contains information about the state of processes and threads in the system.
7. The /opt file system contains optional software, such as applications.



Did u know? What are the characteristics of file?

/bin	Symbolic link to the /usr/bin directory.
/dev	Contains device nodes for special files for local devices. The /dev directory contains special files for tape drives, printers, disk partitions, and terminals.
/etc	Contains configuration files that vary for each machine. Examples include: <ul style="list-style-type: none"> • /etc/hosts • /etc/passwd
/export	Contains the directories and files on a server that are for remote clients.
/home	Serves as a mount point for a file system containing user home directories. The /home file system contains per-user files and directories. In a standalone machine, a separate local file system is mounted over the /home directory. In a network, a server might contain user files that should be accessible from several machines. In this case, the server's copy of the /home directory is remotely mounted onto a local /home file system.
/lib	Symbolic link to the /usr/lib directory, which contains architecture-independent libraries with names in the form lib*.a.
/sbin	Contains files needed to boot the machine and mount the /usr file system. Most of the commands used during booting come from the boot image's RAM disk file system; therefore, very few commands reside in the /sbin directory.
/tmp	Serves as a mount point for a file system that contains system-generated temporary files.
/u	Symbolic link to the /home directory.
/usr	Serves as a mount point for a file system containing files that do not change and can be shared by machines (such as executable programs and ASCII documentation). Standalone machines mount a separate local file system over the /usr directory. Diskless and disk-poor machines mount a directory from a remote server over the /usr file system.
/var	Serves as a mount point for files that vary on each machine. The /var file system is configured as a file system because the files that it contains tend to grow. For example, it is a symbolic link to the /usr/tmp directory, which contains temporary work files.



Caselet

Windows File Systems

The configuration of my system is as follows: Pentium III, 40 GB hard disk, Samsung CD-RW and D-link external modem. My system is very slow and a lot of applications have expired due to shareware and demo software. I formatted my system about three years ago and would like to reformat it now. I know there are three types of formats: fat, fat32, and ntfs that are possible. Could you explain what they mean and which one I ought to opt for? Also, what operating system should I install?

Srinath Rao

The newer operating systems such as Windows 2000 and Windows XP support three file systems for formatting the hard disk. They are FAT, FAT32 and NTFS file systems.

Basically, a file system is a system for organising directories and files, generally in terms of how it is implemented in the disk operating system.

Contd...

Notes

FAT: The fat (file allocation table) file system is an old, out-dated file system also called FAT16. Old operating systems such as MS-DOS, Windows 3.1 and Windows 95 (prior to OSR2) used FAT file systems. Windows 2000 supports the maximum size of fat file system, 4 GB only. So it is advisable not to use FAT file system.

FAT32: Operating systems such as Windows 95 OSR2, Windows 98, Me, 2000, and Windows XP support fat32 partitions. The fat32 file system is more efficient than fat because it supports larger partition sizes. For example, Windows 2000 will format FAT32 partitions up to 32GB in size. If you would like to use dual boot with Windows 98 or Windows Me with later OS such as Windows 2000/XP/2003, then you will have to use FAT32 instead of NTFS file systems because Windows 98/Me will not support NTFS file systems. You can convert the FAT32 file system to NTFS file system anytime. After converting to NTFS you cannot revert back to the FAT32 file system.

To convert a FAT or FAT32 file system to NTFS, please do the following: click Start - Run - type "convert C: /fs:ntfs" and click ok. Here C refers to the c drive to convert.

NTFS: The Windows NT File System (NTFS) is the latest file system supported by Windows 2000/XP/2003. It is a highly reliable and recoverable file system. The additional features of NTFS file systems are that they support NTFS compression (compression attribute), "Encrypting File System (EFS)" that is Encrypting attribute, Disk quotas, mounting volume as directory, local file and folder security. It formats NTFS partitions up to 2 TB. If you would like to use only one operating system (or dual boot without Windows 98/Me) then it is advisable always to use the NTFS file system.

Regarding the Operating System, Windows 98 is obsolete. Microsoft itself has withdrawn support for Windows 98. You can install either Windows 2000 or Windows XP. If you are installing Windows 2000, please make sure service pack 4 is installed and for Windows XP service pack 2 is essential. It is advisable to use the latest operating system Windows XP with service packs and required Windows updates.

Source: <http://www.thehindubusinessline.in/ew/2006/06/26/stories/2006062600200400.htm>

3.6 Summary

File is a named collection of data stored in a device. File manager is an integral part of the operating system which is responsible for the maintenance of secondary storage. File system is a set of abstract data types that are implemented for the storage, hierarchical organization, manipulation, navigation, access, and retrieval of data. Disk file system is a file system designed for the storage of files on a data storage device, most commonly a disk drive, which might be directly or indirectly connected to the computer. Flash file system is a file system designed for storing files on flash memory devices. Network file system is a file system that acts as a client for a remote file access protocol, providing access to files on a server. Flat file system is a file system where is no subdirectories and everything is stored at the same (root) level on the media, be it a hard disk, floppy disk, etc. Directory is simple file containing the indexing of other files, which may in turn be directories if a hierarchical indexing scheme is used.

3.7 Keywords

Directory: It is simple file containing the indexing of other files, which may in turn be directories if a hierarchical indexing scheme is used.

Disk file system: It is a file system designed for the storage of files on a data storage device, most commonly a disk drive, which might be directly or indirectly connected to the computer.

File manager: It is an integral part of the operating system which is responsible for the maintenance of secondary storage.

File system: It is a set of abstract data types that are implemented for the storage, hierarchical organization, manipulation, navigation, access, and retrieval of data.

File: It is a named collection of data stored in a device.

Flash file system: It is a file system designed for storing files on flash memory devices.

Flat file system: It is a file system where no subdirectories are present and everything is stored at the same (root) level on the media, be it a hard disk, floppy disk, etc.

Network file system: It is a file system that acts as a client for a remote file access protocol, providing access to files on a server.

3.8 Self Assessment

Fill in the blanks:

1. A flash file system is a file system designed for storing files on memory devices.
2. The contains pointers to the various blocks.
3. In a flat file system, there are no -everything is stored at the same (root) level on the media, be it a hard disk, floppy disk, etc.
4. The file would contain pointers to secondary index files, which would point to the actual data items.
5. In two level directory structure one is directory and the other is user file directory.
6. Stopping in this case would require stopping two systems – nfsfs and nfs.
7. A is a file system designed for the storage of files on a data storage device, most commonly a disk drive.
8. file systems are one technique used to introduce transaction-level consistency to file system structures.
9. The directory specifies the home or login directory for the account.
10. Information in the file is in order, one record after the other.
11. processing introduces the guarantee that at any point while it is running.
12. File systems share much in common with database technology, but it is debatable whether a file system can be classified as a
13. The file system was an early research interest of Dennis Ritchie of Unix fame.
14. The system contains information about the state of processes and threads in the system.
15. Linux kernel and provides capabilities

Notes

3.9 Review Questions

1. What is a directory? Scrutinize whether we can consider a directory as a file or not.
2. Analyze the need of flash file system with example.
3. Make distinction between file system and file manager. Give examples.
4. Examine the factors that you think influence the 'CWD'?
5. Make distinction between disk file system and flash file system. Give examples.
6. Red Hat Linux uses the User Private Group scheme, each user will be assigned to a default group consisting only of the user. Enlighten the statement.
7. What do you see as the difference between File systems and Operating Systems?
8. Make distinction between Sequential Access and Direct Access with examples.
9. Examine the advantages of the hierarchical architecture over a simple one-level indexing one.
10. Most operating systems provide a file system, as a file system is an integral part of any modern operating system. Comment.

Answers: Self Assessment

- | | | | |
|-------------------------------------|------------------|---------------------|------------------|
| 1. flash | 2. index | 3. subdirectories | 4. primary index |
| 5. master file | 6. NFS | 7. disk file system | 8. Journaling |
| 9. home | 10. Processed | 11. Transaction | |
| 12. special-purpose database (DBMS) | 13. hierarchical | | |
| 14. /proc file | 15. windowing | | |

3.10 Further Readings



Books

Andrew M. Lister, *Fundamentals of Operating Systems*, Published By Wiley
Andrew S. Tanenbaum, *Modern Operating System*, Published By Prentice Hall
Colin Ritchie, *Operating Systems*, Published By BPB Publications
Silberschatz Galvin, *Operating System Concepts*, Published By Addison sley



Online links

http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
<http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/index.html>
<http://www.pathname.com/fhs/>

Unit 4: Configuring Desktop

Notes

CONTENTS

Objectives

Introduction

- 4.1 Configuring Desktop
 - 4.1.1 Using the Desktop as-is
 - 4.1.2 Switching between Desktop Environments
 - 4.1.3 Basic Operation of the Desktop
- 4.2 Understanding the Run Levels
- 4.3 Managing Users
- 4.4 Summary
- 4.5 Keywords
- 4.6 Self Assessment
- 4.7 Review Questions
- 4.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss configuring desktop
- Learn working with desktop control center
- Understanding run levels
- Know managing users

Introduction

At the heart of all the software on your computer is the Linux kernel – the software that provides the core services that comprise an operating system.

You can work with the services that the kernel provides in a couple of ways. One way is through a terminal window and a set of text commands to manipulate the machine, much like old-time PC users used DOS to do their work.

But many people don't want to do this. They prefer a graphical user interface on top of the kernel, so they don't have to remember commands or long strings of fussy parameters – indeed, so they don't even have to know how to type!

In this unit you will study the process of configuring desktop such as switching between desktop environments, basic operations of desktop, managing users, etc.

4.1 Configuring Desktop

At the heart of all the software on your computer is the Linux kernel – the software that provides the core services that comprise an operating system.

You can work with the services that the kernel provides in a couple of ways. One way is through a terminal window and a set of text commands to manipulate the machine, much like old-time PC users used DOS to do their work.

But many people don't want to do this. They prefer a graphical user interface on top of the kernel, so they don't have to remember commands or long strings of fussy parameters – indeed, so they don't even have to know how to type!

With Windows, the operating system and the GUI on top of it are bound together so tightly that it's virtually impossible to run the OS without the GUI anymore. Furthermore, with Windows, you don't have a choice of GUIs. This is not the case with Linux.

The X Window System (commonly known simply as "X") is a piece of software that runs on top of the Linux kernel and provides windowing capabilities. X has been through a number of versions, with version 11 finally gaining critical mass and developing widespread are GNOME and KDE, and popular distributions include at least one of these desktops (and usually both).



Example: SuSE ships with KDE "out of the box." Red Hat upset a lot of folks when it released Red Hat Linux 8.0, which presented the user with a combination of GNOME and KDE, using components of both in a theme called "Bluecurve." Fedora Core continues that tradition, using Bluecurve as the default environment and theme.

You can install either GNOME or KDE, or both, during installation. Depending on your choice, you will get the applications that fit with the desktop environment you selected.



Example: If you selected GNOME, you'll get OpenOffice.org; if you selected KDE, you'll get KOffice. If you chose both, you'll get all applications from both desktop environments.

I'll concentrate on GNOME and Bluecurve in this book, mentioning KDE when appropriate. If you're interested in other window managers, you can find a central repository at acceptance. This version is known as "X11" and you may have already seen references to it.

X11 has been through a number of upgrades, each of which has attempted to maintain backward compatibility. The current release number is 6, so you may also see references to "X11R6." The first time you saw that string – X11R6 – it probably seemed obscure to the point of distraction, but now it's not so bad, is it?

X is a foundation upon which people can create software applications that take advantage of those windowing capabilities. One particular breed of applications is known as "window managers" – and a number of people and groups have created them. As a result, there isn't just one GUI available for Linux; there are many – about a dozen, all told. Names of popular window managers that you might have heard of include Enlightenment, Metacity, Sawfish, and WindowMaker. But these are just window drawing packages – they handle how windows are drawn and how they appear, how they interact with each other, and so on. Everything that appears on the screen shows up in a window, and the window manager simply provides the engine to control the windows.

But there's more to a desktop than just windows. As you've already seen, there's a menuing system and a panel, to name a couple of obvious items. And there are tools that provide a GUI interface to the operating system (not just end-user applications). The rise in the number of window managers has brought forth a "super-class" of tools called desktop environments. These provide a complete interface to the operating system, including menus, panels, utilities and applications, and all sorts of other fun stuff. The two most popular desktops.

4.1.1 Using the Desktop as-is

The first thing you'll notice, regardless of whether you select GNOME or KDE during installation, is that the standard Fedora Core desktop is rather spartan. Because there isn't a profit-oriented company behind any of the various desktop organizations, you won't see a desktop full of icons that have been placed there for a goodly dose of lucre.

Instead, you'll see three default icons: one that leads you to your home directory, one for trash, and one ("Start Here") that gives you a convenient location for the items you typically want to use most often. See Figure 4.1 for the basic GNOME desktop.

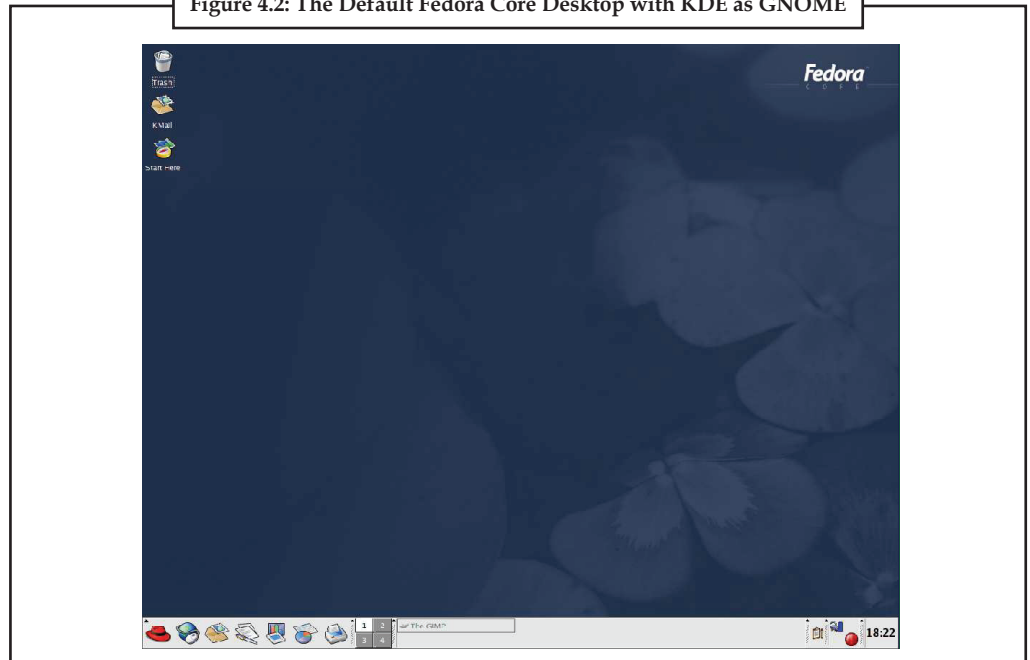
The KDE desktop isn't any more complicated (see Figure 4.1).

Figure 4.1: The Default Fedora Core desktop with GNOME



Notes

Figure 4.2: The Default Fedora Core Desktop with KDE as GNOME



The bar at the bottom of the screen is called the “panel” and works much the same way that the taskbar does in Windows. You may have noticed that the bottom of the screen looks considerably different between Figures 4.1 and 4.2, but it’s not because GNOME and KDE are that different. Rather, it’s because I personally use GNOME as my default environment, and have customized the panel to suit my preferences. Figure 4.1 shows the panel loaded up with things I like and use on a regular basis. I’m also writing this book using OpenOffice.org on Fedora Core, so many applications appear in the panel because I’m taking screen shots as I write.

Conversely, the KDE panel is less populated because I don’t use it all that often, and didn’t have a reason to heavily customize it.

4.1.2 Switching between Desktop Environments

Fedora Core comes with both GNOME and KDE installed, and GNOME is the default environment. If you want to switch from GNOME to KDE, click Main Menu | Preferences | More Preferences | Desktop Switching Tool (see Figure 4.3).

Once you’ve selected another desktop environment and clicked OK, you’ll discover that the desktop doesn’t change automatically. Instead, you have to perform an operation called “restarting the X window server,” or, sometimes more tersely, simply “restarting X.” And just how would one do that, you may wonder?

The simplest and most reliable way is to log out and log back in. When you log back in, the desktop you chose will be in charge.

Figure 4.3: The Desktop Switcher



4.1.3 Basic Operation of the Desktop

Because GNOME is the default, it is going to cover it in fair detail. However, we'll also discuss some KDE applets and features. As a result, this discussion assumes that you've installed all of GNOME and KDE.

There are four primary components to the GNOME desktop environment:

1. The desktop is the entire area on the screen except for the bar at the bottom.
2. The panel (Linux-speak for the Windows taskbar) is the strip at the bottom that contains a variety of icons and widgets.
3. The Main Menu (the Linux version of the Windows Start menu) is displayed by clicking on the red fedora icon at the far left of the panel.
4. The system applets, which provide a graphical interface to using the OS, are found through various menu options via the Main Menu.

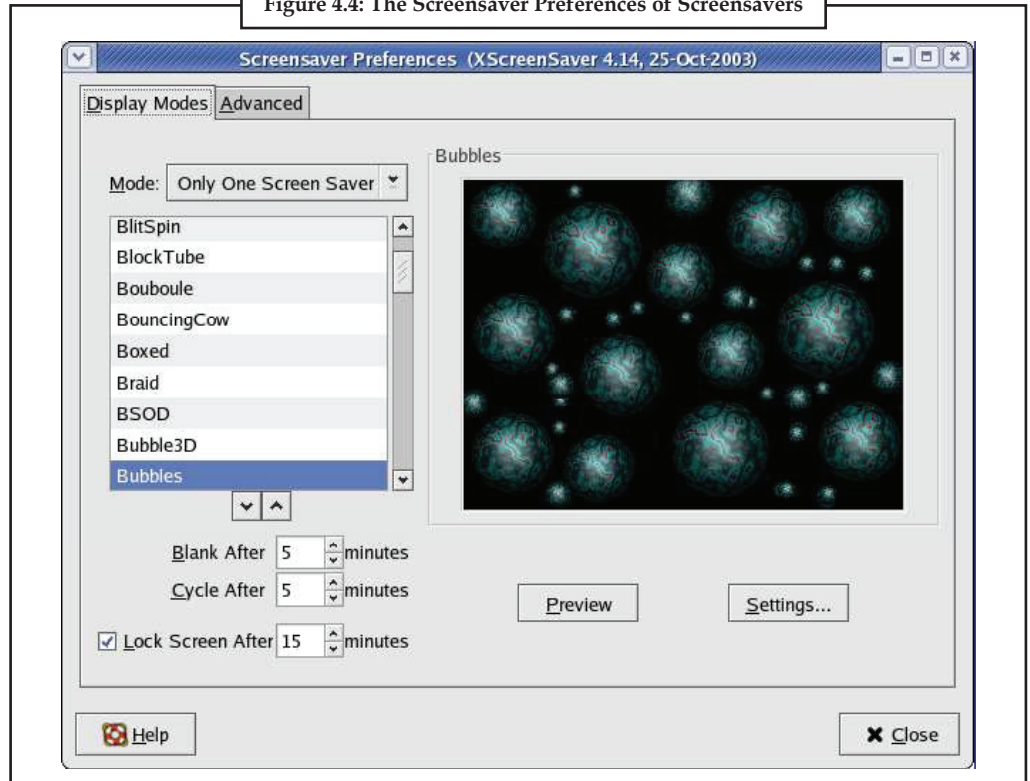
Desktop

The GNOME desktop looks and feels much like other graphical desktops, complete with icons, wallpaper (in Fedora Core, they're called Desktop Backgrounds), screensavers, and, when applications are open, windows. Double-clicking an icon on the desktop runs the program or opens the file associated with the icon. Right-clicking an open area of the desktop displays a context menu from which you can choose a different background.

Screensavers can be configured by clicking Main Menu | Preferences | Screensavers, as shown in Figure 4.4.

Notes

Figure 4.4: The Screensaver Preferences of Screensavers




Windows in GNOME work much the same as with Windows. The icons in the upper right serve to minimize, maximize (or revert to original size), and close the window.

The Window menu in the upper left has the typical Minimize, Maximize, Move, Resize, and Close options that you'd expect from using Microsoft Windows. The rollup menu item isn't found in Windows, but is known as "window shade" on Mac OS. The next couple of options in the Window menu, Put on All Workspaces, and Move to Workspace 1/2/3/4, refer to workspaces, a feature of Linux that isn't found in Windows.

You can resize windows by using the sides and corners, while clicking and dragging on the title bar moves the window around. GNOME has a couple of nice touches that you don't find in

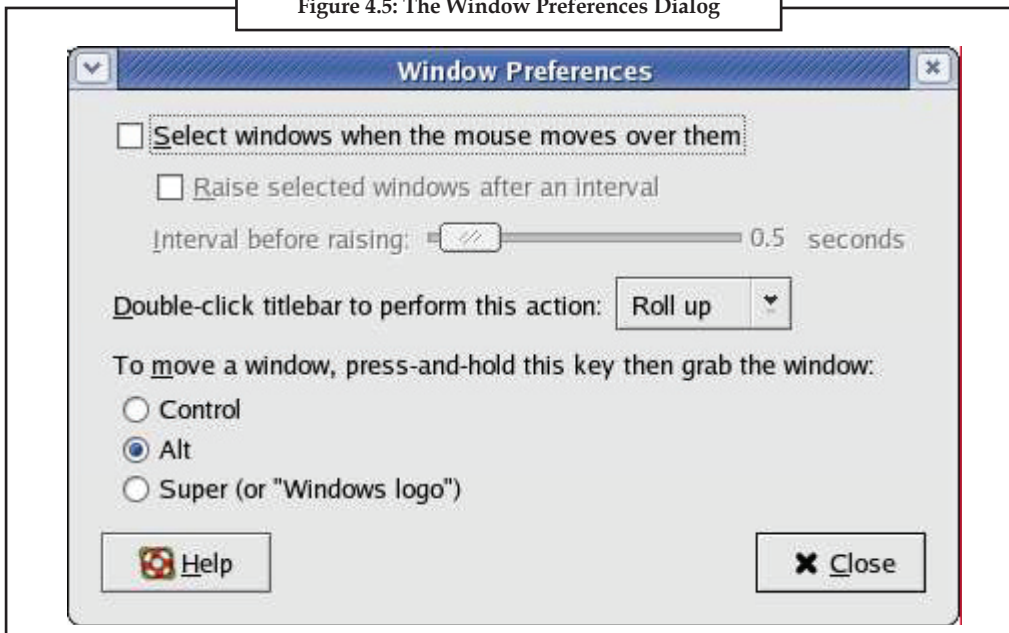
Windows; these can be configured via Main Menu | Preferences | Window Preferences, as shown in Figure 4.5.



Note Note that you can choose to automatically select a window simply by moving the mouse over it.

(I've tried this and it's always driven me crazy because it doesn't bring the selected window to the front, but that's my own personal preference.) Double-clicking the title bar can either roll up the window (like a window shade) or maximize it. And you can move a window by holding down a key of your choice and then dragging the window, and resize the window by holding down the same key while also holding down the middle mouse button. The default key for both of these operations is Alt, but you can change it in the Window Preferences dialog.

Figure 4.5: The Window Preferences Dialog



Main Menu icon (the red fedora) is used to display a menu that launches a variety of functions and applications. Next to the Main Menu icon is a series of icons for commonly used applications. The panel is automatically populated during installation with some application launchers, such as the Mozilla browser and OpenOffice.org.

The right half of my panel, shown in figure, also has some standard items and some customized items. The first icon on the left is the Workspace Switcher applet. Next to it is the System Monitor applet. Continuing to the right are a series of boxes that represent running applications. The area of the panel that contains all of the boxes is called the Taskbar. And finally, the far right of the panel is the Notification Area, where a number of icons display as appropriate. Let's look at each of these in more detail.

The Workspace Switcher, always placed on the panel during installation, is a mechanism that is new to Windows users, but once you get used to it, you'll wonder how you ever got along without it. You're probably used to the problem of having so many windows open on your desktop that it gets confusing which window does what—and using Alt-Tab to move between windows isn't as handy when you've got 15 or 20 in the list. Imagine if you could create separate desktops for groups of windows. All of the windows related to a single project would be on one desktop, while the windows that you always have open—such as your e-mail client, a Web browser, and perhaps a music jukebox or an IM application—would be contained in a separate window. A workspace is just that: a complete desktop that you can set up for a specific purpose. The Workspace Switcher allows you to switch between one workspace and another.

If you look closely at the Workspace Switcher in Figure 4.6, you'll see four squares arranged in a larger square. Each square represents one workspace. You'll see a number of small boxes inside the upper left box; those are individual application windows on the desktop in that workspace. If I'd had applications open in other workspaces, you'd see miniature windows in those workspace boxes as well, as in Figure 4.6.

Notes

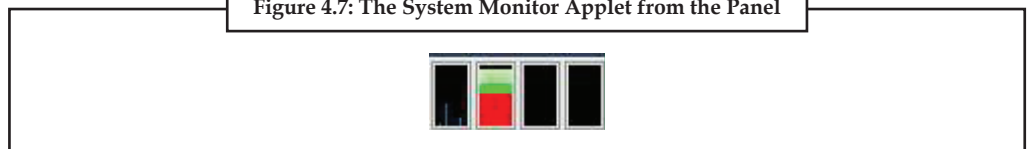
Figure 4.6: You can Navigate the Workspace Switcher with the Keyboard's Arrow Keys



You can navigate between workspaces via a mouse click, or you can use the keyboard. To do so, press Ctrl-Alt to bring forward a dialog displaying all of the windows, as shown in Figure 4.6. Keeping the Ctrl-Alt keys depressed, use the up/down/left/right arrows to move to the window you want to switch to. Note that the focus doesn't automatically move from one row to the next; if you want to move from the upper right to the lower left, you need to press the left arrow and the down arrow, not the right arrow.

The System Monitor applet (see Figure 4.7), the set of four boxes to the right of the Workspace Switcher, isn't normally placed on the panel; I'll address it in the "Customize the desktop" section later in this unit.

Figure 4.7: The System Monitor Applet from the Panel



The Taskbar displays a box for each running application, as shown in Figure 4.8.

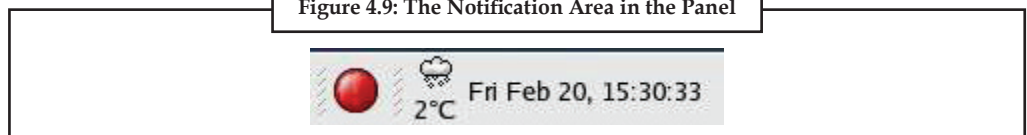
Figure 4.8: The Taskbar Displays Boxes for each Running Application



Depending on the height of the panel (which you can customize), the application boxes in the Taskbar section may show up in one or more rows. The bars shrink as their number increases, to the point that the legend may not be readable anymore. Holding your mouse over a specific box will display a tooltip with the full name.

The Notification Area displays the time and the Red Hat Network Alert Notification Tool by default, as shown in Figure 4.9. This area also displays other icons during other operations.

Figure 4.9: The Notification Area in the Panel



For example, when you print something, a printer icon will appear; when you gain root authentication (explained later), a key icon will be displayed. You can add other notification type tools there as well.

Main Menu

Notes

Clicking the Main Menu icon opens a hierarchical menu bar, as shown in Figure 4.10. Clicking any of the menu items with an arrow pointing to the right opens a subsidiary menu, and so on.

Figure 4.10: The Main Menu's items Cascade into Submenus



As you can see, menu items are grouped according to functional areas, instead of everything being grouped under one huge Program Files option, or hidden behind rarely used menu pads.

However, because it's possible to install a large number of applications in any functional area, the installation program puts the most commonly used programs for the desktop environment you've selected in the Main Menu and moves the rest to a "More... applications" submenu.



Example: If your desktop environment is GNOME, but you've installed KDE applications as well, the KDE programs will be placed on the "More..." menu.

Applets

You can control and manage your computer system by using GUI applets that are found under various menu options, similar to the applets found in Windows Control Panel. Here, again, things are a little different: Applets are separated according to function, instead of jamming them all in one place.



Example: You can choose how the interface looks on your desktop by changing various options under Main Menu | Preferences. System settings, on the other hand, are found under (gasp) Main Menu | System Settings.

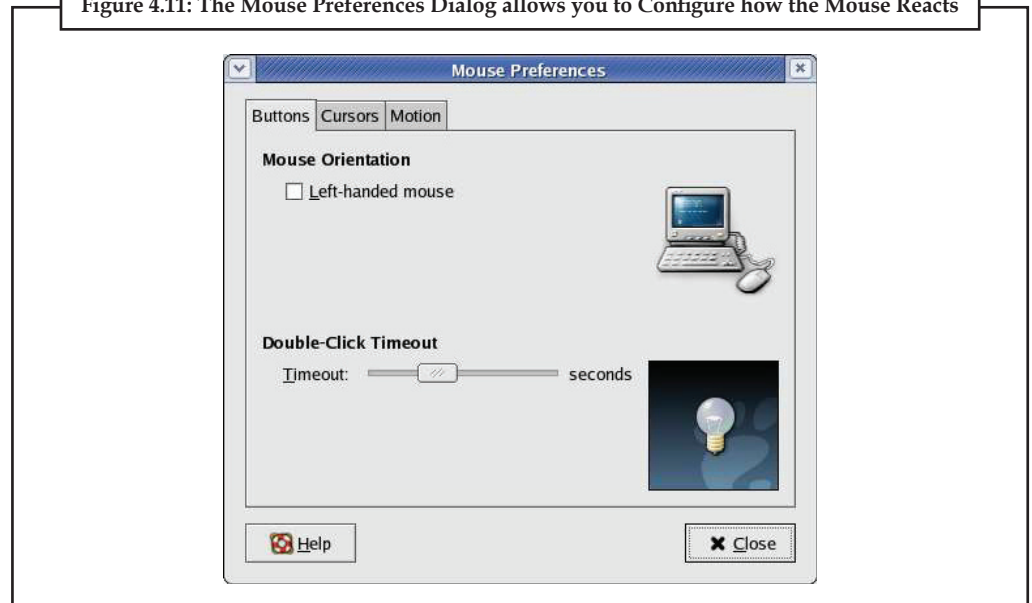
If you open both Main Menu | Preferences, and Main Menu | System Settings, you'll see a number of options that appear to be similar or downright identical. For example, consider the Login Photo option under Preferences and Login Screen under Systems Settings; also Network

Notes

Proxy under Preferences and Network under Settings. And there are “Keyboard” and “Mouse” options on both. The options, however, lead to dialogs that provide different types of choices.

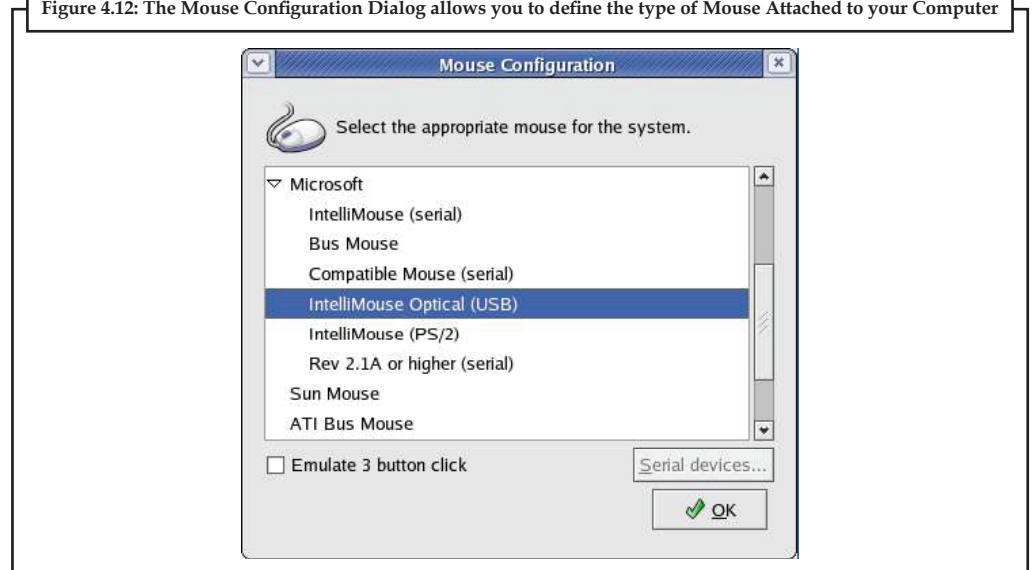
Figure 4.11 shows the Mouse Preferences, where you see options for using a left-handed mouse and setting the double-click speed.

Figure 4.11: The Mouse Preferences Dialog allows you to Configure how the Mouse Reacts



In Figure 4.12, the Mouse Configuration dialog, you can choose which type of mouse is being used with the computer.

Figure 4.12: The Mouse Configuration Dialog allows you to define the type of Mouse Attached to your Computer



As a power user, you don't need a “This is the ON switch” guide, leading you through the nose on each of these applets. Their usage will be obvious as you look through them. Thus, I suggest that you just spend a few moments scrolling through the various menu options under

Preferences, System Settings, and System Tools to get a feel for what's available via a GUI oriented applet. By the way, it's important to mention again that these applets are simply graphical

interfaces to programs that provide the ability to configure and customize the system. Every function you perform via an applet in the menu can also be done via commands. In fact, in many instances, the commands offer additional options that haven't yet migrated to the graphical applets.

Desktop Control Centre

Key features

Extreme performance, full control

1. Overclock CPU, memory, and bus speeds for optimal performance
2. Integrated auto-tuning makes system optimization easy while maintaining system stability

4.2 Understanding the Run Levels

If you're one of those who took a chance and got one of the Caldera Previews or got a Red Hat distribution on your system, one of your original thoughts may have been the same as mine: What happened to `/etc/rc.local`? Where am I supposed to put my custom commands? [One answer: `/etc/rc.d/rc.local` is available on Red Hat systems – ED] What if I don't want the HTTP server to start?

For those of you out there who administer Sun Solaris machines, this looks quite familiar. But I was just scratching my head for a while until I wound up administering a system, and it all became clear. Time to share the knowledge.

The idea behind the setup is to make everything script-based. For each run level, scripts are run to start each individual service, instead of having a few large files to edit by hand. These scripts are located in `/etc/rc.d/init.d`, and most take as an option **start** or **stop**. This is to allow the specific programs to start (on bootup) or stop (on shutdown).

This setup involves a bunch of directories under `/etc/rc.d/`. These are:

rc0.d Contains scripts to run when the system shuts down. Technically, `halt` or `shutdown` bring the system to runlevel 0. This directory is mostly made up of kill commands.

rc1.d through rc3.d Scripts to run when the system changes runlevels. Runlevel 1 is usually single-user mode, runlevel 2 is for multi-user setup without NFS, and runlevel 3 is full multi-user and networking.

Runlevel 4 is typically unused.

rc5.d Scripts to start the system in X11 mode. This is the same as runlevel 3, with the exception that the `xdm` program starts, which provides a graphical login screen.

rc6.d Scripts to run when the system reboots. These scripts are called by a `reboot` command.

init.d Actually contains all of the scripts. The files in the `rc?.d` directories are really links to the scripts in the `init.d` directory.

The Boot Sequence

Now that we know where files are located, let's look at what happens in a normal Red Hat boot sequence.

Once the system boots, `/etc/rc.d/rc.sysinit` is run first. The starting runlevel (specified in `/etc/inittab`) is found, and the `/etc/rc.d/rc` script is run, with the sole option being the runlevel we want to go to. For most startups, this is runlevel 3.

Notes

The rc program looks in the /etc/rc.d/rc3.d directory, executing any **K*** scripts (of which there are none in the rc3.d directory) with an option of **stop**. Then, all the **S*** scripts are started with an option of **start**. Scripts are started in numerical order – thus, the S10network script is started before the S85httpd script. This allows you to choose exactly when your script starts without having to edit files. The same is true of the **K*** scripts.

Let’s look at what happens when we switch runlevels – say from runlevel 3 (full networking and multi-user mode) to runlevel 1 (single-user mode).

First, all the **K*** scripts in the level to which the system is changing are executed. My Caldera Preview II (Red Hat 2.0) setup has seven K scripts and one S script in the /etc/rc.d/rc.1/ directory. The K scripts shut down nfs, sendmail, lpd, inet, cron, and syslog. The S script then kills off any remaining programs and executes **init -t1 S**, which tells the system to really go into single-user mode.

Once in single-user mode, you can switch back to full multi-user mode by typing **init 3**.

Side-stepping init

There are two additional points I can make here.

First, you can selectively start and stop scripts, even those not native to your runlevel. Executing scripts in /etc/rc.d/init.d/ with an option of **start** or **stop** will start up or stop the programs or services which the script controls. This allows you to turn off NFS from runlevel 3 while keeping all other systems active, for example. Similarly, you can start NFS back up when you are ready.



Caution Stopping NFS in this case would require stopping two systems – nfsfs and nfs.


The nfsfs script will mount or unmount any of the NFS-mounted file-systems listed in your /etc/fstab. The nfs script would then shut down the processes associated with NFS, in this case mountd and nfsd.

So the proper procedure for shutting down NFS would be:

```
# /etc/rc.d/init.d/nfsfs stop
Unmounting remote filesystems.
# /etc/rc.d/init.d/nfs stop
Shutting down NFS services: rpc.mountd rpc.nfsd
#
```

And starting NFS would be:

```
# /etc/rc.d/init.d/nfs start
Starting NFS services: rpc.mountd rpc.nfsd
# /etc/rc.d/init.d/nfsfs start
Mounting remote filesystems.
#
```



Task “The rc program looks in the /etc/rc.d/rc3.d directory, executing any K* scripts.” Comment

4.3 Managing Users

Notes

Almost all Linux distribution comes with its own set of GUI tools to manage users and groups.



Example:

1. Red Hat comes with redhat-config-user
2. Suse comes with Yast
3. Debian Linux and other distro come with users-admin GUI tool set
4. FreeBSD with sysinstall
5. Solaris comes with Solaris Management Centre (SMC)

Command Line User Management

Command line utilities let you create, modify, delete, and list both user and group accounts.

User info

The `id` command prints information for a certain user. Use it like this:

```
# id username
```

To Create a New User

```
# useradd -c "My Example User" username
```

```
# passwd username
```

The created user is initially in an inactive state. To activate the user you have to assign a password with `passwd`. Some useful `useradd` options include the following:

`-c` : sets a comment for the user.

`-s` : is used in order to define the user's default login shell. If not used, then the system's default shell becomes the user's default login shell.

`-r` : creates a user with UID<500 (system account)

`-d` : sets the user's home directory. If not used, the default home directory is created (`/home/username/`)

`-M` : the home directory is not created. This is useful when the directory already exists.

To create a user that does not have the ability to login to a shell, issue the following commands:

```
# useradd -c "This user cannot login to a shell" -s /sbin/nologin username
```

```
# passwd username
```

Change the User's Password

To change a user's password:

```
# passwd username
```

If it's used without specifying a username, then the currently logged in user's password is changed.

Notes



Did u know? What will happen if user account is used without specifying a username?

Add a User to a Group

Usermod is used to modify a user account's settings. Check the manpage for all the available options. One useful use of this command is to add a user to a group:

```
# usermod -a -G group1 username
```

The -a option is critical. The user is added to group1 while he continues to be a member of other groups. If it's not used, then the user is added only to group1 and removed from any other groups. So, take note.

Remove a User from a Group

Removing a user from a group is a bit trickier. Unfortunately, there is no direct command, at least not in Fedora or RHEL, that can do that from command line. At first you need to get a list of groups that your user is a member of:

```
# id -nG username
group1 group2 group3 ....
```

Then you need to put all these groups as a comma-separated list to the usermod -G option, except for the group from which you want the user to be removed. So, to remove the user from group2, issue the command:

```
# usermod -G group1,group3,... username
```

Lock and Unlock User Accounts

Other common usermod uses are to lock and unlock user accounts. To lock out a user:

```
# usermod -L username
```

To unlock the user:

```
# usermod -U username
```

Delete a User

Userdel is used to delete a user account. If the -r option is used then the user's home directory and mail spool are deleted too:

```
# userdel -r username
```

Create a New Group

To create a new group, issue the command:

```
# groupadd groupname
```

The -r option can be used to create a group with GID<500 (system).

Change a Group's Name

Groupmod can be used to change a group name:

```
# groupmod -n newgroupname groupname
```

Delete a Group

Notes

Groupdel can delete a group:

```
# groupdel groupname
```

In order to delete a user's primary group (usually this is the group with name equal to the username) the respective user must be deleted previously.

You can find more info in the manpages, but these will do in most cases.

User LinuxConf to Manipulate Users and Groups

Linuxconf is a utility that allows you to configure and control various aspects of your system, and is capable of handling a wide range of programs and tasks. Fully documenting Linuxconf could be a separate book in its own right and certainly more than we can cover in this unit. So we'll focus on those areas that address common tasks such as adding new users and getting connected to a network.

Linuxconf allows you to configure and control various aspects of your system. After configuring your systems settings through Linuxconf, the changes are not activated immediately. You must activate the changes by choosing File => Act/Changes from the pulldown menu in the GUI version of Linuxconf, clicking on an Accept button in Web-based Linuxconf, or selecting the Accept button in text-mode Linuxconf.

Linuxconf has four user interfaces:

1. **Text-based:** Using the same user interface style as the Red Hat Linux text-mode installation program, the text-based interface makes it easy to navigate your way through Linuxconf if you aren't running X. If you are running X, you can switch to a virtual console, log in as root, and type linuxconf to bring up text-mode Linuxconf.

Use the [Tab] and [arrow] keys to navigate the text-mode screens. A down arrow on a line indicates that a pulldown menu exists on that line. The [Ctrl]-[X] key combination will make pulldown menus appear.

2. **Graphical User Interface (GUI):** Linuxconf can take advantage of the X Window System. Red Hat Linux includes a GUI interface for Linuxconf called gnome-linuxconf.

This document will display Linuxconf screens using the gnome-linuxconf interface, but you shouldn't have any trouble using the other interfaces with the instructions provided here.

3. **Web-based:** A Web-based interface makes remote system administration easy; it can also be displayed with the Lynx text-mode browser.

To use the Linuxconf Web interface, use your browser to connect to port 98 on the machine running Linuxconf (i.e., http://your_machine:98).


Before you use the Web-based interface, you'll need to configure Linuxconf to allow connections from the machine running the browser. See the section called Enabling Web-Based Linuxconf Access for instructions on enabling Web access to Linuxconf.

4. **Command line:** Linuxconf's command-line mode is handy for manipulating your system's configuration in scripts.

Notes

Linuxconf will start in either character-cell or X mode, depending on your DISPLAY environment variable. The first time you run Linuxconf, an introductory message will be shown; although it is only displayed once, accessing help from the main screen will give you the same basic information.

Linuxconf includes some context-specific help. For information on any specific aspect of Linuxconf, select Help from the screen you'd like help with.



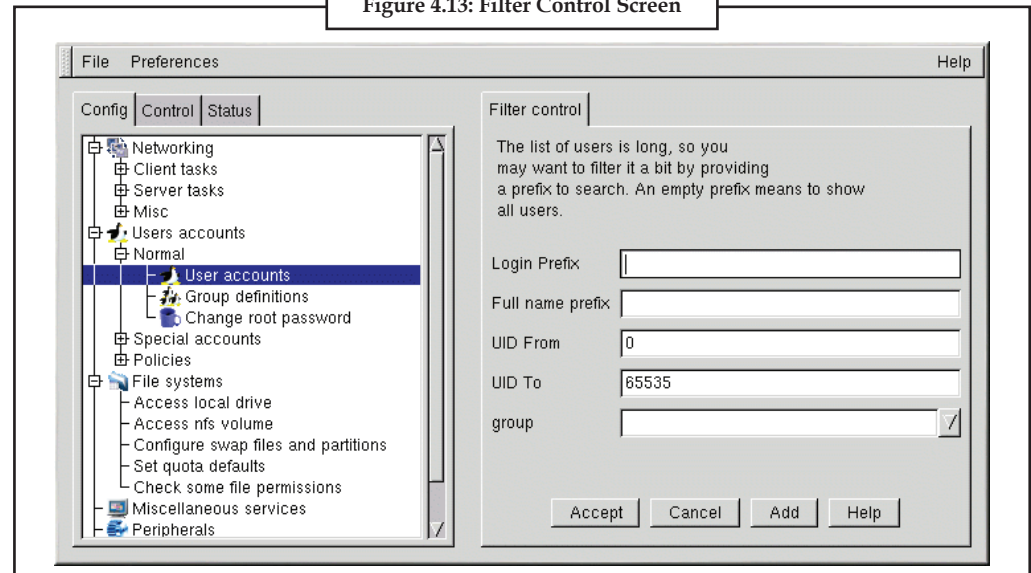
Note Not all help screens are complete at this time; as help screens are updated, they will be included in subsequent versions of Linuxconf.

Adding a User Account

Adding a user is one of the most basic tasks you will encounter in administering your system. To add a user:

Open Config => Users accounts => Normal => User accounts. Linuxconf may show you a filter screen.

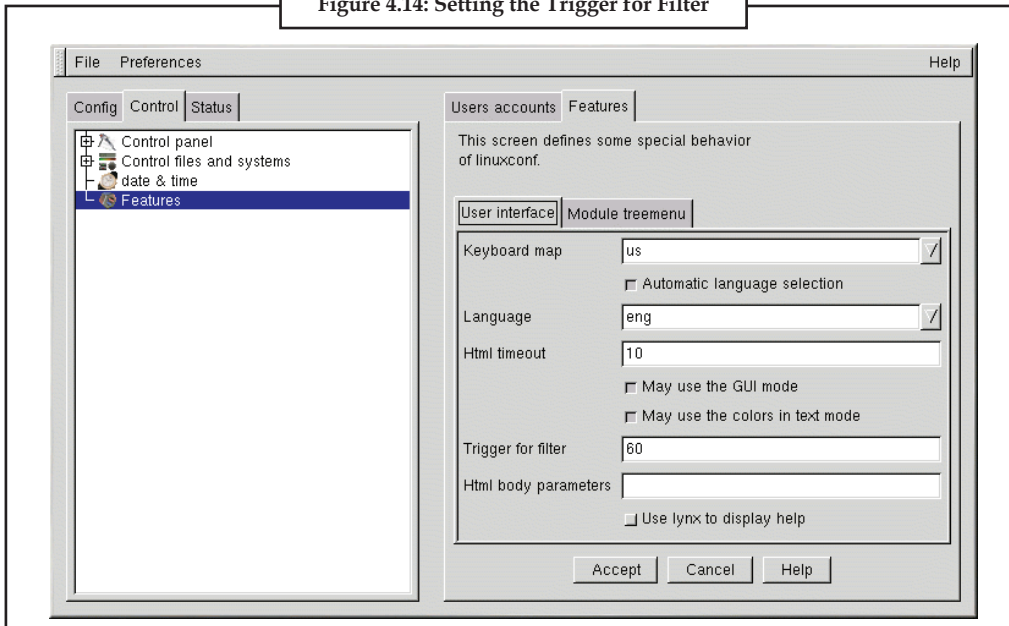
Figure 4.13: Filter Control Screen



You can use the filter screen to select a smaller range of accounts than the full list. To get the full list, select Accept without changing any of the parameters. For detailed information on the various filters, select the Help button on the Filter control screen. Once you've applied or bypassed the filter, you'll see the Users accounts tab.

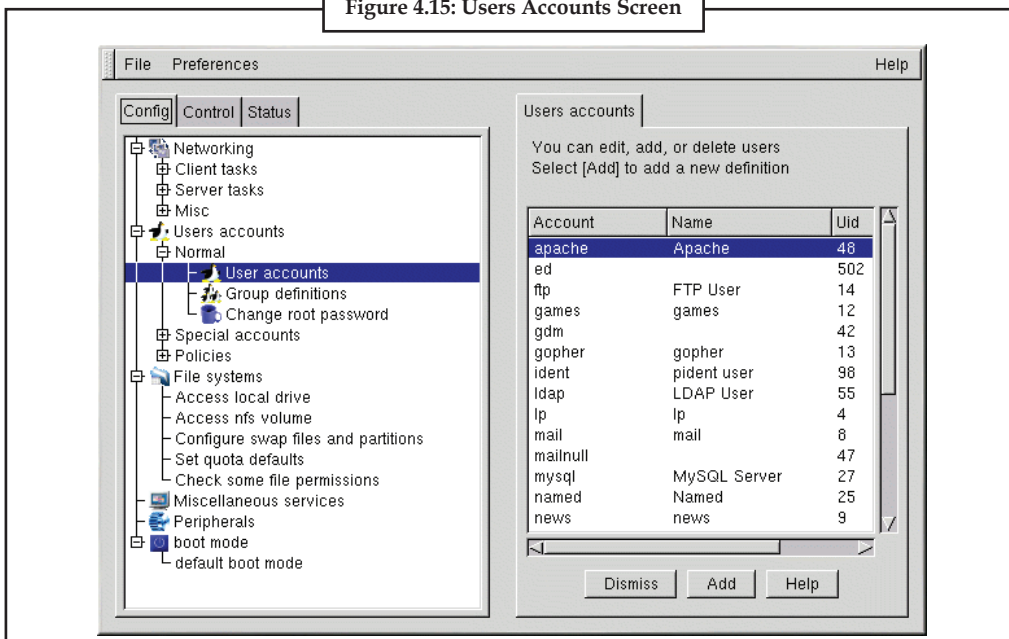
You can control the filter using Control => Features. You'll see the Features tab, which allows you to set the Trigger for filter parameter, as shown in Figure 4.14.

Figure 4.14: Setting the Trigger for Filter



The Trigger for filter field sets the number of entries that will pop up a filter screen.

Figure 4.15: Users Accounts Screen

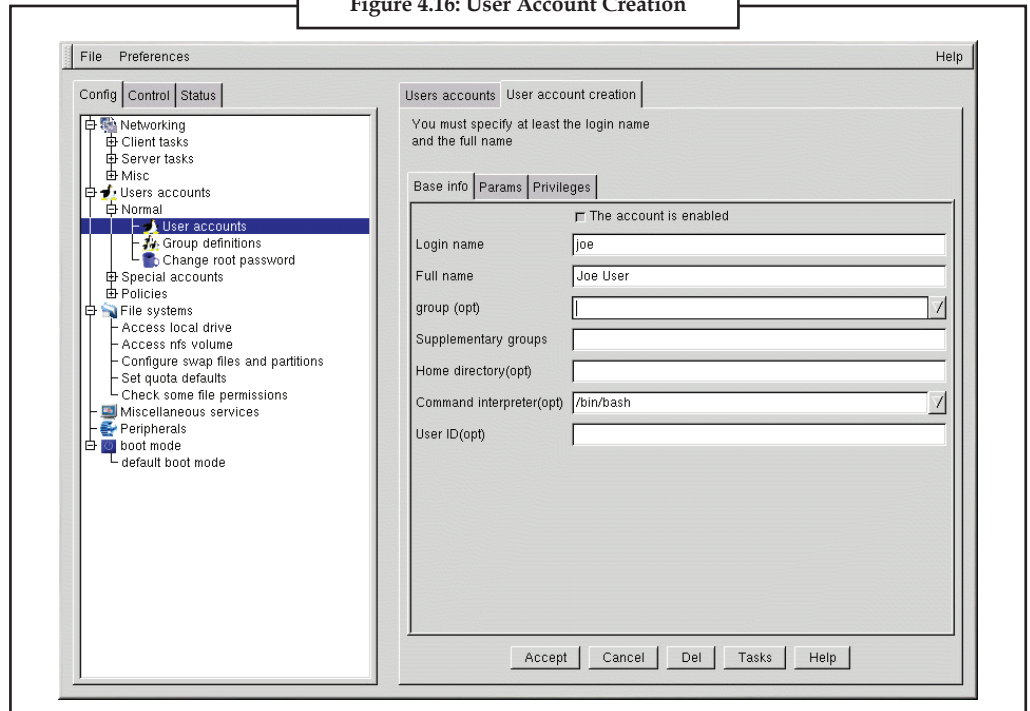


Select Add. This will open the User account creation tab (see Figure 4.16).

The User account creation screen includes the Base info, Params and Privileges sections. Only the Login name is required, but you should be aware of the other fields, which you may or may not want to fill in.

Notes

Figure 4.16: User Account Creation



Base info for User Accounts

The Login name is the name of the account and is usually all lowercase letters. First or last names, initials or some combination thereof are fairly common login names. For a user named John T. Smith, smith, john, jts, or jsmith would be common user names. Of course spike or something else works just fine, too. You can also use numbers, so jts2 would be fine if you had a second person with the same initials. There is no default for this field.

The Full name is the name of the user or the account. For an individual, it would be their name, John T. Smith for example. If the account represents a position rather than a person, the full name might be the title. So an account called webmaster might have a full name of Red Hat Webmaster or just Webmaster. There is no default for this field.

Since Red Hat Linux uses the User Private Group scheme, each user will be assigned to a default group consisting only of the user. For more information on User Private Groups, see the Official Red Hat Linux Reference Guide.

In the Supplementary groups field, you can specify additional groups. Group names should be separated by spaces. The default for this field is blank, meaning no supplementary groups are specified.

The Home directory specifies the home or login directory for the account. The default is /home/login, where login is replaced by the login name. A home directory is your starting point in the directory structure when you log in, or if in X, for each Xterm window opened. This is also where account specific preference files are stored.

The Command interpreter is the default shell for the account. The bash shell is the default shell for Red Hat Linux.

The UserID (UID) is the number associated with each user account. This is automatically generated by the system when the account is created, so just leave this field blank. The system uses the UID to identify an account.



Task

Write the steps to add the user account.

Params for User Accounts

The Params are used for password and account management. By default, all of the settings are Ignored, so they are unused. Must keep # days sets a minimum number of days for a user's password.

The Must change after # days field can be set to make a user's password expire after a certain number of days. If you want to warn them that the password is going to expire, the Warn # days before expiration field should be used.

If you'd like to have their account set to expire after a certain number of days, use the Account expire after # days field. You could alternatively set an Expiration date.

Privileges for User Accounts

In the Privileges section, you can grant access and/or control over various aspects of system configuration. As a default, regular users are denied all privileges on this screen. You may instead choose to grant or to silently grant them specific privileges. The difference between Granted and Granted/silent is that if the privilege is granted, Linuxconf will ask for the user's password before allowing them the privilege. If the privilege is granted silently, Linuxconf will not prompt for their password.

Generally, careful system administrators won't grant users any system configuration privileges unless it is absolutely necessary. If you do grant privileges, be careful when granting them silently. If a user with silently granted privileges logs into his/her machine and walks away, their privileges are wide open for the next person who sits down at their desk. Silently granted privileges are less risky if used on machines in a physically restricted area.

May use Linuxconf: The user is allowed to access all of Linuxconf's capabilities, and they can set up or change linuxconf parameters. Note that use of linuxconf is separate from the privilege of activating configuration changes. System administrators might want to grant the use of Linuxconf, but deny the activation privilege, so that the sysadmin has a final "yes/no" on whether to activate any configuration changes.

May activate config changes: After you change a parameter in Linuxconf, at some point you'll have to indicate to Linuxconf that the changes you made should be applied. Depending upon the flavor of Linuxconf that you're using, you might do this by choosing File => Act/Changes from the pulldown menu in the GUI version of Linuxconf, or clicking on an Accept button in Web-based Linuxconf, or selecting an Accept button in text-mode Linuxconf, etc.

You can grant the privilege of activating changes to a user. In that case, the user will be able to activate any changed system configuration parameters in Linuxconf.

May shutdown: A user can be granted the right to shutdown the system. Note that Red Hat Linux is set in /etc/inittab to cleanly shutdown following the [Ctrl]-[Alt]-[Del] keystroke combination.

You can also grant the user the privileges to switch network modes, to view system logs, and even give someone superuser equivalence.

Once you have entered the login name and any other desired information, select the Accept button at the bottom of the screen. If you decide against creating a new user, select Cancel instead.

Notes

When you click on Accept, Linuxconf will prompt you to enter the password. You'll have to re-type the password, to prevent unusable passwords caused by typos. Passwords must be at least six characters in length, but you can increase the required length and set other parameters for users' passwords at the Users Accounts => Policies => Password & Account Policies screen.

Good passwords contain a combination of letters, numbers, and special characters. A password should use both upper case and lower case letters. Don't use your username, your anniversary, your social security number, your dog's name, your middle name or the word root. Don't use any variation of a word associated with your account or with yourself. Don't use a word that can be found in a dictionary; dictionary words are easy to crack.

A simple technique for creating a password is to use the first letters from each word of a phrase that is familiar to you (a line from a favorite song might be appropriate). Make a few letters uppercase, and insert a few numbers and/or special characters in place of letters and you'll have a decent password.

Press the Accept button again when finished. The system will let you know if it thinks the password is easy to crack; if you get a warning message, don't use the password.

Modifying a User Account

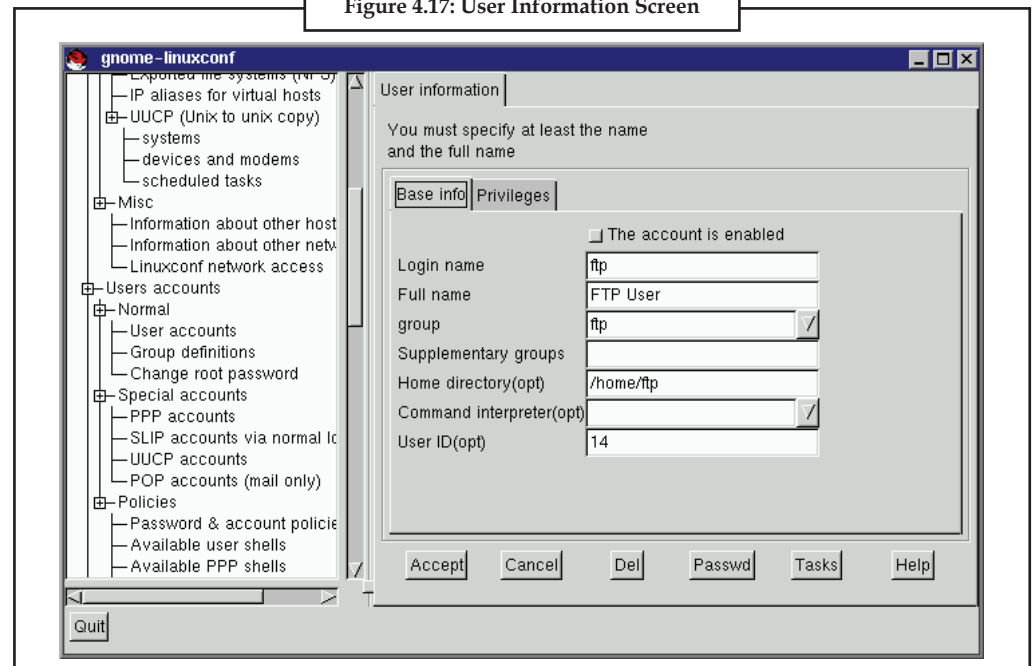
Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [User accounts]. This will open the Users accounts tab.

If you have more than 15 accounts on the system, Linuxconf will provide you with a filter screen. You can use this to select a smaller range of accounts than the full list. To get the full list, select Accept without changing any of the parameters. For detailed information on the various filters, select the Help button on the Filter control screen.

Select the account you wish to modify. This will open the User information tab.

Figure 4.17: User Information Screen



On the User information screen, the information can be changed as desired. To implement the changes select Accept. If you decide against making any changes select Cancel. This guarantees that no changes are made.

Changing a User's Password

Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [User accounts]. This will open the Users accounts tab.

If you have more than 15 accounts on the system, Linuxconf will provide you with a filter screen. You can use this to select a smaller range of accounts than the full list. To get the full list, select Accept without changing any of the parameters. For detailed information on the various filters, select the Help button on the Filter control screen.

Select the account whose password you wish to change. This will open the User information tab.

Select Passwd from the options at the bottom of the screen.

Linuxconf will then prompt you to enter the new password. There is also a field called Confirmation where you will need to type the password again. This is to prevent you from mistyping the password. Passwords must be at least 6 characters in length. They may contain numbers as well as a mix of lowercase and uppercase letters. If you decide against changing the password, just hit Cancel. Once you have entered the new password select Accept.

Changing the Root Password

Because of the security implications of root access, Linuxconf requires you to verify that you currently have access to the root account.

Open Config → Users accounts → Normal → Change root password.

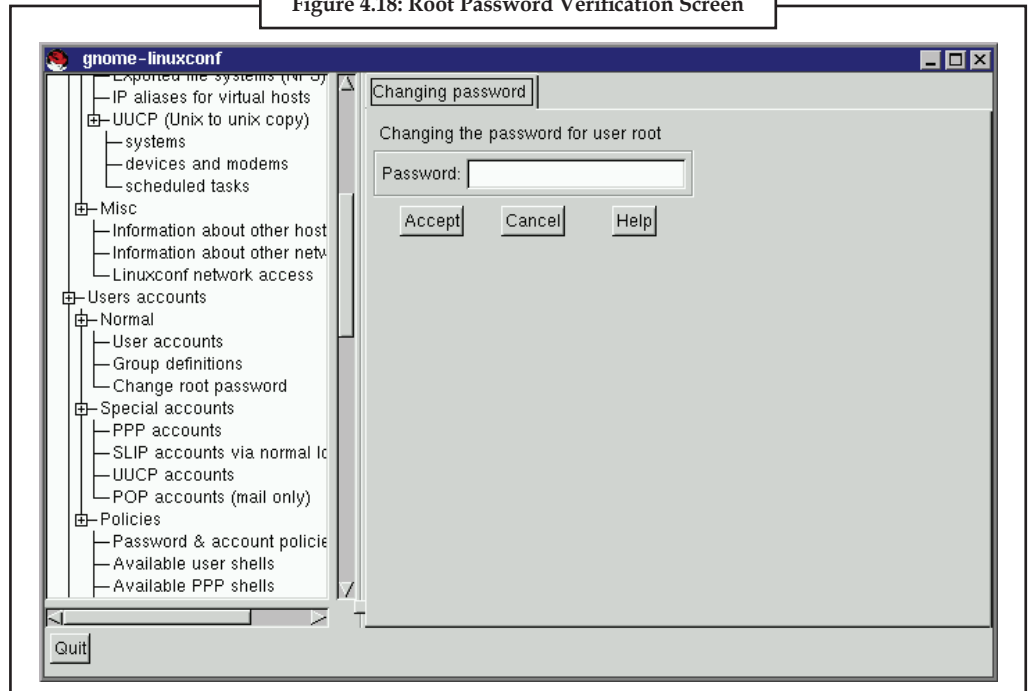
You'll first need to enter the current root password to verify access to the root account.

Once you have entered the current root password, you will be prompted for a new password. In the Confirmation field, type the password again. This is to prevent you from mistyping the password. See the section called Adding a User Account if you need guidance on choosing a password. Be sure to choose a good password! If you decide against changing the root password, just select Cancel. Once you have entered the new password, select Accept.


The screen is a little confusing because neither the title, nor the description really explains the screen's purpose. Linuxconf seems to be asking for the new password, which isn't actually the case. Instead, Linuxconf wants the current root password to verify access to the root account. Linuxconf does require root access to run, but once running there's nothing to keep anyone from sitting down at the computer if the person using Linuxconf steps out for a minute. The potential pitfalls are extensive! If the person who was originally using Linuxconf logs out of root, they won't be able to get back into it. A lack of validation would also give free reign over the computer to whoever had changed root's password.

Notes

Figure 4.18: Root Password Verification Screen



Once you have entered root’s current password, it will prompt you for a new password. There is also a field called Confirmation where you will need to type the password again. This is to prevent you from mistyping the password. Passwords must be at least 6 characters in length. They may contain numbers as well as a mix of lowercase and uppercase letters. If you decide against changing the root password, just hit Cancel. Once you have entered the new password select Accept.



Note Change takes place immediately and is effective not only for logging in as root, but also for becoming root using the su command.

Disabling a User Account

Why disable an account? Good question! There’s no single answer, but we can provide some reasons why this option is available. The biggest reason is security. For example, you may have created a special account to be used by clients, co-workers, or friends to access specific files on your system. This account gets used from time to time, but should only be used when you know there’s a need. Leaving an unused account around is a target for people who’d want to break into your system. Deleting it requires you to recreate it every time you want to use it. Disabling an account solves both problems by allowing you to simply select or de-select a check-box.

To disable an account:

Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [User accounts].

De-select the check-box that states that The account is enabled. Select the Accept button at the bottom of the window and you’re all set.

The account is disabled and can be enabled later using a similar method.

Enabling a User Account

By default, all newly created user accounts are enabled. If you need to enable an account, you can use Linuxconf to do it.

Start Linuxconf by typing `linuxconf` at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [User accounts].

Select the account you want to enable.

Select the the account is enabled check-box and then select Accept at the bottom of the screen.

Deleting a User Account

While there are a couple options that let you retain files associated with an account, any information or files deleted are gone and effectively unrecoverable. Take care when using this option.

To delete an account:

Start Linuxconf by typing `linuxconf` at the shell prompt.

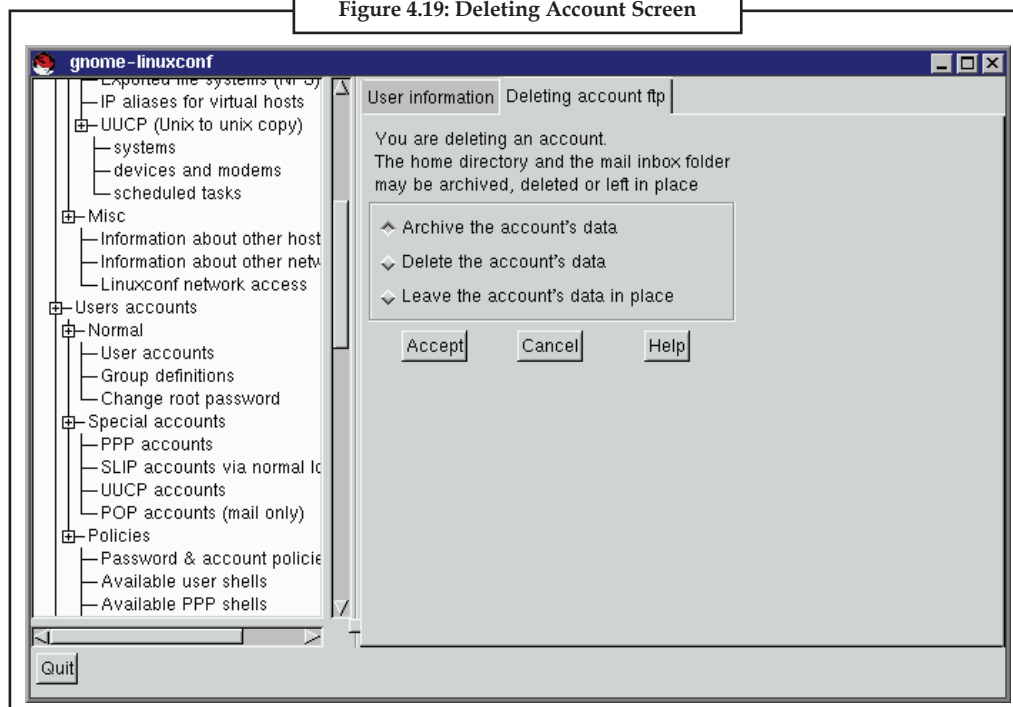
Open [Config] → [Users accounts] → [User accounts].

On the User accounts screen select the account you wish to delete.

At the bottom of the User information screen select Del to delete the account.

Linuxconf will then prompt you with a list of options.

Figure 4.19: Deleting Account Screen



The default option is to archive the account's data. The archive option has the following effects:

1. Removes the user from the user accounts list;
2. Takes everything contained in the user's home directory and archives it (using tar and gzip compression), storing the resulting file in a directory called `oldaccounts`. For an account named `useraccount` the file name would be similar to:

```
useraccount-1999-10-10-497.tar.gz
```

Notes

The date indicates when the account was deleted, and the number following it is the ID of the process that actually performed the deletion. The oldaccounts directory is created in the same place as all of your user directories, and is created automatically the first time you remove a user account using this option.

3. Files not contained in the user's home directory, but owned by that user remain. The file is owned by the deleted account's user ID (UID). If you create a new account and specifically assign it the UID of a deleted account, it will then become the owner of any remaining files.

Selecting Delete the account's data on the Deleting account <accountname>:

1. Remove the user from the user accounts list;
2. Remove the user's home directory and all its contents.

Files not contained in the user's home directory, but owned by that user will remain on the system. The file will still be owned by the deleted account's user ID (UID). If you create a new account and specifically assign it the UID of a deleted account, it will then become the owner of any such "orphaned" files.

Selecting Leave the account's data in place on the Deleting account <accountname>:

1. Remove the user from the user accounts list;
2. Leave the user's home directory (with all its files) in place.

Files and directories owned by the deleted account's user ID (UID) will remain on the system. If you create a new account and specifically assign it the UID of a deleted account, it will then become the owner of these "orphaned" files.



Case Study

P4 Desktop from Acer India

Acer India has launched a P4 desktop priced at ₹ 17,999. The Acer Power 330 series PC is based on Intel's Pentium 4 Processor with hyper-threading technology. The computer boasts a 3.06 Ghz processor with 1 MB L2 cache and 128 MB RAM, a 40 GB Hard Disk Drive. It also offers a 52x CD-ROM drive, black 104 keys keyboard, optical mouse, multiple ports, and a 15" CRT Black Monitor. The desktop runs on Linux operating system and comes with a lifetime anti-virus from E Trust.

4.4 Summary

The X Window System (commonly known simply as "X") is a piece of software that runs on top of the Linux kernel and provides windowing capabilities. X has been through a number of versions, with version 11 finally gaining critical mass and developing widespread are GNOME and KDE, and popular distributions include at least one of these desktops (and usually both). Fedora Core comes with both GNOME and KDE installed, and GNOME is the default environment. If you want to switch from GNOME to KDE, click Main Menu | Preferences | More Preferences | Desktop Switching Tool. The GNOME desktop looks and feels much like other graphical desktops, complete with icons, wallpaper (in Fedora Core, they're called Desktop Backgrounds), screensavers, and, when applications are open, windows. You can control and manage your computer system by using GUI applets that are found under various menu options, similar to the applets found in Windows Control Panel. Linuxconf is a utility that allows you to configure and control various aspects of your system, and is capable of handling a wide range of programs and tasks.

4.5 Keywords

Notes

Linuxconf: It is a utility that allows you to configure and control various aspects of your system, and is capable of handling a wide range of programs and tasks.

Main Menu: This icon (the red fedora) is used to display a menu that launches a variety of functions and applications.

Params: The Params are used for password and account management.

4.6 Self Assessment

Fill in the blanks:

1. The System is a piece of software that runs on top of the Linux kernel and provides windowing capabilities.
2. The Main Menu (the Linux version of the Windows Start menu) is displayed by clicking on the icon at the far left of the panel.
3. The Area displays the time and the Red Hat Network Alert Notification Tool by default.
4. is a utility that allows you to configure and control various aspects of your system
5. After you change a in Linuxconf, at some point you'll have to indicate to Linuxconf that the changes you made should be applied.
6. Linuxconf allows you to and control various aspects of your system.
7. The Notification Area displays the time and the Notification Tool by default.
8. Executing scripts in /etc/rc.d/init.d/ with an option of will start up or stop the programs or services which the script controls.
9. The creation screen includes the Base info, Params and Privileges sections.
10. The shell is the default shell for Red Hat Linux.
11. The are used for password and account management.
12. The bar at the bottom of the screen is called the ".....".
13. The system applets, which provide a graphical interface to using the OS, are found through various menu options via the
14. For each run level, are run to start each individual service, instead of having a few large files to edit by hand.
15. utilities let you create, modify, delete, and list both user and group accounts.

4.7 Review Questions

1. What is a directory? Scrutinize whether we can consider a directory as a file or not.
2. Analyze the need of flash file system with example.
3. Make distinction between file system and file manager. Give examples.
4. Examine the factors that you think influence the 'CWD'?

Notes

5. Make distinction between disk file system and flash file system. Give examples.
6. Red Hat Linux uses the User Private Group scheme, each user will be assigned to a default group consisting only of the user. Enlighten the statement.
7. What do you see as the difference between File systems and Operating Systems?
8. Make distinction between Sequential Access and Direct Access with examples.
9. Examine the advantages of the hierarchical architecture over a simple one-level indexing one.
10. Most operating systems provide a file system, as a file system is an integral part of any modern operating system. Comment.

Answers: Self Assessment

- | | | | |
|-------------------------------------|------------------|---------------------|------------------|
| 1. flash | 2. index | 3. subdirectories | 4. primary index |
| 5. master file | 6. NFS | 7. disk file system | 8. Journaling |
| 9. home | 10. Processed | 11. Transaction | |
| 12. special-purpose database (DBMS) | 13. hierarchical | | |
| 14. /proc file | 15. windowing | | |

4.8 Further Readings



Books

Andrew M. Lister, *Fundamentals of Operating Systems*, Published By Wiley
Andrew S. Tanenbaum, *Modern Operating System*, Published By Prentice Hall
Colin Ritchie, *Operating Systems*, Published By BPB Publications
Silberschatz Galvin, *Operating System Concepts*, Published By Addison sley



Online links

http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
<http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/index.html>
<http://www.pathname.com/fhs/>

Unit 5: Connecting to Internet

Notes

CONTENTS

Objectives

Introduction

5.1 Network Configuration Tool

5.2 Connecting to LAN

5.3 Summary

5.4 Keyword

5.5 Self Assessment

5.6 Review Questions

5.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Understand network configuration tools
- Discuss connection with LAN

Introduction

This unit covers TCP/IP networking, network administration and system configuration basics. Linux can support multiple network devices. The device names are numbered and begin at zero and count upwards. In Linux, each network device has its own name. Ethernet devices will typically have names such as 'eth0' or 'eth1'. Dial-up connections typically have names such as 'ppp0', and some wireless lan cards might have names like 'wlan0'.

5.1 Network Configuration Tool

Linux can support multiple network devices. The device names are numbered and begin at zero and count upwards.



Example: A computer running two ethernet cards will have two devices labeled /dev/eth0 and /dev/eth1.

Linux TCP/IP Network Configuration Files

File	Description
/etc/resolv.conf	List DNS servers for internet domain name resolution. Manual page for: /etc/resolv.conf
/etc/hosts	Lists hosts to be resolved locally (not by DNS). Manual page for: /etc/hosts

Contd...

Notes

/etc/nsswitch.conf	List order of host name search. Typically look at local files, then NIS server, then DNS server. Manual page for: /etc/nsswitch.conf
Red Hat/Fedora/CentOS: /etc/sysconfig/network	Specify network configuration. eg. Static IP, DHCP, NIS, etc.
Red Hat/Fedora/CentOS: /etc/sysconfig/network-scripts/ifcfg-device	Specify TCP network information.
Ubuntu/Debian: /etc/network/interfaces	Specify network configuration and devices. eg. Static IP and info, DHCP, etc.

Domain Resolution Configuration Files

1. **File: /etc/resolv.conf** - host name resolver configuration file

```
search name-of-domain.com - Name of your domain or ISP's domain if
using their name server

nameserver XXX.XXX.XXX.XXX - IP address of primary name server
nameserver XXX.XXX.XXX.XXX - IP address of secondary name server
```

This configures Linux so that it knows which DNS server will be resolving domain names into IP addresses. If using DHCP client, this will automatically be sent to you by the ISP and loaded into this file as part of the DHCP protocol. If using a static IP address, ask the ISP or check another machine on your network. Red Hat/Fedora GUI: /usr/sbin/system-config-network (select tab "DNS").

2. **File: /etc/hosts:** locally resolve node names to IP addresses

```
127.0.0.1    your-node-name.your-domain.com    localhost.localdomain
localhost

XXX.XXX.XXX.XXX node-name
```

Note when adding hosts to this file, place the fully qualified name first. (It helps sendmail identify your server correctly) i.e.:

```
XXX.XXX.XXX.XXX superserver.yolinux.com superserver
```

This informs Linux of local systems on the network which are not handled by the DNS server. (or for all systems in your LAN if you are not using DNS or NIS)


The file format for the hosts file is specified by RFC 952.

3. **Red Hat/Fedora configuration GUI:** /usr/sbin/system-config-network (select tab "Hosts").

4. **File: /etc/nsswitch.conf:** System Databases and Name Service Switch configuration file

```
hosts: files dns nisplus nis
```

This example tells Linux to first resolve a host name by looking at the local hosts file(/etc/hosts), then if the name is not found look to your DNS server as defined by /etc/resolv.conf and if not found there look to your NIS server.



Note In the past this file has had the following names: /etc/nsswitch.conf, /etc/svc.conf, /etc/netsvc.conf, ... depending on the distribution.

Fedora/ Red Hat Network Configuration Files

Notes

1. `/etc/sysconfig/network`
Red Hat network configuration file used by the system during the boot process.
2. File: `/etc/sysconfig/network-scripts/ifcfg-eth0` Configuration settings for your first ethernet port (0). Your second port is eth1.
3. File:
 - (a) `/etc/modprobe.conf` (kernel 2.6)
 - (b) `/etc/modules.conf` (kernel 2.4)
 - (c) (or for older systems: `/etc/conf.modules`)

Example statement for Intel ethernet card:

```
alias eth0 eepro100
```

Modules for other devices on the system will also be listed. This tells the kernel which device driver to use if configured as a loadable module. (default for Red Hat)

Fedora/ Red Hat Network GUI Configuration Tools

The following GUI tools edit the system configuration files. There is no difference in the configuration developed with the GUI tools and that developed by editing system configuration files directly.

TCP/IP Ethernet Configuration

1. **Network configuration:**
`/usr/sbin/system-config-network` (FC-2/3) GUI shown here --->
`/usr/bin/redhat-config-network` (`/usr/bin/neat`) (RH 7.2+ FC-1)
2. **Text console configuration tool:**
`/usr/sbin/system-config-network-tui` (Text User Interface (TUI) for Fedora Core 2/3)
`/usr/bin/redhat-config-network-tui` (RH 9.0 - FC-1)
3. **Text console network configuration tool.**
 First interface only - eth0: `/usr/sbin/netconfig`
4. **`/usr/bin/netcfg` (GUI) (last available with RH 7.1)**



Did u know? Is there no difference in the configuration developed with the GUI ?

Gnome Desktop

Gnome Desktop Network Configuration

`/usr/bin/gnome-network-preferences` (RH 9.0 - FC-3)

Notes

Proxy configuration. Choose one of three options:

1. Direct internet connection
2. Manual proxy configuration (specify proxy and port)
3. Automatic proxy configuration (give URL)

Assigning an IP Address

Computers may be assigned a static IP address or assigned one dynamically. Typically a server will require a static IP while a workstation will use DHCP (dynamic IP assignment). The Linux server requires a static IP so that those who wish to use its resources can find the system. It is more easily found if the IP address does not change and is static. This is not important for the Linux client workstation and thus it is easier to use an automated Dynamic Host Configuration Protocol (DHCP) for IP address assignment.

Static IP Address Assignment

Choose one of the following methods:

1. **Command Line:**

`/sbin/ifconfig eth0 192.168.10.12 netmask 255.255.255.0 broadcast 192.168.10.255`

Network address by convention would be the lowest: 192.168.10.0

Broadcast address by convention would be the highest: 192.168.10.255

The gateway can be anything, but following convention: 192.168.10.1



Note

the highest and lowest addresses are based on the netmask. The previous example is based on a netmask of 255.255.255.0

2. **Red Hat/Fedora GUI tools:**

`/usr/bin/ncat` Gnome GUI network administration tool. Handles all interfaces. Configure for Static IP or DHCP client. (First available with Red Hat 7.2.)

`/usr/bin/netcfg` (Handles all interfaces) (last available in Red Hat 7.1)

3. **Red Hat/Fedora Console tools:**

`/usr/sbin/system-config-network-tui` (Text User Interface)

`/usr/sbin/netconfig` (Only seems to work for the first network interface eth0 but not eth1,...)

4. Directly edit configuration files/scripts. See format below.

The `ifconfig` command does NOT store this information permanently. Upon reboot this information is lost. Manually add the network configuration to `/etc/sysconfig/network-scripts/ifcfg-eth0` (Red Hat/Fedora/CentOS) for the first NIC, `ifcfg-eth1` for the second, etc, or `/etc/network/interfaces` (Ubuntu) as shown below. Any other commands you may want to add to the system boot sequence can be added to the end of the file `/etc/rc.d/rc.local`. The commands `netcfg` and `netconfig` make permanent changes to system network configuration files located in `/etc/sysconfig/network-scripts/`, so that this information is retained and used upon system boot.

The IANA has allocated IP addresses in the range of 192.168.0.0 to 192.168.255.255 for private networks.

Notes

Helpful tools

Network Calculators: Subnet mask calculator, node calculator, mask inverter, ...

IP subnet calculator

5.2 Connecting to LAN

For this step, work is needed both on the GNU/Linux host and on the iPAQ.

USB Networking on the GNU/Linux Host

You need to have a supported USB hub in your PC, and have the kernel module installed for it. You should also obtain the kernel module named `usbnet`, which is bundled in Linux kernel since version 2.4.10, but working version is only bundled since 2.4.13. However, I found the experimental module in my stock Red Hat 7.2 distribution (kernel 2.4.7), and it seems not to behave too badly...

It is normally found as `/lib/modules/kernel_version/kernel/drivers/usb`.

If you cannot find the module, but have a supported kernel, you need to enable the `CONFIG_USB_USBNET` option in the kernel config (from the main config menu, go into "USB Support", you may need the "Code maturity level option" set to "EXPERIMENTAL"), and compile it with a command line like:

```
bash$ gcc -O2 -I /usr/src/linux/include -DMODULE -D__KERNEL__ -c usbnet.c
```



Note

You must have gcc and correct kernel headers installed in order for this to work.

Otherwise, consider upgrading your kernel. As an alternative, you can continue to download all the required files using the serial method, it will only be slower.

Once the module is compiled, copy `usbnet.o` into the module directories and update the module dependencies with:

```
bash# cp build_dir/usbnet.o /lib/modules/kernel_version/kernel/drivers/usb
```

```
bash# depmod -a
```

You can then insert it with:

```
bash# modprobe usbnet
```

Notes



Caselet

Configuring Internet Connection

I bought a new PC with internal modem. For the time being I do not plan to go in for broadband connection. I would like to instead opt for the free dial-up connection provided by BSNL. Please tell me how I configure this. (I already have a BSNL landline connection.)

JACOBS

Please first make sure that the modem is already installed and shows in the “Device Manager.” (To check this, click Start - Run - devmgmt.msc).

BSNL provides free dialer for Netone - Account less Internet Access. You can download it and install it from [http:// netonechn.bsnl.co.in/NetOne.exe](http://netonechn.bsnl.co.in/NetOne.exe) (805 KB). The next step is to register your telephone number in the BSNL Web site. Dial using following information from netone or any dialer. - Dial No : 172222 - Username: netone - Password: netone Once it is connected, open Internet Explorer and go to [http:// netonechn.bsnl.co.in/registration/](http://netonechn.bsnl.co.in/registration/) and Fill up the Registration Form. Users other than those in Chennai will need to refer to the following URL for the respective cities: [http://www.bsnl.co.in/service/ cities_cli.htm](http://www.bsnl.co.in/service/cities_cli.htm). (The above registration may also be done through any Internet connected machine well in advance by using the above URL.)

Once you have successfully registered, please disconnect the existing connection in the system tray. In netone dialer, henceforth, your username should be your telephone number and the password the one you registered with netone site and the dial number is 172222. For more information, please refer [http:// netonechn.bsnl.co.in/](http://netonechn.bsnl.co.in/) If you have any problem, you can call the BSNL Help desk #1500/1957. Since this supports only up to 56 kpbs, it is advisable to go for broadband, 2 Mbps @ ₹ 250 a month.

Source: <http://www.thehindubusinessline.in/ew/2007/11/05/stories/2007110550120402.htm>

5.3 Summary

Linux can support multiple network devices. The device names are numbered and begin at zero and count upwards. You need to have a supported USB hub in your PC, and have the kernel module installed for it.

5.4 Keyword

File: /etc/hosts: locally resolve node names to IP addresses

5.5 Self Assessment

Fill in the blanks:

1. Linux can support multiple network devices. The device names are numbered and begin at and count upwards.
2. file is used to list hosts to be resolved locally.
3. If using DHCP client, this will automatically be sent to you by the and loaded into this file as part of the DHCP protocol.

- Notes**
4. If using a IP address, ask the ISP or check another machine on your network.
 5. The file format for the hosts file is specified by
 6. /etc/sysconfig/network is the network configuration file used by the system during the boot process.
 7. There is no difference in the configuration developed with the tools and that developed by editing system configuration files directly.
 8. The server requires a static IP so that those who wish to use its resources can find the system.
 9. The commands netcfg and make permanent changes to system network configuration files located in /etc/sysconfig/network-scripts/, so that this information is retained and used upon system boot.
 10. Computers may be assigned a static IP address or assigned one
 11. alias eth0 eepro100 is the example statement for card.
 12. File: /etc/resolv.conf is the name resolver configuration file
 13. file list order of host name search.
 14. A computer running two ethernet cards will have two devices labeled and
 15. Ubuntu/Debian: /etc/network/interfaces file is used to specify and devices. eg. Static IP and info, DHCP, etc.

5.6 Review Questions

1. Analyze the impact of various Linux TCP/IP Network Configuration Files.
2. Make distinction between /etc/hosts file and /etc/nsswitch.conf file.
3. Evaluate the concept of Domain Resolution Configuration Files
4. Analyze the process that configures Linux so that it knows which DNS server will be resolving domain names into IP addresses.
5. File: /etc/hosts locally resolve node names to IP addresses. Enlighten the statement with example.
6. Give example of System Databases and Name Service Switch configuration file and explicate it.
7. Examine the concept of Fedora / Red Hat Network Configuration Files.
8. Scrutinize the GUI tools that are used to edit the system configuration files.
9. Computers may be assigned a static IP address or assigned one dynamically. Comment.
10. Evaluate the methods used for Static IP Address Assignment.

Answers: Self Assessment

- | | |
|-------------|---------------|
| 1. zero | 2. /etc/hosts |
| 3. ISP | 4. Static |
| 5. RFC 952. | 6. Red Hat |

Notes

- | | |
|---------------------------|---------------------------|
| 7. GUI | 8. Linux |
| 9. netconfig | 10. dynamically |
| 11. Intel Ethernet | 12. Host |
| 13. /etc/nsswitch.conf | 14. /dev/eth0, /dev/eth1. |
| 15. network configuration | |

5.7 Further Readings



Books

- Brian Ward, *How Linux Works*, No Starch Press.
- Christopher Negus, *Linux Bible*, Wiley.
- Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.
- Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.
- Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

- http://oreilly.com/catalog/debian/chapter/book/ch11_01.html
- http://www.astahost.com/info.php/Can39t-Connect-Internet-Linux_t14887.html
- http://wiki.answers.com/Q/How_do_you_connect_to_the_Internet_in_Linux

Unit 6: Domain Name System

Notes

CONTENTS

Objectives

Introduction

- 6.1 Domain Name System (DNS)
 - 6.1.1 Configuring DNS Server
 - 6.1.2 DNS Server
 - 6.1.3 Dynamic DNS
 - 6.1.4 Other DNS Server Files
 - 6.1.5 DNS Tools, Testing, and Troubleshooting
 - 6.1.6 A Simple DNS Server
 - 6.1.7 DNS Server Functions
- 6.2 Summary
- 6.3 Keywords
- 6.4 Self Assessment
- 6.5 Review Questions
- 6.6 Further Readings

Objectives

After studying this unit, you will be able to:

- Define domain name system

Introduction

DNS acts as a directory service for all of these systems and let you to specify each one by its hostname. A telephone book allows you to look up an individual person by name and get their telephone number, their unique identifier on the telephone system's network. DNS allows you to look up individual server by name and get its IP address, its unique identifier on the Internet.

6.1 Domain Name System (DNS)

There are other hostname-to-IP directory services in use, mostly for LANs. Windows LANs can use WINS. UNIX LANs can use NIS. But because DNS is the directory service for the Internet (and can also be used for LANs) it is the most widely used. UNIX LANs could always use DNS instead of NIS, and starting with Windows 2000 Server, Windows LANs could use DNS instead of, or in addition to, WINS. And on small LANs where there are only a few machines you could just use HOSTS files on each system instead of setting up a server running DNS, NIS, or WINS.

Notes

6.1.1 Configuring DNS Server

Domain Name System converts the name of a Web site for example (www.Ignou.ac.in) to an IP address (220.227.168.115). This step is significant, because the IP address of a Web site's server.



Caution The Web site's name is not used in routing traffic over the Internet.

DNS Domains

Every person in the world has a first name and a last, or family, name. The same thing is true in the DNS world: A family of Web sites can be loosely described a domain.



Example: The domain Ignou.ac.in has a number of children, such as www.Ignou.ac.in and mail.Ignou.ac.in for the Web and mail servers, respectively.

BIND

BIND is an acronym for the Berkeley Internet Name Domain project, which is a group that sustains the DNS-related software suite that runs under Linux. The most well known program in BIND is named, the daemon that responds to DNS queries from remote machines.

DNS Clients

A DNS client doesn't store DNS information; it has to refer to a DNS server to get it. The only DNS configuration file for a DNS client is the /etc/resolv.conf file, which defines the IP address of the DNS server it should use. You shouldn't need to configure any other files. You'll become well acquainted with the /etc/resolv.conf file soon.



Task A family of Web sites can be loosely described a domain.

Authoritative DNS Servers

Authoritative servers offer the definitive information for your DNS domain, such as the names of servers and Web sites in it. They are the last word in information related to your domain.

How DNS Servers find out your Site Information?

There are 13 root reliable DNS servers (super duper authorities) that all DNS servers query first. These root servers know all the authoritative DNS servers for all the main domains - .com, .net, and the rest. This layer of servers keeps track of all the DNS servers that Web site systems administrators have assigned for their sub domains.



Example: When you register your domain my-site.com, you are in fact inserting a record on the .com DNS servers that point to the authoritative DNS servers you assigned for your domain. (More on how to register your site later.).

When to use a DNS Caching Name Server?

Notes

Most servers don't ask authoritative servers for DNS directly, they generally ask a caching DNS server to do it on their behalf. These servers, through a process called recursion, sequentially query the authoritative servers at the root, main domain and sub domain levels to get eventually get the specific information requested. The most frequently requested information is then stored (or cached) to reduce the lookup overhead of subsequent queries.

If you want to promote your Web site `www.my-site.com` to the rest of the world, then a regular DNS server is what you require.

After you set up your caching DNS server, you have to configure each of your home network PCs to use it as their DNS server. If your home PCs get their IP addresses using DHCP, then you have to configure your DHCP server to make it aware of the IP address of your new DNS server, so that the DHCP server can advertise the DNS server to its PC clients. Off-the-shelf router/firewall appliances used in most home networks usually can act as both the caching DNS and DHCP server, rendering a separate DNS server is unnecessary.



Did u know? Is setting up a caching DNS server is fairly straightforward?

When to use a Static DNS Server?

If your ISP provides you with a fixed or static IP address, and you wish to host your own Web site, then a regular authoritative DNS server would be the way to go. A caching DNS name server is used as a reference only; regular name servers are used as the authoritative source of information for your Web site's domain.

How to get your own Domain?

Whether or not you use static or dynamic DNS, you need to register a domain.

Dynamic DNS providers regularly offer you a sub domain of their own site, such as `my-site.dnsprovider.com`, in which you register your domain on their site.

If you choose to create your very own domain, such as `my-site.com`, you have to register with a company focus in static DNS registration and then point your registration record to the intended authoritative DNS for your domain. Popular domain registrars include VeriSign, Register Free, and Yahoo.

If you want to use a dynamic DNS provider for your own domain, then you have to point your registration record to the DNS servers of your dynamic DNS provider. We will discuss this later.

Basic DNS Testing of DNS Resolution

As you know, DNS resolution maps a Fully Qualified Domain Name (FQDN), such as `www.ignou.ac.in`, to an IP address. This is also known as a forward lookup. The reverse is also true: By performing a reverse lookup, DNS can determine the fully qualified domain name associated with an IP address.

Many dissimilar Web sites can map to a single IP address, but the reverse isn't true; an IP address can map to only one FQDN. This means that forward and reverse entries frequently don't match. The reverse DNS entries are usually the responsibility of the ISP hosting your site, so it is quite common for the reverse lookup to resolve to the ISP's domain. This isn't an important factor for

Notes

most small sites, but some e-commerce applications require matching entries to operate correctly. You may have to ask your ISP to make a custom DNS change to correct this.

There are a number of commands you can use do these lookups. Linux uses the host command, for example, but Windows uses nslookup.

The Host Command

The host command accepts arguments that are either the fully qualified domain name or the IP address of the server when providing results. To carry out a forward lookup, use the syntax:

```
[root@bigboy tmp]# host www.Ignou.ac.in
www.Ignou.ac.in has address 220.227.168.115
[root@bigboy tmp]#
```

To perform a reverse lookup

```
[root@bigboy tmp]# host 220.227.168.115
34.71.115.65. in-addr.arpa domain name pointer 65-115-71-34.myisp.net.
[root@bigboy tmp]#
```

As you can see, the forward and reverse entries don't match. The reverse entry matches the entry of the ISP.

The nslookup Command

The nslookup command provides the same results on Windows PCs. To perform forward lookup, use.

```
C:\> nslookup www.Ignou.ac.in
Server: 192-168-1-200.my-site.com
Address: 192.168.1.200
Non-authoritative answer:
Name: www.Ignou.ac.in
Address: 220.227.168.115
C:\>
```

To Perform a reverse lookup

```
C:\> nslookup 220.227.168.115
Server: 192-168-1-200.my-site.com
Address: 192.168.1.200
Name: 65-115-71-34.my-isp.com
Address: 220.227.168.115
C:\>
```

Downloading and Installing the BIND Packages

Notes

Most RedHat and Fedora Linux software products are obtainable in a package format. When searching for the file, remember that the BIND package's filename usually starts with the word "bind" followed by a version number, as in bind-9.2.2.P3-9.i386.rpm.



Note Unless otherwise stated, the sample configurations covered in this unit will be for Redhat/Fedora distributions.

How to Get BIND Started

Setting up your DNS server is easy to do, but the procedure differs between Linux distributions.

Redhat / Fedora

You can use the `chkconfig` command to get BIND configured to start at boot

```
[root@bigboy tmp]# chkconfig named on
```

To start, stop, and restart BIND after booting, use:

```
[root@bigboy tmp]# /etc/init.d/named start
```

```
[root@bigboy tmp]# /etc/init.d/named stop
```

```
[root@bigboy tmp]# /etc/init.d/named restart
```

Remember to restart the BIND process every time you make a change to the configuration file for the changes to take effect on the running process.

Debian / Ubuntu

You can use the `sysv-rc-conf` command to get BIND configured to start at boot

```
[root@bigboy tmp]# sysv-rc-conf bind on
```

To start, stop, and restart BIND after booting, use

```
[root@bigboy tmp]# /etc/init.d/bind start
```

```
[root@bigboy tmp]# /etc/init.d/bind stop
```

```
[root@bigboy tmp]# /etc/init.d/bind restart
```

However the startup script and installation package name refers to `bind`, the name of the daemon that runs is `named` just like it is with Redhat / Fedora. Also remember to restart the BIND process every time you make a change to the configuration file for the changes to take effect on the running process.

The `/etc/resolv.conf` File

DNS clients (servers not running BIND) use the `/etc/resolv.conf` file to conclude both the location of their DNS server and the domains to which they belong. The file generally has two columns; the first contains a keyword, and the second contains the desired values separated by commas.

Notes

Table 6.1: Keywords In /etc/resolv.conf

Keyword	Value
Nameserver	IP address of your DNS nameserver. There should be only one entry per "nameserver" keyword. If there is more than one nameserver, you'll need to have multiple "nameserver" lines.
Domain	The local domain name to be used by default. If the server is bigboy.my-web-site.org, then the entry would just be my-web-site.org
Search	If you refer to another server just by its name without the domain added on, DNS on your client will append the server name to each domain in this list and do a DNS lookup on each to get the remote servers' IP address. This is a handy time saving feature to have so that you can refer to servers in the same domain by only their servername without having to specify the domain. The domains in this list must separated by spaces.

Obtain a sample configuration in which the client server's main domain is my-site.com, but it also is a member of domains my-site.net and my-site.org, which should be searched for shorthand references to other servers. Two name servers, 192.168.1.100 and 192.168.1.102, provide DNS name resolution:

```
search my-site.com my-site.net my-site.org
nameserver 192.168.1.100
nameserver 192.168.1.102
```

The first domain scheduled after the search directive must be the home domain of your network, in our case ignou.ac.in. Placing a domain and search entry in the /etc/resolv.conf is redundant, therefore.

Important File Locations

The locations of the BIND configuration files vary by Linux distribution, as you will soon see.

RedHat / Fedora

RedHat / Fedora BIND normally runs as the named process owned by the unprivileged named user.

Sometimes BIND is also installed using Linux's chroot characteristic to not only run named as user named, but also to limit the files named can see. When installed, named is fooled into thinking that the directory /var/named/chroot is actually the root or / directory. Therefore, named files normally found in the /etc directory are found in /var/named/chroot/etc directory instead, and those you'd expect to find in /var/named are actually located in /var/named/chroot/var/named.

The benefit of the chroot feature is that if a hacker enters your system via a BIND exploit, the hacker's access to the rest of your system is isolated to the files under the chroot directory and nothing else. This type of security is also known as a chroot jail.

You can determine whether you have the chroot add-on RPM by using this command, which returns the name of the RPM.

```
[root@bigboy tmp]# rpm -q bind-chroot
bind-chroot-9.2.3-13
[root@bigboy tmp]#
```

There can be uncertainty with the locations: Regular BIND installs its files in the normal locations, and the chroot BIND add-on RPM installs its own versions in their chroot locations. Unfortunately, the chroot versions of some of the files are empty. Before starting Fedora BIND, copy the configuration files to their chroot locations:

```
[root@bigboy tmp]# cp -f /etc/named.conf /var/named/chroot/etc/
[root@bigboy tmp]# cp -f /etc/rndc.* /var/named/chroot/etc/
```

Before you go to the next step of configuring a regular name server, it is important to understand exactly where the files are located.



Task

Describe why the first domain scheduled after the search directive must be the home domain of your network?

Table 6.2: Differences in Fedora and Redhat DNS File Locations

File	Purpose	BIND chroot Location	Regular BIND Location
named.conf	Tells the names of the zone files to be used for each of your website domains.	/var/named/chroot/etc	/etc
rndc.key rndc.conf	Files used in named authentication	/var/named/chroot/etc	/etc
zone files	Links all the IP addresses in your domain to their corresponding server	/var/named/chroot/ var/named	/var/named

Fedora Core installs BIND chroot by default. RedHat 9 and earlier don't.

Debian / Ubuntu

With Debian / Ubuntu, all the configuration files, the primary `named.conf` file and all the DNS zone files reside in the `/etc/bind` directory.

Unlike in Redhat / Fedora, references to other files within these configuration files must include the full path. The named daemon won't automatically assume they are located in the `/etc/bind` directory.

Configuring your Nameserver

Assume your ISP assigned you the subnet 97.158.253.24 with a subnet mask of 255.255.255.248 (/29).

Configuring `resolv.conf`

You'll have to build your DNS server refer to itself for all DNS queries by configuring the `/etc/resolv.conf` file to reference localhost only.

```
nameserver 127.0.0.1
```

Creating a `named.conf` Base Configuration


The `/etc/named.conf` file contains the main DNS configuration and tells BIND where to find the configuration, or zone files for each domain you own. This file generally has two zone areas:

1. Forward zone file definitions list files to map domains to IP addresses.
2. Reverse zone file definitions list files to map IP addresses to domains.

Notes

Some versions of BIND will come with a /etc/amed.conf file configured to work as a caching nameserver which can be transformed to an authoritative nameserver by adding the correct references to your zone files. Please proceed to the next section if this is the case with your version of BIND.

In additional cases the named.conf configuration file may be hard to find. Some versions of Linux install BIND as a default caching nameserver using a file names /etc/named.caching-nameserver.conf for its configuration. In such cases BIND becomes an authoritative nameserver when a correctly configured /etc/named.conf file is created.



Note BIND comes with samples of all the primary files you need. Below Table explains their names and purpose in more detail.

The Primary BIND Configuration Files

File	Description
/etc/named.conf	The main configuration file that lists the location of all your domain's zone files
/etc/named.rfc1912.zones	Base configuration file for a caching name server.
/var/named/named.ca	A list of the 13 root authoritative DNS servers.

The first task is to make sure your DNS server will listening of requests on all the required network interfaces. The options section of named.conf may be configured to listen completely on its internal hidden localhost interface with an IP address of 127.0.0.1 as we see in this example.

```
# File: /etc/named.conf
Options {
    Listen-on port 53 {127.0.0.1; };
};
```

If other devices are going to rely on your server for queries, then you'll need to either modify this or add a selected number of IP addresses on your server. In this example, we allow queries on any interface.

```
Listen-on port 53 {any ;};
```

In this example, we allow queries on localhost and address 192.168.1.100.

```
listen-on port 53 { 127.0.0.1; 192.168.1.100;};
```

Always make sure localhost, 127.0.0.1 is included.

While it is not required, it is a good practice to configure your DNS server's named.conf file to support BIND views. This will be discussed next.

Configuring BIND Views in named.conf

Our sample scenario believes that DNS queries will be coming from the Internet and that the zone files will return information related to the external 97.158.253.26 address of the Web server. What do the PCs on your home network need to see? They need to see DNS references to the real IP address of the Web server, 192.168.1.100, because NAT won't work properly if a PC on your home network attempts to connect to the external 97.158.253.26 NAT IP address of your

Web server. Don't worry. BIND figures this out using its views feature which allows you to use predefined zone files for queries from certain subnets. This means it's possible to use one set of zone files for queries from the Internet and another set for queries from your home network. Here's a summary of how it's done:

1. If your DNS server is also performing as a caching DNS server, then you'll also need a view for localhost to use. We'll use a view called `localhost_resolver` for this.
2. Place your zone statements in the `/etc/named.conf` file in one of two other view sections. The first section is known as `internal` and lists the zone files to be used by your internal network. The second view called `external` lists the zone files to be used for Internet users.



Example: You could have a reference to a zone file called `my-site.zone` for lookups allied to the `97.158.253.X` network which Internet users would see. This `/etc/named.conf` entry would be inserted in the `external` section. You could also have a file called `my-site-home.zone` for lookups by home users on the `192.168.1.0` network. This entry would be inserted in the `internal` section. Creating the `my-site-home.zone` file is fairly easy: Copy it from the `my-site.zone` file and replace all references to `97.158.253.X` with references to `192.168.1.X`.

3. You must also tell the DNS server which addresses you feel are internal and external. To do this, you must first describe the internal and external networks with access control lists (ACLs) and then refer to these lists within their respective view section with the `match-clients` statement. Some built-in ACLs can save you time:
 - (a) `localhost`: Refers to the DNS server itself
 - (b) `localnets`: Refers to all the networks to which the DNS server is directly connected
 - (c) `any`: which is self explanatory.

Let's observe BIND views more carefully using a number of sample configuration snippets from the `/etc/named.conf` file I use for my home network. All the statements below were inserted after the options and controls sections in the file. I have selected generic names `internal`, for views given to trusted hosts (home, non-internet or corporate users), and `external` for the views given to Internet clients, but they can be named whatever you wish.



Task

Describe the queries in your home networks

First let's talk about how we should refer to the zone files in each view.

Forward Zone File References in `named.conf`

Let's describe how we point to forward zone files in a typical `named.conf` file.

In this example the zone file is named `my-site.zone`, and, though not explicitly stated, the file `my-site.zone` should be located in the default directory of `/var/named/chroot/var/named` in a chroot configuration or in `/var/named` in a regular one. With Debian / Ubuntu, references to the full file path will have to be used. Use the code:

```
Zone "my-web-site.org" {
    type master;
    notify no;
    allow-query { any; };
    file "my-site.zone";
};
```

Notes

In addition, you can insert more entries in the named.conf file to reference other Web domains you host. Here is an case for another-site.com using a zone file named another-site.zone.

```
zone "another-site.com" {  
  
    type master;  
  
    notify no;  
  
    allow-query { any; };  
  
    file "another-site.zone";  
  
};
```

6.1.2 DNS Server

As a service, DNS is crucial to the operation of the Internet. When you enter www.some-domain.com in a Web browser, it's DNS that takes the www host name and translates it to an IP address. Without DNS, you could be connected to the Internet just fine, but you ain't goin' no where. Not unless you keep a record of the IP addresses of all of the resources you access on the Internet and use those instead of host/domain names.

So when you visit a Web site, you are in fact doing so using the site's IP address even though you specified a host and domain name in the URL. In the background your computer quickly queried a DNS server to get the IP address that corresponds to the Web site's server and domain names. Now you know why you have to specify one or two DNS server IP addresses in the TCP/IP configuration on your desktop PC (in the resolv.conf file on a Linux system and the TCP/IP properties in the Network Control Panel on Windows systems).

A "cannot connect" error doesn't essentially indicate there isn't a connection to the destination server. There may very well be. The error may indicate a failure in "resolving" the domain name to an IP address. I use the open source Firefox Web browser on Windows systems because the status bar gives more informational messages like "Resolving host", "Connecting to", and "Transferring data" rather than just the generic "Opening page" with IE. (It also seems to render pages faster than IE.)

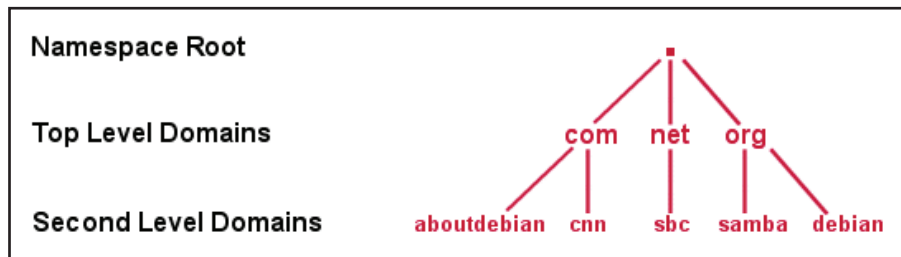
In short, always check for right DNS operation when troubleshooting a problem involving the inability to access an Internet resource. The ability to resolve names is critical, and later in this page we'll show you some tools you can use to investigate and verify this capability.

When you are surfing the Web viewing Web pages or sending an e-mail your workstation is sending queries to a DNS server to determine server/domain names. (Back on the Modems page we showed you how to set up your resolv.conf file to do this.) When you have you own Web site that other people visit you need a DNS server to respond to the queries from their workstations.

When you visit Web sites, the DNS server your workstation question for name resolution is typically run by your ISP, but you could have one of your own. When you have your own Web site the DNS servers which respond to visitors queries are typically run by your Web hosting provider, but you could likewise have your own one of these too. Actually, if you set up your own DNS server it could be used to respond to both "internal" (from your workstation) and "external" (from your Web site's visitors) queries.

Even if you don't have your own domain name, or even your own LAN, you can still have advantage from using a DNS server to allow others to access your Debian system. If you have a single system connected to the Interent via a cable or DSL connection, you can have it act as a Web/e-mail/FTP server using a neat service called "dynamic DNS" which we'll cover later. Dynamic DNS will even work with a modem if you want to play around with it.

Finding a single server out of all of the servers on the Internet is like annoying to find a single file on drive with thousands of files. In both cases it helps to have some hierarchy built into the directory to logically group things. The DNS “namespace” is hierarchical in the same type of upside-down tree structure seen with file systems. Just as you have the root of a partition or drive, the DNS namespace has a root which is signified by a period.



When specifying the absolute path to a file in a file system you start at the root and go to the file:

```
/etc/bind/named.conf
```

When specifying the absolute path to a server in the DNS namespace you start at the server and go to the root:

```
www.aboutdebian.com.
```

Period after the ‘com’ as it’s significant. It’s how you specify the root of the namespace. An absolute path in the DNS namespace is called a FQDN (Fully Qualified Domain Name). The use of FQDNs are prevalent in DNS configuration files and it’s important that you always use that trailing period.

Internet resources are typically specified by a domain name and a server hostname. The www part of a URL is often the hostname of the Web server (or it could be an alias to a server with a different host name). DNS is basically just a database with records for these hostnames. The directory for the entire telephone system is not stored in one huge phone book. Rather, it is broken up into many pieces with each city having, and maintaining, its piece of the entire directory in its phone book. By the same token, pieces of the DNS directory database (the “zones”) are stored, and maintained, on many different DNS servers located around the Internet. If you want to find the telephone number for a person in Poughkeepsie, you’d have to look in the Poughkeepsie telephone book. If you want to find the IP address of the www server in the some-domain.com domain, you’d have to query the DNS server that stores the DNS records for that domain.

The entries in the database map a host/domain name to an IP address. Here is a naive logical view of the type of information that is stored (we’ll get to the A, CNAME, and MX designations in a bit).

A	www.their-domain.com	172.29.183.103
MX	mail.their-domain.com	172.29.183.217
A	debian.your-domain.com	10.177.8.3
CNAME	www.your-domain.com	10.177.8.3
MX	debian.your-domain.com	10.177.8.3

This is why a real Internet server needs a static (unchanging) IP address. The IP address of the server’s NIC connected to the Internet has to match whatsoever address is in the DNS database. Dynamic DNS does provide a way around this for home servers however, which we’ll see later.

Notes

When you want to browse to `www.their-domain.com` your DNS server (the one you state in the TCP/IP configuration on your desktop computer) most likely won't have a DNS record for the `their-domain.com` domain so it has to contact the DNS server that does. When your DNS server contacts the DNS server that has the DNS records (referred to as "resource records" or "zone records") for `their-domain.com` your DNS server gets the IP address of the `www` server and relays that address back to your desktop computer. So which DNS server has the DNS records for a particular domain?

When you register a domain name with somebody like Network Solutions, one of the things they ask you for are the server names and addresses of two or three "name servers" (DNS servers). These are the servers where the DNS records for your domain will be stored (and queried by the DNS servers of those browsing to your site). Typically, when you host your Web site using a Web hosting service they not only provide a Web server for your domain's Web site files but they will also provide a DNS server to store your domain's DNS records. In other words, you'll want to know who your Web hosting provider is going to be before you register a domain name (so you can enter the provider's DNS server information in the name servers section of the domain name registration application). Once you've got your Web site up and running on your Web hosting provider's servers and someone surf's to your site, the DNS server they specified in their local TCP/IP configuration will query your hosting provider's DNS servers to get the IP address for your Web site. The DNS servers that host the DNS records for your domain, i.e. the DNS servers you specify in your domain name registration request, are the authoritative DNS servers for your domain. The surfer's DNS server queries one of your site's authoritative DNS servers to get an address and gets an authoritative response. When the surfer's DNS server relays the address information back to the surfer's local PC it is a "non-authoritative" response because the surfer's DNS server is not an authoritative DNS server for your domain.



Example: If you surf to MIT's Web site the DNS server you have specified in your TCP/IP configuration queries one of MIT's reliable DNS servers and gets an authoritative response with the IP address for the 'www' server. Your DNS server then sends a non-authoritative response back to your PC. You can easily see this for yourself. At a shell prompt, or a DOS window on a newer Windows system, type in:

```
nslookup www.mit.edu
```

First you'll observe the name and IP address of your locally-specified DNS server. Then you'll see the non-authoritative response your DNS server sent back containing the name and IP address of the MIT Web server.

If you're on a Linux system you can also notice which name server(s) your DNS server contacted to get the IP address. At a shell prompt type in:

```
whois mit.edu
```

and you'll see three authoritative name servers listed with the hostnames STRAWB, W20NS, and BITSY. The 'whois' command only returns the contents of a site's domain record.



Did u know? So where do you get the "name servers" information for your domain?

Don't puzzle DNS zone records with domain records. Your domain record is created when you fill out a domain name registration application and is maintained by the domain registration service (like Network Solutions) you used to register the domain name. A domain only has one domain record and it contains administrative and technical contact information as well as entries for the authoritative DNS servers (aka "name servers") that are hosting the DNS records for the domain. You have to enter the hostnames and addresses for multiple DNS servers in your domain record for redundancy (fail-over) purposes.

DNS records (aka zone records) for a domain are stored in the domain's zone file on the authoritative DNS servers. Naturally, it is stored on the DNS servers of whatever Web hosting service is hosting your domain's Web site. However, if you have your own Web server (rather than using a Web hosting service) the DNS records could be hosted by you using your own authoritative DNS servers (as in MIT's case), or by a third party like EasyDNS.

In short, the name servers you specified in your domain record host the domain's zone file consisting the zone records. The name servers, whether they be your Web hosting provider's, those of a third party like EasyDNS, or your own, which host the domain's zone file are authoritative DNS servers for the domain.

Because DNS is so imperative to the operation of the Internet, when you register a domain name you must specify a minimum of two name servers. If you set up your own authoritative DNS servers for your domain you must set up a minimum of two of them (for redundancy) and these would be the servers you specify in your domain record. While the multiple servers you specify in your domain record are authoritative for your domain, only one DNS server can be the primary DNS server for a domain. Any others are "secondary" servers. The zone file on the primary DNS server is "replicated" (transferred) to all secondary servers. As a result, any changes made to DNS records must be made on the primary DNS server. The zone files on secondary servers are read-only. If you made changes to the records in a zone file on a secondary DNS server they would simply be overwritten at the next replication. As you will see below, the primary server for a domain and the replication frequency are specified in a special type of zone record.

Early on in this page we said that the DNS zone records are stored in a DNS database which we now know is called a zone file. The term "database" is used quite loosely. The zone file is actually just a text file which you can edit with any text editor. A zone file is domain-specific. That is, each domain has its own zone file. Actually, there are two zone files for each domain but we're only concerned with one right now. The DNS servers for a Web hosting provider will have many zone files, two for each domain it's hosting zone records for. A zone "record" is, in most cases, nothing more than a single line in the text zone file.

There are different types of DNS zone records. These several record types give you flexibility in setting up the servers in your domain. The most common types of zone records are:

1. An A (Address) record is a "host record" and it is the most ordinary type. It is simply a static mapping of a hostname to an IP address. A common hostname for a Web server is 'www' so the A record for this server gives the IP address for this server in the domain.
2. An MX (Mail eXchanger) record is specially for mail servers. It's a special type of service-specifier record. It identifies a mail server for the domain. That's why you don't have to enter a hostname like 'www' in an e-mail address. If you're running Sendmail (mail server) and Apache (Web server) on the same system (i.e. the same system is acting as both your Web server and e-mail server), both the A record for the system and the MX record would refer to the same server. To offer some fail-over protection for e-mail, MX records also have a Priority field (numeric). You can enter two or three MX records each pointing to a different mail server, but the server specified in the record with the highest priority (lowest number) will be chosen first. A mail server with a priority of 10 in the MX record will receive e-mail before a server with a priority of 20 in its MX record. Note that we are only talking about receiving mail from other Internet mail servers here. When a mail server is sending mail, it acts like a desktop PC when it comes to DNS. The mail server looks at the domain name in the recipient's e-mail address and the mail server then contacts its local DNS server (specified in the resolv.conf file) to get the IP address for the mail server in the recipient's domain. When an authoritative DNS server for the recipient's domain receives the query from the sender's DNS server it sends back the IP addresses from the MX records it has in that domain's zone file.

Notes

3. A CNAME (Canonical Name) record is an alias record. It's a way to have the same physical server react to two different hostnames. Let's say you're not only running Sendmail and Apache on your server, but you're also running WU-FTP so it also acts as an FTP server. You could create a CNAME record with the alias name 'ftp' so people would use ftp.your-domain.com and www.your-domain.com to access different services on the same server.
4. Another use for a CNAME record was demonstrated in the example near the top of the page. Suppose you name your Web server 'debian' instead of 'www'. You could simply create a CNAME record with the alias name 'www' but with the hostname 'debian' and debian's IP address.
5. NS (Name Server) records specify the authoritative DNS servers for a domain.
6. There can be multiples of all of the above record types. There is one unusual record type of which there is only one record in the zone file. That's the SOA (Start Of Authority) record and it's the first record in the zone file. An SOA record is only present in a zone file located on authoritative DNS servers (non-authoritative DNS servers can cache zone records). It specifies such things as:
 - (a) The primary authoritative DNS server for the zone (domain).
 - (b) The e-mail address of the zone's (domain's) administrator. In zone files, the '@' has a specific meaning so the e-mail address is written as me.my-domain.com.
 - (c) Timing information as to when secondary DNS servers should refresh or expire a zone file and a serial number to specify the version of the zone file for the sake of comparison.

The SOA record is the one that takes up several lines.


Several important points to note about the records in a zone file:

Records can specify servers in other domains. This is most frequently used with MX and NS records when backup servers are located in a different domain but receive mail or resolve queries for your domain.

There must be an A record for systems specified in all MX, NS, and CNAME records.

A and CNAME records can specify workstations as well as servers (which you'll see when we set up a LAN DNS server).

Now let's look at a typical zone file. When a Debian system is set up as a DNS server the zone files are stored in the /etc/bind directory.



Note In a zone file the two parentheses around the timer values act as line-continuation characters as does the '\ ' character at the end of second line. The ';' is the comment character. The 'IN' indicates an INternet-class record.

6.1.3 Dynamic DNS

If you set up a Debian system to act as a combination firewall, NAT, and home Web server you (and others if you wish) can contact the Web pages on it (such as your Web cam images) from a remote location by entering the system's IP address in the URL. The IP address would be whatever is assigned to you by your ISP. The problem is that, unless you pay extra to have a static IP address, the IP address assigned by your ISP will change from time to time and trying keeping up with these changes can be a pain. You can get around this by using a host and domain name to

access your system instead of an IP address. Being able to access your system using a consistent name in the URL even though the IP address changes is a major benefit of dynamic DNS.

Dynamic DNS (DDNS) is the capability for a host (your Debian server) to update its own DNS A record. A host's IP address (or what appears to be its IP address) can change when you use a home broadband service such as cable or DSL, or when you dial into an ISP (PPP connection) using a modem. If you have a broadband connection, DDNS allows you to have a full-time Internet server even though you don't have a static IP address.

You run a small DDNS client on your server that launch DNS record update requests to the DDNS server. If you have your own domain name, the DDNS server is the one that's listed as the primary name server in your domain record. Most DNS servers do not support dynamic updates by default. They have to be configured to listen for dynamic updates. When your server is booted up (or you run the client software manually) it sends a request to the DDNS server to check/update the IP address in the A record for your server. If you've pulled a different IP address from your ISP since the last time a request was sent, the A record is updated with this new IP address.

When you use a firewall router, what appears to be your server's IP address is in fact the IP address on the "external" router interface. As mentioned on the Networking page, the router does NAT and this address translation can cause difficulties for dynamic DNS. ddclient is a DDNS client that works with firewalls, is compatible with a number of DDNS services, and is available as a Debian package.

Dynamic DNS with your own Domain

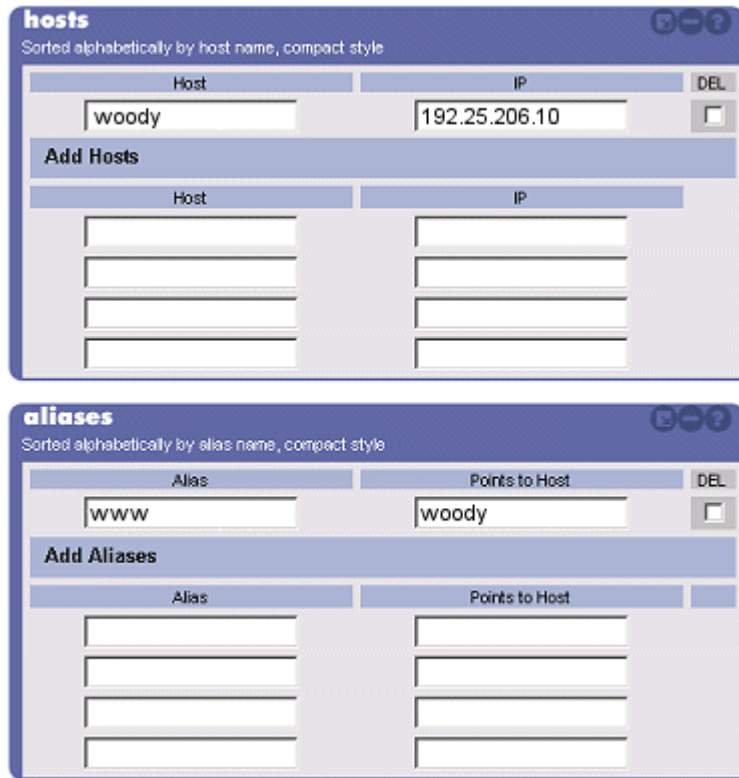
You can use dynamic DNS if you previously have, or want to have, your own registered domain name. You may want your own domain name for several reasons:

1. You would like to set up production Internet servers for an organization or business with static IP addresses.
2. You would like to use your own domain name with your home server(s).
3. You would like to (as in my case) set up a "non-production" domain just for playing around with. A non-production domain would allow you to examine how DNS works by playing around with the zone record values. Being that there are no production servers in the domain, there's no problem if you screw something up. (A non-production domain is a real domain with whatever name you choose but you just use it with test servers. Naturally, you can make it a production domain at any time just by setting up "real" servers.)

For this I use the Domain Name+DNS Only Service bundle from EasyDNS.com since it kills two birds with one stone (and because they have toll-free telephone tech support). EasyDNS will not only host your zone files on their DNS server, but register your domain name (and annually renew the domain name registration) all for \$35/year. That's a pretty good deal as well as being convenient. You don't have to go to one place to register/renew your domain name, and then go to another place to host your DNS records. When you register a domain name with EasyDNS they'll set up some preliminary zone records for you and you just go in and add/modify/delete records.

EasyDNS offer a Web interface for DNS management so you can play around with the settings, change server names, create alias records, etc.

Notes



The finest part is support for dynamic DNS is included in their DNS offerings so you can use them for home and test servers that don't have static IP addresses (and ddclient will work with them too).



As a side note, what if you previously have a domain name with servers that have static IP addresses? Most places that will host your DNS records (like your ISP or Web hosting provider) won't let you even see them much less work with them. Having EasyDNS host your DNS records will allow you to have some control in the management of your DNS. Their straight DNS records hosting service (without the domain registration/renewal piece) costs \$20/year. Just be sure to update your domain record with the EasyDNS name servers information once you sign up for the service and get your DNS records set up. (You also have the option of transferring your domain to them, i.e. making them the domain name registrar, if you want to take advantage of the single-payment convenience thing.)

Having your own non-production domain will not only allow you to play around with zone records, but you can experiment with having your own authoritative DNS server. One of EasyDNS's servers would be the primary authoritative name server and you could set up your Debian server as a secondary. Then you'd just use the 'nameservers' link in the Web interface to enter your server's hostname and IP address as a secondary server entry. This way you could play with the "zone transfers" that take place between authoritative servers.

Notes

But the profit of having your own non-production domain goes beyond just DNS. It also comes in handy for testing Sendmail e-mail server and Apache Web server configurations, etc. For instance, you can see if your Debian system properly sends and receives e-mail for your non-production domain. Or you could install a test certificate (available for free from most certifying authorities like Thawte or Verisign) on your Debian system acting as a Web server so you can investigate SSL functionality. Just about any type of Internet server you want to play with will have more functionality when you can give it a registered domain name that has DNS resolution capabilities. And if you don't have any plans to eventually use it as a production domain, just let it expire after the first year is up and the knowledge gained will be well worth the 35 bucks. Support for dynamic DNS is disabled by default which is fine if you do have static IP address(es) on your server(s).

mx settings
Sorted alphabetically by zone, compact style

Zone	Handled by host	Pref.	DEL
my-last-name.net	woody.my-last-name.net.	5	<input type="checkbox"/>

Add MX records

Zone	Handled by host	Pref.
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

To use easyDNS as a backup mail spool, add the server **smtp.easydns.com** with the lowest priority setting (i.e. the higher preference number). You can then also add with an even higher preference to get double back-up spools

ddclient Configuration File for EasyDNS

We'll install ddclient in a bit. It'll prompt you for the essential configuration information during the install. When it's finished it'll create the `/etc/ddclient.conf` file and it should look something like this (the information you enter during the client install is in blue):

```
# Configuration file for ddclient generated by debconf
#
# /etc/ddclient.conf

pid=/var/run/ddclient.pid
protocol=easydns
use=if, if=eth0
server=members.easydns.com
login=bgates
password=luvlinux
my-last-name.net
```

Notes

If your server is behind a cable/DSL router (such as a Linksys, DLink, or Netgear) or some other type of firewall or proxy server, replace the line:

```
use=if, if=eth0
```

with the line:

```
use=web, web=support.easydns.com/utils/get_ip.php
```

This basically uses a page on EasyDNS's Web site to display your 'outside' IP address. The ddclient software will read the IP address off the returned HTML code and send it to EasyDNS. It'll do this periodically which is necessary with the changing IP addresses you get with cable and DSL services.

Allowing dynamic DNS using the EasyDNS Web interface is simply a matter of clicking on the "disabled" link as illustrated below and acknowledging the change on the subsequent confirmation page.



Once you've got your configuration file set up and you've set your domain for dynamic DNS, you can test your ddclient configuration to certify it's working with the command:

```
ddclient -daemon=0 -noquiet -debug -verbose
```

If you use Apache's virtual hosts feature to host multiple Web sites on your server and you have multiple domain names registered with EasyDNS you can update the dynamic DNS for all the domains simultaneously by separating each of the domains with a comma (,) like so:

```
# Configuration file for ddclient generated by debconf
#
# /etc/ddclient.conf
pid=/var/run/ddclient.pid
protocol=easydns
use=if, if=eth0
server=members.easydns.com
login=bgates
password=luvlinux
my-last-name.net,moe.com,larry.com,curly.com
```

The EasyDNS Web interface let you to add/modify A records, MX records and priorities, aliases (CNAME records), and even the time intervals in the SOA record, all by clicking on the "dns" link shown in the figure above. The "nameservers" link takes you to a page which lists the authoritative DNS server information (EasyDNS's name servers) for your domain.



Did u know? **How is the Support for dynamic DNS is disabled?**

Your own DNS Server

Notes

If you have your own domain name and you also wish to try running your own DNS server, EasyDNS.com has a Secondary DNS Service for \$15/year which takes some of the risk out of running your own DNS. You set their servers up to transfer zone information from your DNS server. You would then enter your DNS server address as the primary in your domain record, and the EasyDNS DNS server addresses as the secondary DNS servers in your domain record. Then, should your DNS server ever fail, name resolution queries will go to the EasyDNS servers.

Your own Domain without your own Servers

Would you like your own domain name and get e-mail and Web traffic to your domain without all the work of setting up your own e-mail and Web servers? Piece of cake!

Now get the Domain Name+DNS Plus Service pack from EasyDNS.com for \$55/year. This service includes the domain name registration (and renewals) and you'll be able to:

1. Have your domain name annually renewed automatically when you restore the service.
2. Have e-mails sent to a you@yourdomain.com e-mail address automatically forwarded to anything existing e-mail address you want (such as your e-mail address provided by your ISP) without having to pay for additional e-mail services.
3. Use their EasySMTP service (included with the Plus bundle) to send e-mail messages by your you@yourdomain.com e-mail address.
4. Have Web requests to http://www.yourdomain.com automatically forwarded to any URL you want (like your personal Web space you get from your ISP).

Your own domain name is nice to have for some reasons. You may want to use your last name for your domain name (if it's available). Some of the benefits of having your own domain name include:

1. You can set up lastname.com e-mail addresses for every member of your family (called "mailmaps") and basically forward their e-mail to their existing e-mail account even if everyone uses a different ISP or Web-mail service.
2. With your own domain name, your e-mail address and Web URL will always remain similar no matter how often you switch ISPs or Web-mail services.

Since so many sites use your e-mail address as a login ID it's a real pain to change your e-mail address. Not to mention notifying everyone that your e-mail address is different. With e-mail forwarding using your own domain name, if you switch ISPs you simply change the forwarding address. The ability to have a consistent e-mail address is valuable for students as their e-mail providers change while they go from high school to college to their first job (and having a lastname.com e-mail address looks good on a resume too). It allows you to have the same e-mail address if you're forced to change ISPs because you move to a different city. It also protects you from ISP mergers, failures, and name changes. And having a consistent Web URL via Web forwarding means you won't lose all the search engine traffic to your Web pages if the URL to your personal Web space changes.

One important thing about the EasyDNS.com service is that it supports SPF TXT records. This allows you to set up Sender Policy Framework for your mail server which prevents spammers from sending e-mails using your domain name. SPF is fast becoming the standard for spam prevention. You use it to specify the IP addresses of servers which are allowed to send e-mails on behalf of your domain (usually only one or two). When a spammer sends an e-mail using your domain name, and the receiving e-mail (SMTP) server is configured to use SPF, the receiving e-mail server checks the source address of the e-mail against the allowed addresses listed in your SPF TXT record in your DNS. If there's no match the mail is discarded.

Notes



Task

Analyse the use of domain without servers.

Domain-less DNS

If you have a broadband Internet connection with no static IP and have no desire to have your own domain name, you can use a free service offered by dyndns.org to set up a home Web/e-mail/ftp server. It offers a dynamic DNS service which will redirect traffic to your server using their domain name.

With this free service you use your server's hostname but dyndns.org's domain name. You're mainly just adding/modifying an A record for your server in their zone file. Your Web server would have a URL like:

```
http://your-hostname.dyndns.org
```

E-mail addressed to your server would have to have an address like:

```
you@your-hostname.dyndns.org
```

As you'll be using your hostname with dyndns.org's domain name, you have to confirm your hostname isn't the same as that of anyone else using their service. As a result, you'll want to come up with a hostname for your server that's really unique. Recall that you set the hostname during the installation. You can always change it by editing the `/etc/hosts` file. However, you'll also need to check for the current hostname in the configuration files of any server applications that may use it, such as Sendmail and Apache, and edit those files as well.

If you connect your Linux server to the Internet using a modem (we show you how on the Modems page), you'll want a way to keep your connection up long enough for any dynamic DNS changes to take effect and this could take up to 45 minutes. Most ISPs will drop an inactive connection before that. You can use the ping command to keep your PPP connection up. The trick is to run it in the background and set it so it only sends a ping once every five minutes. Pick a Web site and enter:

```
ping -i 300 www.chosen-site.com > /dev/null &
```

Just don't forget to bring it to the foreground and stop it once you've disconnected your modem connection. To bring it to the foreground simply type:

```
fg ping
```

and then press Ctrl-C to exit the ping program.

ddclient Configuration File for dyndns.org

If you selected the dyndns.org service when you installed ddclient your `/etc/ddclient.conf` file should look something like this:

```
# Configuration file for ddclient generated by debconf
#
# /etc/ddclient.conf
pid=/var/run/ddclient.pid
protocol=dyndns2
use=if, if=ppp0
server=members.dyndns.org
```

```
login=bgates
```

```
password=luvlinux
```

```
your-debian-box-hostname.dyndns.org
```

Note that this file point to the ppp0 (dialup modem) interface was entered during the installation rather than the 'eth0' that you would use for a network card.

If your server is behind a cable/DSL router (such as a Linksys, DLink, or Netgear) or a few other type of firewall or proxy server, replace the line:

```
use=if, if=ppp0
```

with the line:

```
use=web, web=checkip.dyndns.com/, web-skip=' Current IP Address:'
```



Note This basically uses a page on dyndns.org's Web site to display your 'outside' IP address. The ddclient software will read the IP address off the returned HTML code.

Installing ddclient

Before installing this package is sure to sign up for an account with EasyDNS or dyndns.org. You'll require your account username and password when you install the package. With your account set up you install the package by typing in apt-get install ddclient at the shell prompt. You'll then be prompted for the following:

1. Select the service you want to use.
2. The next screen may appear confusing if you selected EasyDNS in Step 1 because it prompts you for "your DynDNS fully qualified domain names" and then gives examples for dyndns.org. What they mean by the "DynDNS" is "Dynamic DNS", not "DynDNS.org". The "fully qualified" is also a bit misleading. You don't need to enter a trailing period after the TLD (.com, .net, or .org Top Level Domain). All you need to do is enter your server's hostname followed by your, or dyndns.org's, domain name. Examples:

```
debian.gates.com
```

or

```
very-unique-hostname.dyndns.org
```
3. Enter the username you choose when you signed up with your service.
4. Enter the password you choose when you signed up with your service.
5. Enter the interface that will be connecting to the service. This will most likely be 'eth0' for an ethernet card (even if it is connected to a LAN which has a firewall router) or 'ppp0' for modem use (note that that's a zero on the end, not the letter O).
6. If you entered 'ppp0' you'll be requested if you want ddclient to run automatically every time you connect. You may want to select No here so you have the option of running it or not.
7. You'll then be asked if you want to run ddclient as a daemon. If this server is going to be a full-time Web or e-mail server with a broadband connection you should answer Yes to this.

Notes

The client will now be installed and the suitable configuration file like the ones shown above will be created. Even though the file was created for you, we showed you the typical files for both `dyndns.org` and EasyDNS services in case you need to edit them at a later point. If you want to examine your config file you can do so using the nano text editor with the command:

```
nano /etc/ddclient.conf
```

If you're using a modem connection you'll want to initial connect to your ISP with the `pon` command. If you didn't set `ddclient` to run as a daemon then just type in:

```
ddclient
```

at the shell prompt once you're connected. The resulting memo will tell you what IP address your external interface has (and what the DNS record will be updated with).

As mentioned previously, it will take awhile for this update to take affect. To see if it has taken affect yet, try pinging using your domain name and see if the returned IP address matches what was indicated in the message when you started `ddclient`.



Caution

If you used the above ping command in the background to keep your connection up you can still issue a second ping command in the foreground to check the returned IP address.

6.1.4 Other DNS Server Files

Given that a DNS server can host the zone files for lots of different domains, each having two zone files, it needs a way to tell which zone files are for which domains. It does this in the `named.conf` file which, like the zone files themselves, is located in the `/etc/bind` directory (which you'll see when we install Bind shortly).

Of the two zone files for each domain the one we've been talking about all along has been for forward lookups (resolving names to IP addresses). This zone file is usually named `db.my-last-name.net`.

DNS also offers a "reverse lookup" function that allows you to decode IP addresses to host/domain names. The information that allows this to happen is stored in the second zone file. Here's a reverse-lookup zone file that corresponds to the simpler zone file we showed earlier:

```
$TTL 86400
1.168.192.in-addr.arpa. IN SOA ns1.easydns.com. \
me.my-name.com. (
2004011522 ; Serial no., based on date
21600 ; Refresh after 6 hours
3600 ; Retry after 1 hour
604800 ; Expire after 7 days
3600 ; Minimum TTL of 1 hour
)
51 IN PTR debian
@ IN NS ns1.easydns.com.
@ IN NS ns2.easydns.com.
```

NS records are the same but there's no A records. And because we only have one system handling all three Web, e-mail, and FTP server functions we only need one PTR record. A PTR (Pointer) record is the opposite of an A record. It has the host part of the IP address and gives the corresponding hostname. Typically you want a PTR record for every A record in the forward-lookup file provided the server is in the domain. We don't have PTR records for the name servers above because they're in a different domain (and thus in a different address space).

Why is only the host part of the IP address required in this file? Because the network portion of the IP address is used when naming the reverse-lookup zone file, and it's reversed. Because 192.168.1.x is a Class C network, the first three octets make up the network portion of the IP address so it's used in the zone file name. Only the last octet specifies the individual host so it's used to specify the host in PTR records. With the above example IP address, the zone file would be named:

```
db.1.168.192
```

The reverse-lookup zone file is also situated in the /etc/bind directory. There's another place this naming convention is used. Take a look at the start of the SOA record. The domain is specified as

```
1.168.192.in-addr.arpa
```

in-addr.arpa is the default domain for all reverse lookups.

6.1.5 DNS Tools, Testing, and Troubleshooting

When you're testing changes to your DNS records things may not act the way you expect them to. What you need is some endurance. Most DNS servers cache lookups. If you make a change to a zone record on EasyDNS or dyndns.org, or the IP address you pulled from your ISP changes and ddclient sends the update, it'll take the DNS servers at EasyDNS or dyndns.org up to 15 minutes to update. Then the DNS server that your desktop system is using to resolve names may cache the old information for another 20 to 30 minutes.

If you're using a Windows system to test DNS changes remember that it also has a DNS cache. You can clear it manually in a DOS window with the command:

```
ipconfig /flushdns
```

As a result, if you create a change to your zone records give it at least 45 minutes before you try to see if the changes had the desired effect. Web browsers also cache name-to-address information. If you're using a Web browser to test your changes, you may want to go and delete all the files in the browser's cache directory as well.

The above make playing around with dynamic DNS when using a modem sort of a pain. You have to keep the connection up for for at least 45 minutes because if you disconnect, you'll pull a different IP address when you reconnect and your DNS records will have invalid IP addresses. That's why I showed you how to run the ping command in the background to keep the dial-up connection alive.

A DNS problem will likely be in one of three places:

1. The DNS server addresses specified in the TCP/IP configuration on the PC you are using to do the pinging are not correct.
2. The registrar's domain record does not have the correct name server hostnames and/or addresses.
3. The authoritative DNS servers for the domain do not have the domain's zone records configured properly.

Notes

The most essential tool for testing DNS is the ping command. If you can ping a Web server using its IP address but not it's domain name, you have a DNS problem. If you can ping a server using its domain name you'll notice that the server's IP address is also displayed. Verifying that this is the correct IP address will verify that DNS is working properly. Another thing ping can tell you is if you're pinging an actual server or an alias. Using the MIT example again, you may type in

```
ping www.mit.edu
```

but the response will be something like

```
Pinging DANDELION-PATCH.mit.edu
```

One more common tool for testing DNS is nslookup (name server lookup) and it's available on Linux systems and NT-class Windows systems (NT-WS, 2000 Pro). As you saw earlier in this page this command will show you what name server your PC is using to resolve names, as well as return hostname and address information on the server that's specified as the target of the command. However, it also has an interactive mode that increase its usefulness. If you simply type in:

```
nslookup
```

and you'll get a > prompt. There are some statements that you can enter at his prompt. One helpful one is when you want your system to send queries to a different, other than the default, name server. At the prompt type in the 'server' command followed by the IP address of the DNS server to use:

```
server 192.168.10.10
```

Then you just type in the domain name you want information on at the prompt. You'll see in the answer that the name server being queried has changed to the one you specified. Type 'exit' at the prompt when you're done. Another similar tool on Linux systems is the dig command. You can specify the alternate DNS server to use on the command line:

```
dig 192.168.10.10 mit.edu any
```

The any parameter tells it to return information on all record types. Check the man pages for dig and nslookup for more information.

If you wish to make sure that BIND isn't having a problem with your zone files, you can check the syslog after you boot the system (which is when BIND starts up and reads the zone files). At a shell prompt just type in:

```
nano /var/log/syslog
```

and look near the bottom of the file. You'll see messages when BIND was started. Ensure to see if any of them refer to any errors that were encountered. If it didn't have a problem with the zone file you'll see it referenced along with:

```
loaded serial 1
```

indicating that it has set the serial number (version) to 1.

6.1.6 A Simple DNS Server

As stated earlier, the most widely used DNS application is called BIND and installing it is simply a matter of entering the command:

```
apt-get install bind9
```

Congratulations! You now have a easy DNS server. Now just change the DNS server settings in the TCP/IP configuration files on the workstations on your LAN so that they start using this server as their preferred DNS server. You can use your ISP's DNS server(s) as alternate servers

as this will provide some redundancy if your server ever goes down. You'll also want to modify the `/etc/resolv.conf` file on the DNS server itself so that it points to itself. Do that by opening the file in a text editor with the command:

```
nano /etc/resolv.conf
and making sure the first nameserver line is:
nameserver 127.0.0.1
```

Why is setting up a easy DNS server so easy? Because of things called "root hints". The root hints are a list of root-level DNS servers in the `/etc/bind/db.root` file. Your simple DNS server will query a root server to get the addresses of authoritative DNS servers for each given domain (so it can contact those authoritative DNS servers to get the IP addresses of the desired hosts).

Just keep in mind that your simple DNS server needs a 24/7 connection to the Internet. Or it at least needs to be connected to the Internet any time any system on your LAN needs to access anything on the Internet.

6.1.7 DNS Server Functions

You can set up a DNS server for several different reasons:

1. **Internet Domain Support:** If you have a domain name and you're operating Web, e-mail, FTP, or other Internet servers, you'll use a DNS server to respond to resolution questions so others can find and access your server(s). This is a serious undertaking and you'd have to set up a minimum of two of them. On this page we'll refer to these types of DNS servers as authoritative DNS servers for reasons you'll see later. However, there are alternatives to having your own authoritative DNS server if you have (or want to have) your DNS records for you. Even if someone else is taking care of your domain's DNS records you could still set up one of the following your own domain name. You can have someone else host types of DNS servers.
2. **Local Name Resolution:** Related to the above scenario, this type of DNS server would resolve the hostnames of systems on your LAN. Typically in this scenario there is one DNS server and it does both jobs. The first being that it receives queries from workstations and the second being that it serves as the authoritative source for the responses (this will be more clear as we progress). Having this type of DNS server would eliminate the need to have (and manually update) a HOSTS file on each system on your LAN. On this page we'll refer to these as LAN DNS servers.

During the Debian installation you are asked to supply a domain name. This is an internal (private) domain name which is not noticeable to the outside world so, like the private IP address ranges you use on a LAN, it doesn't have to be registered with anyone. A LAN DNS server would be authoritative for this internal, private domain. For security reasons, the name for this internal domain should not be the same as any public domain name you have registered. Private domain names are not restricted to using one of the established public TLD (Top Level Domain) names such as `.com` or `.net`. You could use `.corp` or `.inc` or something else for your TLD. Since a single DNS server can be authoritative for multiple domains, you could use the same DNS server for both your public and private domains. However, the server would need to be accessible from both the Internet and the LAN so you'd need to locate it in a DMZ. Though you want to use different public and private domain names, you can use the same name for the second-level domain.



Example: My-domain.com for the public name and my-domain.inc for the private name.

Notes

3. **Internet Name Resolution:** LAN workstations and other desktop PCs need to send Internet domain name resolution queries to a DNS server. The DNS server most frequently used for this is the ISP's DNS servers. These are often the DNS servers you specify in your TCP/IP configuration. You can have your own DNS server respond to these resolution queries instead of using your ISP's DNS servers. My ISP recently had a problem where they would intermittently lose connectivity to the network segment that their DNS servers were connected to so they couldn't be contacted. It took me about 30 seconds to turn one of my Debian systems into this type of DNS server and I was surfing with no problems. On this page we'll refer to these as simple DNS servers. If a simple DNS server fails, you could just switch back to using your ISP's DNS servers. As a matter of fact, given that you typically specify two DNS servers in the TCP/IP configuration of most desktop PCs, you could have one of your ISP's DNS servers listed as the second (fallback) entry and you'd never miss a beat if your simple DNS server did go down. Turning your Debian system into a simple DNS server is simply a matter of entering a single command.

Don't get from this that you need three different types of DNS servers. If you were to set up a couple authoritative DNS servers they could also provide the functionality of LAN and simple DNS servers. And a LAN DNS server can simultaneously provide the functionality of a simple DNS server. It's a progressive type of thing.

If you were going to set up reliable DNS servers or a simple DNS server you'd have to have a 24/7 broadband connection to the Internet. Naturally, a LAN DNS server that didn't resolve Internet host/domain names wouldn't need this.

A DNS server is just a Debian system running a DNS application. The most extensively used DNS application is **BIND** (Berkeley Internet Name Domain) and it runs a daemon called **named** that, among other things, responds to resolution queries.



Case Study

Net Domain Name System gets 10 Applications

As part of the process of expanding the Internet's domain name system, the Internet Corporation for Assigned Names and Numbers (ICANN) has received 10 applications for sponsored top-level domains (sTLDs). Domains that have been applied for include ``.asia'`; ``.post'`; ``.mail'`; ``.travel'`; ``.tel'` and ``.jobs'`.

While an un-sponsored top-level domain operates under policies established by the global Internet community through the ICANN process, a sponsored top-level domain is a specialised one that has a sponsor representing the specific community that is most affected by the top-level domain.

So, for instance, the Bern-based Universal Postal Union has applied for the ``.post'` top-level domain, while the New York-based The Travel Partnership Corporation, a non-profit consortium of international travel organisations, has applied for the ``.travel'` top-level domain.

Other applications received by ICANN include those from the Hong Kong-based DotAsia Organisation Ltd for the ``.asia'` top-level domain, the Anti-Spam Community Registry's application for the ``.mail'` top-level domain and that of the US-based Society for Human Resources Management for ``.jobs'`. Interestingly, the Toronto-based International Foundation for Online Responsibility has applied for the ``.xxx'` top-level domain, while there are two applications for the ``.tel'` top-level domain.

The applications are currently open for 'public comment', which will close on April 30. Once this is over, an independent evaluation panel will review the applications for the

Contd...

sTLDs. If an application passes this review process, its sponsors can commence technical and contract negotiations with ICANN for allocation and sponsorship of the desired top-level domain.

Several organisations that have applied for these top-level domains have been formed specifically for this purpose. These include The Travel Partnership Corporation and DotAsia Organisation Ltd. The last expansion of top-level domains took place in 2000, when ICANN created seven new top-level domains, including three sTLDs – ‘.aero’, ‘.coop’ and ‘.museum’.

Notes

6.2 Summary

DNS is critical to the operation of the Internet. When you enter `www.some-domain.com` in a Web browser, it's DNS that takes the `www` host name and translates it to an IP address. If you set up a Debian system to act as a combination firewall, NAT, and home Web server you (and others if you wish) can access the Web pages on it (such as your Web cam images) from a remote location by entering the system's IP address in the URL. The IP address would be whatever is assigned to you by your ISP. If you have a broadband Internet connection without a static IP and have no desire to have your own domain name, you can use a free service offered by `dyndns.org` to set up a home Web/e-mail/ftp server. Given that a DNS server can host the zone files for many different domains, each having two zone files, it needs a way to tell which zone files are for which domains. It does this in the `named.conf` file which, like the zone files themselves, is located in the `/etc/bind` directory.

6.3 Keywords

BIND: Berkeley Internet Name Domain

DDNS: Dynamic DNS

DNS: Domain Name Server

FQDN: Fully Qualified Domain Name

LAN: Local Area Network

MX: Mail eXchanger

PTR: Pointer

URL: Universal Resource Locator

6.4 Self Assessment

Fill in the blanks:

1. Domain Name System (DNS) converts the name of a
2. A DNS client doesn't store DNS information; it must always refer to a to get it.
3. Check for correct DNS operation when a problem involving the inability to access an Internet resource.
4. The DNS servers you specify in your domain name application are the authoritative DNS servers for your domain.
5. Easy DNS will not only host your on their DNS server, but register your domain name all for \$35/year.

Notes

6. Make sure the cradle's USB and are connected to the host PC.
7. If using DHCP client, this will automatically be sent to you by the and loaded into this file as part of the DHCP protocol.
8. records specify the authoritative DNS servers for a domain.
9. If your DNS server is also performing as a caching DNS server, then you'll also need a view for to use.
10. for a domain are stored in the domain's zone file on the authoritative DNS servers.
11. A DNS client doesn't DNS information.
12. Most RedHat and Fedora Linux software products are obtainable in a format.
13. Dynamic DNS (DDNS) is the for a host (your Debian server) to update its own DNS A record.
14. zone file definitions list files to map domains to IP addresses.
15. is an acronym for the Berkeley Internet Name Domain project.

6.5 Review Questions

1. What is the processor to get your own domain?
2. Explain the most common types of zone records.
3. Discuss DDNS (Dynamic DNS) and how it can be used in your own domain?
4. Explain the DNS with domain.
5. What is ddclient? Describe the installing processor for ddclient.
6. Many dissimilar Web sites can map to a single IP address, but the reverse isn't true. Why?
7. In your opinion the error may indicate a failure in "resolving" the domain name to an IP address? If yes then why?
8. The host command accepts arguments that are either the fully qualified domain name or the IP address of the server when providing results. Comment
9. RedHat / Fedora BIND normally runs as the named process owned by the unprivileged named user. Why/why not?
10. The registrar's domain record does not have the correct name server hostnames and/or addresses. Is it a problem? Explain

Answers: Self Assessment

- | | | | |
|---------------------|----------------|--------------------|-----------|
| 1. Web site | 2. DNS server | 3. Troubleshooting | |
| 4. registration | 5. zone files | 6. serial cables | 7. ISP |
| 8. NS (Name Server) | 9. localhost | 10. DNS records | 11. Store |
| 12. Package | 13. Capability | 14. Forward | 15. BIND |

6.6 Further Readings

Notes



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

http://oreilly.com/catalog/debian/chapter/book/ch11_01.html

http://www.astahost.com/info.php/Can39t-Connect-Internet-Linux_t14887.html

http://wiki.answers.com/Q/How_do_you_connect_to_the_Internet_in_Linux

Unit 7: Installing Software

CONTENTS

Objectives

Introduction

7.1 RPM Meaning

7.2 Adding and Removing Packages

7.3 Querying RPM Package

7.3.1 GnoRPM

7.3.2 Compiling Software

7.3.3 Looking for Documentation

7.4 Configuring the Package

7.4.1 Compiling your Package

7.4.2 Installing the Package

7.5 Summary

7.6 Keywords

7.7 Self Assessment

7.8 Review Questions

7.9 Further Readings

Objectives

After studying this unit, you will be able to:

- Understand RPM meaning
- Discuss RPM management tool
- Know adding and removing packages
- Discuss querying RPM packages

Introduction

Several package managers are available for Linux to track and manipulate the applications installed on the system. The most widely used of these Linux package managers is the RPM Package Manager (formerly the Red Hat Package Manager), or RPM for short.

Although RPM was initially developed for Red Hat Linux, a combination of technical features and good timing has resulted in RPM's becoming the *de facto* standard for packaging software on most Linux distributions. The fact that Red Hat released the source code to the RPM software under an open-source license also helped its adoption.

The RPM Package Manager (RPM) is a powerful command line driven package management system capable of installing, uninstalling, verifying, querying, and updating computer software packages. Each software package consists of an archive of files along with information about the package like its version, a description, and the like. There is also a library API, permitting advanced developers to manage such transactions from programming languages such as C or Python.

7.1 RPM Meaning

RPM, the Red Hat Package Manager, is a powerful package manager that you can use to install, update and remove packages. It allows you to search for packages and keeps track of the files that come with each package. A system is built-in so that you can verify the authenticity of packages downloaded from the Internet. Advanced users can build their own packages with RPM.

The Red Hat Package Manager (RPM) is an open packaging system, available for anyone to use, which runs on Red Hat Linux as well as other Linux and UNIX systems. Red Hat, Inc. encourages other vendors to use RPM for their own products. RPM is distributable under the terms of the GPL.

For the end user, RPM makes system updates easy. Installing, uninstalling, and upgrading RPM packages can be accomplished with short commands. RPM maintains a database of installed packages and their files, so you can invoke powerful queries and verifications on your system. If you prefer a graphical interface, you can use Gnome-RPM to perform many RPM commands.

During upgrades, RPM handles configuration files carefully, so that you never lose your customizations – something that you will not accomplish with regular .tar.gz files. The RPM package contains a complete version of the program, which overwrites existing versions or installs as a new package.

For the developer, RPM allows you to take software source code and package it into source and binary packages for end users. This process is quite simple and is driven from a single file and optional patches that you create. This clear delineation of “pristine” sources and your patches and build instructions eases the maintenance of the package as new versions of the software are released.

An RPM package consists of an archive of files and meta-data used to install and erase the archive files. The meta-data includes helper scripts, file attributes, and descriptive information about the package. Packages come in two varieties: binary packages, used to encapsulate software to be installed, and source packages, containing the source code and recipe necessary to produce binary packages.

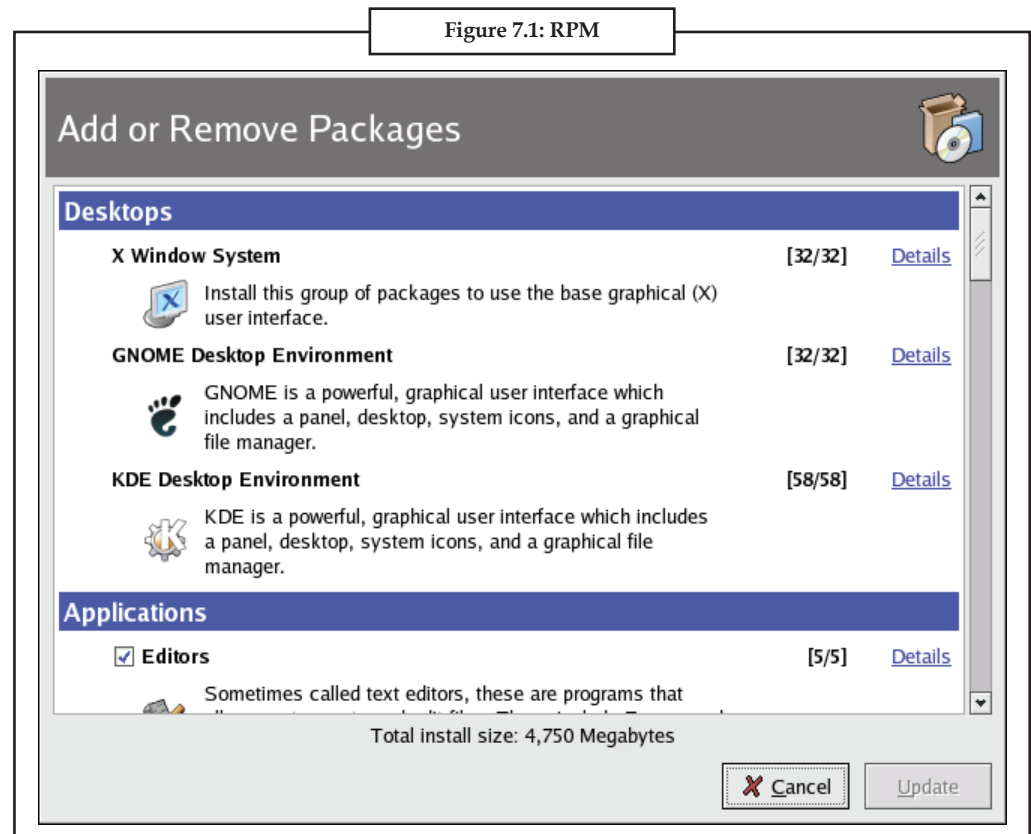
Many other distributions support RPM packages, among the popular ones Mandrake and SuSE Linux. Apart from the advice for your distribution, you will want to read man RPM.

Most packages are simply installed with the upgrade option, whether the package is already installed or not. New kernel packages, however, are installed with the install option which does not overwrite existing version(s) of the package, least to be able to boot your system with the old kernel if the new one does not work.



Did u know? Is the RPM package contains a complete version of the program?

Notes



RPM Management Goals

In order to understand how to use RPM, it can be helpful to understand RPM's design goals:

1. **Upgradability:** Using RPM, you can upgrade individual components of your system without completely reinstalling. When you get a new release of an operating system based on RPM (such as Red Hat Linux), you don't need to reinstall on your machine (as you do with operating systems based on other packaging systems). RPM allows intelligent, fully-automated, in-place upgrades of your system. Configuration files in packages are preserved across upgrades, so you won't lose your customizations. There are no special upgrade files need to upgrade a package because the same RPM file is used to install and upgrade the package on your system.
2. **Powerful Querying:** RPM is designed to provide powerful querying options. You can do searches through your entire database for packages or just for certain files. You can also easily find out what package a file belongs to and from where the package came. The files an RPM package contains are in a compressed archive, with a custom binary header containing useful information about the package and its contents, allowing you to query individual packages quickly and easily.
3. **System Verification:** Another powerful feature is the ability to verify packages. If you are worried that you deleted an important file for some package, simply verify the package. You will be notified of any anomalies. At that point, you can reinstall the package if necessary. Any configuration files that you modified are preserved during reinstallation.
4. **Pristine Sources:** A crucial design goal was to allow the use of "pristine" software sources, as distributed by the original authors of the software. With RPM, you have the pristine sources along with any patches that were used, plus complete build instructions. This is an

important advantage for several reasons. For instance, if a new version of a program comes out, you do not necessarily have to start from scratch to get it to compile. You can look at the patch to see what you might need to do. All the compiled-in defaults, and all of the changes that were made to get the software to build properly are easily visible using this technique.

The goal of keeping sources pristine may only seem important for developers, but it results in higher quality software for end users, too. We would like to thank the folks from the BOGUS distribution for originating the pristine source concept.



Task

Comment.

“RPM allows intelligent, fully-automated, in-place upgrades of your system.”

7.2 Adding and Removing Packages

The install mode, as its name suggests, is used to install RPM packages onto your system. Installing a package is accomplished with the `-i` option:

```
# rpm -i penguin-3.26.i386.rpm
```

Before installing the package, RPM performs several checks. First, it makes sure the package you are trying to install isn't already installed. RPM won't let you install a package on top of itself. It also checks that you are not installing an older version of the package. Next, RPM does a dependency check. Some packages depend on other packages being installed first. In this example, you have just downloaded the latest RPM version of Penguin utilities and now want to install it.

```
# rpm -i penguin-3.26.i386.rpm
```

```
failed dependencies:
```

```
iceberg >= 7.1 is needed by penguin-3.26.i386.rpm
```

This error indicates the penguin package failed to install because it requires the iceberg package with a version equal to or greater than 7.1. You'll have to find and install the iceberg package, and any packages iceberg requires.

Finally, RPM checks to see if any configuration files would be overwritten by the installation of this package. RPM tries to make intelligent decisions about what to do with conflicts. If RPM replaces an existing configuration file with one from the new package, a warning will be printed to the screen.

```
# rpm -I penguin-3.26.i386.rpm
```

```
warning: /etc/someconfig saved as /etc/someconfig.rpmsave
```



Note

need to be made.

It's up to you to look at both files and determine what modifications, if any,

Uninstalling a Package

The RPM `-e` command removes a package from your system. Like Install mode, RPM does some housekeeping before it will let you remove a package. First, it does a dependency check to make sure no other packages depend on the package you are removing. If you have modified any of the

Notes

configuration files, RPM makes a copy of the file, appends .rpmsave onto the end of it, then erases the original. Finally, after removing all files from your system and the RPM database, it removes the package name from the database.

Be very careful about which packages you remove from your system. Like most Linux utilities, RPM assumes omniscience and will silently let you shoot yourself in the foot. Removing the passwd or kernel package would be devastating.

Getting and Unpacking the Package

If a program is not packaged, it will usually come as a compressed archive. The archive may contain source code, precompiled binaries, and/or scripts.

All of these will need to be installed before they can be run. Source code will need to be compiled. Precompiled binaries and scripts will just need to be installed into the correct locations.

Scripts need an interpreter to be installed on the system they will run on – most Linux systems will already have interpreters for the most common scripting languages. Some scripts can be interpreted by a Linux shell.

Download the archive file, and put it in its own directory in /tmp. Then cd to that directory, and run the appropriate un-archive and decompress commands. Run the last extension first, so if a file is called filename.tar.gz, you should ungz it before you untar it. Software that isn't in a package is usually in some sort of archive. The most common archiving system is tar. Archives are then usually compressed, with one of the zip family of compression tools.

While the zip family of tools bear the same name as the Windows zip compression tools, they are only loosely related.

Tarred archives have the file extension .tar. To untar a file, run the command: tar -xvf (file name)

Zipped compressions have the file extension .zip. To unzip a file, run the command: unzip (file name)

Gzipped compressions have the file extension .gz, .Z, .z, .taz and .tgz. To ungz a file, run the command: gunzip (file name)

Bzipped compressions have the file extensions .bz, .bz2, .tbz or .tbz2. To unbzip a file, run the command: bunzip2 (file name)

There may be other archive formats, or other things to be done before compiling or installing the software. If there is anything unusual to be done, there should be instructions on the site where you found the software, or with the downloaded file.



Did u know? What is the most common archiving system?

7.3 Querying RPM Package

One of the strengths of RPM is that, ideally, it accounts for every system or application file on your system. Using RPM's query mode, you can determine which packages are installed on your system or which files belong to a particular package. This can be a big help if you want to locate a file that belongs to a certain package. Query mode can also be used to identify which files are in an RPM file before you install it. This lets you see the files that are going to be installed on your system before they are actually written.

The `-q` switch is used to query packages. By itself, `-q` will give you the version of a specified package. If you want to see which version of the tin newsreader you have on your system, you would issue the following command:

```
# rpm -q tin
tin-1.22-12
```

If you want to see which installed package owns a file, use the `-f` modifier. Here, we want to see which package owns `/etc/passwd`.

```
# rpm -q -f /etc/passwd
setup-1.9.2-1
```

Likewise, if you want to generate a list of files belonging to a certain package, use the `-l` modifier:

```
# rpm -q -l tin
/usr/bin/rtin
/usr/bin/tin
/usr/doc/tin-1.22
/usr/doc/tin-1.22/CHANGES
/usr/doc/tin-1.22/FTP
/usr/doc/tin-1.22/HACKERS
/usr/doc/tin-1.22/INSTALL
/usr/doc/tin-1.22/INSTALL.NNTP
/usr/doc/tin-1.22/MANIFEST
/usr/doc/tin-1.22/README
/usr/doc/tin-1.22/TODO
/usr/man/man1/tin.1
```

One of the most common modifiers to `-q` is `-a`, query all packages on your system. This system has 350 packages installed, but here's a truncated output:

```
# rpm -q -a
setup-1.9.2-1
filesystem-1.3.2-3
basesystem-4.9-3
ldconfig-1.9.5-8
...
code_crusader-1.1.0-1
lyx-0.11.53-1
xforms-0.86-1
wine-981211-1
Listing 1
```

For even more information about a package, use the `-i` (information) modifier:

```
# rpm -q -i passwd
```

Notes

Output is shown in Listing 1. Here's what some of the most important entries mean:

Name: the name of the package

Version: the version of the package

Release: the number of times this package has been released using the same version of the software

Install date: when this package was installed on your system

Group: your RPM database is divided into groups, which describe the functionality of the software. Each time you install a package, it will be grouped accordingly.

Size: the total size in bytes of all the files in the package

License: the license of the original software

Typically, the file name will indicate what's inside the package, but not always. You may receive a package simply named `glibc.rpm`, which isn't very helpful. You can use the `-p` modifier to find out which version and release this RPM contains, then perhaps rename it appropriately.

```
# rpm -q -p glibc.rpm
glibc-2.0.7-29
```



Task What are the uses of `-q` switch in query packages?

7.3.1 GnoRPM

Gnome-RPM (which is also referred to as `gnorpm`) is a graphical front end to the RPM package management system. It runs under X, like Redhat's Glint, but is written in C, and uses the GTK+ widget set and the Gnome Libraries.

Gnome-RPM was written by James Henstridge; RPM 3.0 support was written by Red Hat and additional `rpmfind` code was written by Daniel Veillard.

GnoRPM allows the end-user to easily work with RPM technology; it is fast, powerful and features a friendly interface.

Gnome-RPM is "GNOME-compliant," meaning that it seamlessly integrates into GNOME, the X Window System desktop environment.

With Gnome-RPM, you can easily

1. Install RPM packages
2. Uninstall RPM packages
3. Upgrade RPM packages
4. Find new RPM packages
5. Query RPM packages
6. Verify RPM packages

The interface features a menu, a toolbar, a tree and a display window of currently installed packages.

Operations are often performed in Gnome-RPM by finding and selecting packages, then choosing the type of operation to perform via push-button on the toolbar, through the menu or by right-clicking with the mouse.

Installing a package places all of the components of that package on your system in the correct locations.

Uninstalling a package removes all traces of the package except for configuration files you have modified.

Upgrading a package installs the newly available version and uninstalls all other versions that were previously installed. This allows quick upgrading to the latest releases of packages. Refer to the section called Configuration for information about how to alter the default settings for installing and uninstalling packages.

You can also use the Web find option to search the Internet for newly released packages. You can direct Gnome-RPM to search for particular distributions when you want to look for new packages.

Using Gnome-RPM to perform all of these and many other operations is the same as using RPM from the shell prompt. However, the graphical nature of Gnome-RPM often makes these operations easier to perform.

The usual way to work with Gnome-RPM is to display the available packages, select the package(s) you want to operate on, and then select an option from the toolbar or menu which performs the operation. However, Gnome-RPM is flexible enough to display packages in a variety of views, thanks to the use of filters. Refer to the section called Installing New Packages for more information on using filters to identify packages.

You can install, upgrade or uninstall several packages with a few button clicks. Similarly, you can query and verify more than one package at a time. Because of Gnome-RPM's integration with GNOME, you can also perform installation, query and verification on packages from within the GNOME File Manager.

You can start Gnome-RPM from either an Xterm window or from the GNOME desktop Panel (Main Menu Button → System → GnoRPM).

To start Gnome-RPM from an Xterm window, at the shell prompt, simply type

```
gnorpm &
```

That will bring up the main Gnome-RPM window.



Note

If you would like to install, upgrade or uninstall packages, you must be in root. The easiest way to do this is to type `su` to become root, and then type the root password at a shell prompt. However, it isn't necessary to be root in order to query and verify packages.

There are several parts to the Gnome-RPM interface.

Package Panel: On the left; allows you to browse and select packages on your system.

Display window: To the right of the package panel; shows you contents from folders in the panel.

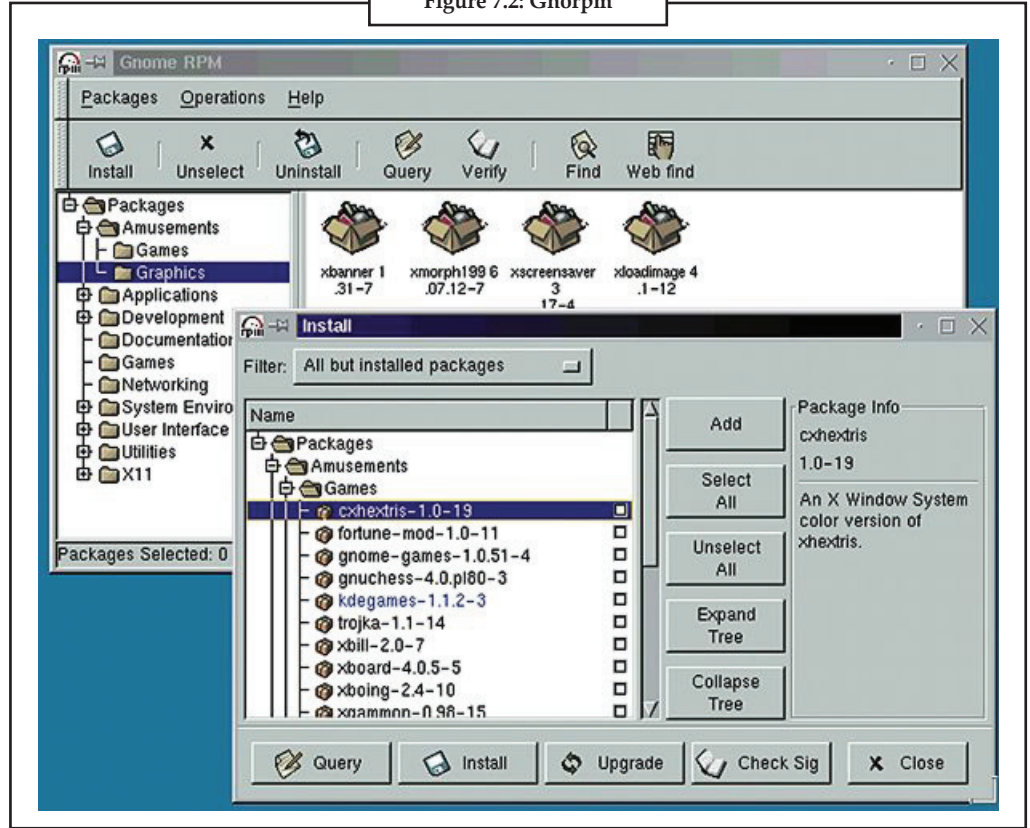
Toolbar: Above the display and panel; a graphical display of package tools.

Menu: Above the toolbar; contains text-based commands, as well as help info, preferences and other settings.

Status bar: Beneath the panel and display windows; shows the total number of selected packages.

Notes

Figure 7.2: GnoRpm



7.3.2 Compiling Software

In Windows, installing software is a matter of clicks. In Linux, Depending on the distribution you have, software can be downloaded in the form of either RPM or Deb packages. Again, you can compile software directly from source code – download the source code which comes in a tarball, unzip it and then compile it.

In fact, this is the way source packages were distributed in the old days, and you might still have to go by this route in some cases. However, to most people out there, compiling from source still feels like voodoo.

Unpacking

Command to use: [tar xvzf mypackage.tar.gz] or [tar xvjf mypackage.tar.bz2]

This is the first thing to be done when you download the software. All the source files, associated libraries and documentation are distributed as compressed archives called tarballs. They are compressed using either gzip or bzip2, and hence the different extensions and the slightly differing switches used in the command.

After unpacking, a directory will be created with the name of the package in the destination folder. Change the directory using cd mypackage and then use ls to explore the directory tree. Make sure to read the readme, install and other documentation.



Caution Some packages might need some additional libraries or might suffer from dependency issues, so it makes sense to know what's needed.

Configuring

Notes

Command to use: `./configure`

After you have unpacked the tarball and have also solved any dependency issues by installing required libraries it's time to go to the next step: configuration. You have to run the command while inside the installed package directory. This command does not change anything substantially. It basically does a house-keeping job, checking whether all the required resources in the form of system libraries are present and then assigning values for system dependent variables. Various switches can be used along with the `./configure` command to change the behaviour of the program.



Example: Appending `-quiet` would stop printing the checking... messages during the configure process. If you know what you are doing you can use `-no-create` to inspect the output files before they are created. Using `-prefix=mydirectory` you can change the path where the Makefile will be created.

After the `./configure` command has run—during which you will see a bunch of messages scrolling up the screen in rapid fire sequence—a Makefile will be created. This Makefile is then used to build the binary which then needs to be installed.

Building

Command to use: `make`

The `make` command uses the Makefile to create installable binaries. Binaries are the Unix equivalent of executables, or `.exe` files. The `make` command is time consuming and results in a whole bunch of messages scrolling across your screen. This part is going to take a lot of time, depending on the package being compiled as well as the system configuration. There will be another bunch of messages scrolling across the screen, sometimes with warnings about some resource being absent. If all is okay, it will display the command prompt. If however, there is some problem it prompts you with appropriate status messages.

Most of the errors in the compilation process are due to missing or incompatible libraries. Say you have a software that depends on GTK+, with the latest version not present. You might then have to download it from the Web. For the most part, if your OS is new you won't have any problems. However, you can always search software repositories provided by your OS vendor. Look for development versions which end with `-devel`.

Installation

Command to use: `make install`

The `make install` command is the equivalent of point and click routine on Windows. The installation time will again depend on how big the software is. Here is a quick guide to all that you need to know about compiling from source and what goes behind the scenes, without leaving anything to chance. Before doing this, you need to log in as root. Since you have followed the best practices and have up till now done everything from a user account type `su` (`sudo` for Ubuntu) and enter the root password. After getting administrator privileges use this command to install the software.

You will have no glitches and every thing will work out fine. Don't forget to log out by using `exit` when you are done. The program will be usually installed in `/usr/local/bin`. However, if you have specified a path during the configuration process, you will have to navigate to that directory to access the program. In most modern Linux distributions, you will see a graphical shortcut and will have to click there to launch the program.

Notes



Did u know? What are the uses of the make command?

7.3.3 Looking for Documentation

Most developers provide README or INSTALL files in the program archive. These are text files that include instructions on compiling and installing the program.

Linux provides a developer's utility called make. This utility allows the developer to provide a script called a Makefile, which, when run through make, will compile the program automatically. The Makefile can also include installation instructions.

In most cases, change to the directory containing the source code, then run the command make, followed by the command make install. There might also a configure script that needs to be run with ./configure before the make command.



Task

"The Makefile can also include installation instructions." Comment

7.4 Configuring the Package

Most packages ship with an auto-configuration script; it is safe to assume they do unless their documentation says otherwise. These scripts are typically named configure, and they take parameters. There are a handful of stock parameters that are available across all configure scripts, but the interesting stuff occurs on a program-by-program basis. Each package will have a handful of features that can be enabled or disabled or that have special values set at compile time, and they must be set up via configure.

To see what configure options come with a package, simply run:

```
./configure --help
```

Yes, those are two hyphens (--) before the word "help."

One commonly available option is --prefix. This option allows you to set the base directory where the package gets installed. By default, most packages use /usr/local. Each component in the package will install into the appropriate directory in /usr/local.

With all of the options you want set up, a final run of configure will create a special type of file called a makefile. Makefiles are the foundation of the compilation phase. Generally, if configure fails you will not get a makefile. Make sure that the configure command did indeed complete without any errors.

7.4.1 Compiling your Package

One of the key benefits of open-source software is that you have the source code in your hands. If the developer chooses to stop working on it, you can continue. If you find a problem, you can fix it. In other words, you are in control of the situation and not at the mercy of a commercial developer you can't control. But having the source code means you need to be able to compile it, too. Otherwise all you have is a bunch of text files that can't do much.

In this section, we will step through the process of compiling the Hello package, a GNU software package that might seem useless at first, but there are reasons for its existence. Most GNU software conforms to a standard method of installing, so let's go ahead and get the package.

Compiling your package is the easy part. All you need to do is run make, like so:

```
make
```


The make tool reads all of the makefiles that were created by the configure script. These files tell make which files to compile and the order in which to compile them –which is crucial since there could be hundreds of source files.

Depending on the speed of your system, the available memory, and how busy it is doing other things, the compilation process could take a while to complete, so don't be surprised.

As make is working, it will display each command it is running and all of the parameters associated with it. This output is usually the invocation of the compiler and all of the parameters passed to the compiler –it's pretty tedious stuff that even the programmers were inclined to automate!

If the compile goes through smoothly, you won't see any error messages. Most compiler error messages are very clear and distinct, so don't worry about possibly missing an error. If you do see an error, don't panic. Most error messages don't reflect a problem with the program itself, but usually with the system in some way or another. Typically, these messages are the result of inappropriate file permissions or files that cannot be found. In the latter case, make sure your path has at the very least the /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin, and /usr/X11R6/bin directories in it. You can see your path by issuing the following command:

```
echo $PATH
```

In general, slow down and read the error message. Even if the format is a little odd, it may explain what is wrong in plain English, thereby allowing you to quickly fix it.



Note

If the error is still confusing, look at the documentation that came with the package to see if there is a mailing list or e-mail address you can contact for help. Most developers are more than happy to provide help, but you need to remember to be nice and to the point.

7.4.2 Installing the Package

Most Linux distributions use package systems, which contain programs ready for installation and a record of what else those programs rely on.

When a package is installed, the installer checks whether the other files the program will need are present. Each installer handles missing files differently.

There are two major types of packages for Linux systems. `.rpm`, used by Red Hat Linux and distributions based on Red Hat (such as SuSE and Mandrake); and `.deb`, used by Debian Linux and distributions based on Debian.

Package managers can also be used to remove or upgrade software.

.rpm and rpm: To install a `.rpm` on a Linux system that uses rpms, first download the file then use the RPM package manager. The graphical versions are `gnorpm` for Gnome systems, and `KPackage` for KDE. If you use neither, the command line version is `RPM - install (package file)`.

On the command line, run `RPM - install - test (package file)` to determine which packages are needed by the package you're trying to install, and any other conflicts that the new software may cause. If this shows packages that aren't installed on your system, `rpm` may need to install those as well.

You can also find out which packages are required with `rpm --query --requires -p (package file)`.

`apt`, `deb` and `dpkg`

deb packages are designed for Debian Linux. The `apt` system is the simplest way to retrieve and install packages for Debian Linux.

Notes

apt: apt is a system for installing packages. It can be used for both types of package, and is primarily used on Debian systems.

To use apt, you need to set up an apt source. On Debian systems, the debian-config program sets up the sources during the installation process. Apt sources are locations where packages and package information can be downloaded. Once you have an apt source set up, the command apt-get install (package name) will install the package for you, and also install anything the package requires.

If you do not know the package name, the command apt-cache search (keywords) will search package names, file names and package descriptions for the keywords; and present a list of matching packages.

```
$ apt-cache search cervisia
cervisia - KDE based CVS frontend
```

The command apt-get update will automatically update the source and package lists, and should be used if you want the latest version of a package.

dpkg: The Debian package manager is another way to install .deb packages. dselect is a graphic front end to dpkg, and can be called by running the command dselect from the command line.

dpkg can be run directly from the command line, if desired.

alien: alien will convert one sort of package to the other. It works on rpm (Red Hat), deb (Debian), slp (Stampede), pkg (Solaris) and tgz (archived and compressed) files.

The alien manual explicitly states not to use it for system-critical programs, only for applications.



Caselet

Hitch while Installing Software

My system has the following configuration: Pentium III processor, 448 MHz, 40GB Hard disk, 128 MB RAM with Windows XP and Linux as the operating systems. Recently I tried to install software named 'Acrobat Reader 5.0.' It appeared to get installed successfully. But when I tried to open it, it displayed the following error message: AcroRd32.exe-Bad Image... The application or DLL C:\windows\System32\oledlg.dll is not a valid image. Please check this against your installation diskette."

What is causing this problem? Please suggest a solution.

Radeep

Oledlg.dll is a windows system file that may possibly have got corrupted in your system. You can replace this file from the Windows XP installation CD or download it from the following URL: <http://www.dll-files.com/dllindex/dll-files.shtml?oledlg>. But before you do so, please copy the original file (oledlg.dll) to somewhere else. To replace from Windows XP installation CD, please follow these steps: Click Start - Run - type cmd. This will bring up the command prompt. Type expand CD-ROM Drive Letter:\i386\oledlg.dl_ c:\windows\system32\oledlg.dll. (For example, if your CD-ROM drive letter is E:, then type "expand e:\i386\oledlg.dl_ c:\windows\system32\oledlg.dll"). Now check whether your adobe application works.

If you still face a problem, the next option is to run "sfc /scannow" from command prompt but you need the Windows XP installation CD.

This problem may be due to the W32/Lovgate-P virus. Please update your anti-virus definition files and then perform a full scan of your system. For more information.

Source: <http://www.thehindubusinessline.in/ew/2005/10/31/stories/2005103100250400.htm>

7.5 Summary

Notes

The RPM Package Manager is a powerful command line driven package management system capable of installing, uninstalling, verifying, querying, and updating computer software packages. Packages come in two varieties: binary packages, used to encapsulate software to be installed, and source packages, containing the source code and recipe necessary to produce binary packages. GnoRPM is a graphical front end to the RPM package management system. It runs under X, like Redhat's Glint, but is written in C, and uses the GTK+ widget set and the Gnome Libraries. GnoRPM allows the end-user to easily work with RPM technology; it is fast, powerful and features a friendly interface. If a program is not packaged, it will usually come as a compressed archive. The archive may contain source code, precompiled binaries, and/or scripts. All of these will need to be installed before they can be run. Source code will need to be compiled. Precompiled binaries and scripts will just need to be installed into the correct locations. Most developers provide README or INSTALL files in the program archive. These are text files that include instructions on compiling and installing the program.

7.6 Keywords

Binary RPM Package: One kind of RPM package which is used to encapsulate software to be installed.

GnoRPM: It is a graphical front end to the RPM package management system.

RPM Package Manager: It is a powerful command line driven package management system capable of installing, uninstalling, verifying, querying, and updating computer software packages.

Source RPM Package: One kind of RPM package which contains the source code and recipe necessary to produce binary packages.

7.7 Self Assessment

Fill in the blanks:

1. The full form of RPM is
2. The Red Hat Package Manager (RPM) is an packaging system
3. Many distributions support RPM packages, among them one is
4. is a graphical front end to the RPM package management system.
5. Gnome-RPM was written by
6. An RPM package consists of an archive of files and used to install and erase the archive files.
7. in packages are preserved across upgrades, so you won't lose your customizations.
8. GnoRPM allows the to easily work with RPM technology.
9. Linux provides a developer's utility called
10. RPM checks to see if any configuration files would be by the installation of this package.
11. When a package is installed, the installer checks whether the other the program will need are present.

Notes

12. The make command uses the to create installable binaries.
13. Scripts need an interpreter to be installed on the system they will run on - most Linux systems will already have for the most common scripting languages.
14. GnoRPM allows the end-user to easily work with technology.
15. Some packages might need some additional or might suffer from dependency issues, so it makes sense to know what's needed.

7.8 Review Questions

1. What is RPM Package Manager?
2. How to use RPM?
3. "The Red Hat Package Manager (RPM) is an open packaging system". Explain.
4. What is Gnome-RPM?
5. How to start Gnome-RPM from an Xterm window?
6. What are several parts to the Gnome-RPM interface?
7. Briefly explain Linux developer's utility "make".
8. In your opinion the RPM -e command removes a package from your system? How
9. "The RPM Package Manager (RPM) is a powerful command line driven package management system capable of installing, uninstalling, verifying, querying, and updating computer software packages." Comment
10. Explain how installing a package places all of the components of that package on your system in the correct locations.

Answers: Self Assessment

- | | |
|-----------------------------|-----------------|
| 1. RedHat Package Manager | 2. Open |
| 3. Mandrake and SuSE Linux. | 4. Gnome-RPM |
| 5. James Henstridge | 6. meta-data |
| 7. Configuration files | 8. end-user |
| 9. make | 10. overwritten |
| 11. files | 12. Makefile |
| 13. Interpreters | 14. RPM |
| 15. libraries | |

7.9 Further Readings



Books

- Brian Ward, *How Linux Works*, No Starch Press.
Christopher Negus, *Linux Bible*, Wiley.
Dee-Ann LeBlan and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins,
Linux in a Nutshell, O'Reilly Media.

Notes

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne
Media.



Online links

<http://www.control-escape.com/linux/lx-swininstall.html>

<http://www.linuxforums.org/forum/linux-tutorials-howtos-reference-material/64958-how-install-software-linux.html>

Unit 8: Shell

CONTENTS

Objectives

Introduction

- 8.1 Different Types of Shells
 - 8.1.1 BASH (Bourne-again Shell)
 - 8.1.2 C Shell (csh)
 - 8.1.3 Korn Shell (ksh)
 - 8.1.4 TCSH
- 8.2 Common Shell Commands
- 8.3 Summary
- 8.4 Keywords
- 8.5 Self Assessment
- 8.6 Review Questions
- 8.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss different types of shells
- Understand common shells commands

Introduction

Computers understand the language of zeros and ones known as binary language. In the early days of computing, instructions were provided using binary language, which is difficult for all of us humans to read and write. Therefore, in an operating system there is a special program called the shell. The shell accepts human readable commands and translates them into something the kernel can read and process.

It is a command language interpreter that executes commands read from the standard input device (keyboard) or from a file. The shell is a user program or it is an environment provided for user interaction.

Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc.

Several shells are available for Linux including:

Notes

Shell Name	Developed by	Where	Remark
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey	Free Software Foundation	Most common shell in Linux. It's Freeware shell.
CSH (C SHell)	Bill Joy	University of California (For BSD)	The C shell's syntax and usage are very similar to the C programming language.
KSH (Korn SHell)	David Korn	AT & T Bell Labs	—
TCSH	See the man page. Type \$ man tcsh	—	TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell (CSH).

To find all of the available shells in your system, type the following command:

```
$ cat /etc/shells
```

Each shell does the same job, but each understands different command syntax and provides different built-in functions.



Did u know? What is shell?

8.1 Different Types of Shells

8.1.1 BASH (Bourne-again Shell)

Bash is a free software UNIX shell written for the GNU Project. Its name is an acronym which stands for Bourne-again shell. The name is a pun on the name of the Bourne shell (sh), an early and important Unix shell written by Stephen Bourne and distributed with Version 7 Unix circa 1978, and “born again”. Bash was created in 1987 by Brian Fox. In 1990 Chet Ramey became the primary maintainer.

Bash is the default shell on most systems built on top of the Linux kernel as well as on Mac OS X and it can be run on most Unix-like operating systems. It has also been ported to Microsoft Windows using Subsystem for UNIX-based Applications (SUA), or POSIX emulation provided by Cygwin and MSYS. It has been ported to MS-DOS by the DJGPP project and to Novell NetWare.

8.1.2 C Shell (csh)

The C shell (csh) is a Unix shell developed by Bill Joy for the BSD Unix system. It was originally derived from the 6th Edition Unix /bin/sh (which was the Thompson shell), the predecessor of the Bourne shell. Its syntax is modeled after the C programming language. The C shell added many feature improvements over the Bourne shell, such as aliases and command history. Today, the original C shell is not in wide use on Unix; it has been superseded by other shells such as the Tenex C shell (tcsh) based on the original C shell code, but adding filename completion and command line editing, features later copied in the Korn shell (ksh), and the GNU Bourne-Again shell (bash). An independently-developed and modernized C shell, created by Nicole Hamilton, also survives on Windows in the form of Hamilton C shell.

Notes

8.1.3 Korn Shell (ksh)

The Korn shell (ksh) is a Unix shell which was developed by David Korn (AT&T Bell Laboratories) in the early 1980s. It is backwards-compatible with the Bourne shell and includes many features of the C shell as well, such as a command history, which was inspired by the requests of Bell Labs users.

The main advantage of ksh over the traditional Unix shell is in its use as a programming language. Since its conception, several features were gradually added, while maintaining strong backwards compatibility with the Bourne shell.

For interactive use, ksh provides the ability to edit the command line in a WYSIWYG fashion, by hitting the appropriate cursor-up or previous-line key-sequence to recall a previous command, and then edit the command as if the users were in edit line mode. Three modes are available, compatible with vi, emacs and gmacs.

Until 2000, Korn Shell remained AT&T's proprietary software. Since then it has been open source software, originally under a license peculiar to AT&T. But, since the 93q release in early 2005, it has been licensed under the Common Public License. Korn Shell is available as part of the AT&T Software Technology (AST) Open Source Software Collection. As ksh was initially only available through a commercial license from AT&T, a number of free and open source alternatives were created. These include the public domain pdksh, the Free Software Foundation's Bourne-Again-Shell bash, and zsh.

Although the ksh93 version added many improvements (associative arrays, floating point arithmetic, etc.), some vendors still ship their own version of the older ksh88 as /bin/ksh, sometimes with extensions (as of 2005 only Solaris and NCR UNIX (a.k.a. MP-RAS) ship ksh88, all other Unix vendors migrated to ksh93 and even Linux distributions started shipping ksh93). There are also two modified versions of ksh93 which add features for manipulating the graphical user interface: dtksh which is part of CDE and tksh which provides access to the Tk widget toolkit.

SKsh is an AmigaOS version, that offers several Amiga-specific features such as ARexx interoperability.

MKS Inc.'s MKS Korn shell is another commercial ksh reimplementations. It was included with Microsoft's Services for Unix (SFU) up to version 2.0. According to David Korn, the MKS Korn shell was not fully compatible with his own Korn shell implementation in 1998.

With the introduction of SFU Version 3.0, Microsoft has replaced the MKS Korn shell with a new and fully POSIX compliant Korn shell as part of the new native Interix subsystem technology. It is supported on Windows NT 4.0 SP6a+, Windows 2000, Windows XP Professional and Windows Server 2003. It is also available in the Subsystem for UNIX-based Applications (SUA) of Windows Vista Enterprise and Ultimate Editions and Windows Server 2008.



Task

"Korn Shell is available as part of the AT&T Software Technology (AST) Open Source Software Collection." Comment

8.1.4 TCSH

TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell. It is a command language interpreter usable both as an interactive login shell and a shell script command processor. It includes a command-line editor, programmable word completion, spelling correction, a history mechanism and job control.

TCSH is a programming language with conditional statements. The 't' in tcsh comes from the T in TENEX, an operating system which inspired Ken Greer, the author of tcsh, with its command-completion feature. Ken Greer worked on tcsh in the late 1970s at Carnegie Mellon University. Paul Placeway from The Ohio State University continued work on it in the 1980s, and since then it has been maintained by numerous people. Wilfredo Sanchez, the former lead engineer of Mac OS X, worked on tcsh in the early 1990s at MIT.

Early versions of Mac OS X shipped with tcsh as the default shell, but the default for new accounts is bash as of 10.3. (tcsh is still provided, and upgrading the OS does not change the shell of any existing accounts.) Iowa State's implementation of MIT's Project Athena (Project Vincent) by default uses tcsh as the default shell, although users can change this.



Note

The tcsh is the default shell of FreeBSD and its descendants like DragonFly BSD, PC-BSD and DesktopBSD.

8.2 Common Shell Commands

ls : list files/directories in a directory, comparable to dir in windows/dos.

ls -al : shows all files (including ones that start with a period), directories, and details attributes for each file.

cd : change directory `cd /usr/local/apache` : go to /usr/local/apache/ directory

cd ~ : go to your home directory

cd - : go to the last directory you were in

cd .. : go up a directory **cat** : print file contents to the screen

cat filename.txt : cat the contents of filename.txt to your screen

chmod: changes file access permissions

The set of 3 go in this order from left to right:

USER - GROUP - EVERYONE

0 = --- No permission

1 = --X Execute only

2 = -W- Write only

3 = -WX Write and execute

4 = R-- Read only

5 = R-X Read and execute

6 = RW- Read and write

7 = RWX Read, write and execute

Usage:

`chmod numberpermissions filename`

`chmod 000` : No one can access

`chmod 644`: Usually for HTML pages

Notes

`chmod 755`: Usually for CGI scripts

chown: changes file ownership permissions

The set of 2 go in this order from left to right:

USER - GROUP

`chown root myfile.txt` : Changes the owner of the file to root

`chown root.root myfile.txt` : Changes the owner and group of the file to root

tail : like cat, but only reads the end of the file

`tail /var/log/messages` : see the last 20 (by default) lines of /var/log/messages

`tail -f /var/log/messages` : watch the file continuously, while it's being updated

`tail -200 /var/log/messages` : print the last 200 lines of the file to the screen

more : like cat, but opens the file one screen at a time rather than all at once

`more /etc/userdomains` : browse through the userdomains file. hit *Space* to go to the next page, *q* to quit

pico : friendly, easy to use file editor

`pico /home/burst/public_html/index.html` : edit the index page for the user's website.

File Editing with VI ssh commands

vi : another editor, tons of features, harder to use at first than pico

`vi /home/burst/public_html/index.html` : edit the index page for the user's website.

While in the vi program you can use the following useful commands, you will need to hit SHIFT

+ : to go into command mode

:q! : This force quits the file without saving and exits vi

:w : This writes the file to disk, saves it

:wq : This saves the file to disk and exists vi

:LINENUMBER : EG **:25** : Takes you to line 25 within the file

:\$: Takes you to the last line of the file

:0 : Takes you to the first line of the file

grep : looks for patterns in files

`grep root /etc/passwd` : shows all matches of root in /etc/passwd

`grep -v root /etc/passwd` : shows all lines that do not match root

ln : create's "links" between files and directories

ln -s /usr/local/apache/conf/httpd.conf /etc/httpd.conf: Now you can edit /etc/httpd.conf rather than the original. changes will affect the original, however you can delete the link and it will not delete the original.

last : shows who logged in and when

last -20 : shows only the last 20 logins

last -20 -a : shows last 20 logins, with the hostname in the last field

w : shows who is currently logged in and where they are logged in from.

who : This also shows who is on the server in an shell.

netstat : shows all current network connections.

netstat -an : shows all connections to the server, the source and destination ips and ports.

netstat -rn : shows routing table for all ips bound to the server.

top : shows live system processes in a nice table, memory information, uptime and other useful info. *top* then type *Shift + M* to sort by memory usage or *Shift + P* to sort by CPU usage

ps: *ps* is short for process status, which is similar to the *top* command. *t*'s used to show currently running processes and their PID.

A process ID is a unique number that identifies a process, with that you can kill or terminate a running program on your server (see *kill* command).

ps U username : shows processes for a certain user

ps aux : shows all system processes

ps aux --forest : shows all system processes like the above but organizes in a hierarchy that's very useful!

touch : create an empty file

touch /home/burst/public_html/404.html : create an empty file called 404.html in the directory /home/burst/public_html/

file : attempts to guess what type of file a file is by looking at it's content.

*file ** : prints out a list of all files/directories in a directory

du : shows disk usage.

du -sh : shows a summary, in human-readable form, of total disk space used in the current directory, including subdirectories.

*du -sh ** : same thing, but for each file and directory. helpful when finding large files taking up space.

Notes

wc : word count

wc -l filename.txt : tells how many lines are in filename.txt

cp : copy a file

cp filename filename.backup : copies filename to filename.backup

cp -a /home/burst/new_design/ /home/burst/public_html/* : copies

all files, retaining permissions form one directory to another.

*cp -av * ../newdir* : Copies all files and directories recurrively in the current directory INTO newdir

mv : Move a file command

mv oldfilename newfilename : Move a file or directory from oldfilename to newfilename

rm : delete a file

rm filename.txt : deletes filename.txt, will more than likely ask if you really want to delete it

rm -f filename.txt : deletes filename.txt, will not ask for confirmation before deleting.

rm -rf tmp/ : recursively deletes the directory tmp, and all files in it, including subdirectories. BE VERY CAREFULL WITH THIS COMMAND!!!

TAR: Creating and Extracting .tar.gz and .tar files

tar -zxvf file.tar.gz : Extracts the file

tar -xvf file.tar : Extracts the file

tar -cf archive.tar contents/ : Takes everything from contents/ and puts it into archive.tar

gzip -d filename.gz : Decompress the file, extract it

ZIP Files: Extracting .zip files shell command

unzip file.zip

Firewall - iptables commands

iptables -I INPUT -s IPADDRESSHERE -j DROP : This command stops any connections from the IP address

iptables -L : List all rules in iptables

iptables -F : Flushes all iptables rules (clears the firewall)

iptables --save : Saves the currenty ruleset in memory to disk
service iptables restart : Restarts iptables

Apache Shell Commands

httpd -v : Outputs the build date and version of the Apache server.

httpd -l : Lists compiled in Apache modules

httpd status : Only works if mod_status is enabled and shows a page of active connections service

httpd restart : Restarted Apache web server



Did u know? What are the top commands is excellent for managing your system processes?



Caselet Of Commands and Batch Files

I need some information about commands in batch file. I mentioned in batch file as @at time:10:15:11.35p "c:\data\byt." But the command is not run in the specified time. The batch file extension is .bat, which is the batchfile I mentioned with path in batch file.

Please tell me what the Schedule service is. How do I use the 'AT' 'at' command to use at command in a batch file? My system runs on: Windows NT 2000. Please suggest a solution with an example.

M.D. Anwar

"AT" is a command line utility used for scheduling tasks. It uses the same "Task Scheduler" service used by the windows scheduler in the control panel. You need to have this service running to have the "AT" command working. Please check out some information on the same by going to the "Services" icon in the control panel.

A correction to the above command is that you should remove the "time" parameter from the command line and just give the command as "at 10:15 c:\data\byt.bat". Since this command runs as a background process you can include "{circ}{gt} c:\output.txt" to the command line so that you can see what has actually happened, so your command will now look like this: "at 10:15 c:\data\byt.bat {circ}{gt} c:\output.txt".

There is one more command utility which you can use to schedule tasks, it is the schtasks.exe. It has all the functionality of "AT" and more. The scheduler in the control panel and schtasks.exe are one and the same. In other words, using schtasks.exe you can handle the windows scheduler from the command prompt itself.

Source: <http://www.thehindubusinessline.in/ew/2004/03/31/stories/2004033100320401.htm>

8.3 Summary

Computers understand the language of zeros and ones known as binary language. In the early days of computing, instructions were provided using binary language, which is difficult for all of us humans to read and write. Bash is a free software UNIX shell written for the GNU Project. Its name is an acronym which stands for Bourne-again shell. The C shell (csh) is a Unix shell developed by Bill Joy for the BSD Unix system. The Korn shell (ksh) is a Unix shell which was developed by David Korn (AT&T Bell Laboratories) in the early 1980s. TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell.

8.4 Keywords

C shell: The C shell (csh) is a Unix shell developed by Bill Joy for the BSD Unix system.

Korn shell: The Korn shell (ksh) is a Unix shell which was developed by David Korn (AT&T Bell Laboratories) in the early 1980s.

Notes

Shell: Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc. Bash is a free software UNIX shell written for the GNU Project.

TCSH: TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell.

8.5 Self Assessment

Fill in the blanks

1. The shell accepts human commands and translates them into something the kernel can read and process.
2. is the default shell on most systems built on top of the Linux kernel as well as on Mac OS X and it can be run on most Unix-like operating systems.
3. An independently-developed and modernized, created by Nicole Hamilton, also survives on Windows in the form of Hamilton C shell.
4. The main advantage of over the traditional Unix shell is in its use as a programming language.
5. As ksh was initially only available through a from AT&T, a number of free and open source alternatives were created.
6. Korn Shell is available as part of the AT&T Software Technology (AST) Collection.
7. TCSH is a with conditional statements.
8. The tcsh is the default shell of FreeBSD and its like DragonFly BSD, PC-BSD and DesktopBSD.
9. Theis a Unix shell developed by Bill Joy for the BSD Unix system.
10. For interactive use, ksh provides the ability to edit the command line in a fashion
11. Korn Shell is available as part of theOpen Source Software Collection.
12.is an AmigaOS version, that offers several Amiga-specific features such as ARexx interoperability.
13.command changes file access permissions
14. LINENUMBER : EG :25 takes you to line within the file
15. TCSH is an enhanced but completely compatible version of theshell.

8.6 Review Questions

1. List out the different types of shells in Linux.
2. Each shell does the same job. Comment
3. Explain the Common Shell Commands
4. Evaluate the uses of File System Commands in Linux. Explain all the file system commands.
5. Is it said that the main advantage of ksh over the traditional Unix shell is in its use as a programming language.Why so?

6. Make distinction between bash shell and C shell. Give examples.
7. The C shell added many feature improvements over the Bourne shell. Enlighten the statement.
8. Elaborate the concept of File Editing with VI ssh commands.
9. Elucidate various Apache Shell Commands.
10. Evaluate the advantage of ksh over the traditional Unix shell.

Notes

Answers: Self Assessment

- | | |
|------------------------------------|-------------------------|
| 1. Readable | 2. Bash |
| 3. C shell | 4. Ksh |
| 5. commercial license | 6. Open Source Software |
| 7. programming language | 8. descendants |
| 9. C shell (csh) | 10. WYSIWYG |
| 11. AT&T Software Technology (AST) | 12. SKsh |
| 13. Chmod | 14. 25 |
| 15. Berkeley UNIX C | |

8.7 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

<http://www.freeos.com/guides/lsst/>

<http://linux.about.com/od/linux101/a/desktop11.htm>

<http://www.linux.com/news/software/applications/17997-what-is-a-linux-shell>

Unit 9: File System Commands

CONTENTS

Objectives

Introduction

9.1 File System Commands

9.1.1 The /etc/passwd File

9.1.2 The /etc/shadow File

9.1.3 The /etc/group File

9.2 Environment Variables

9.3 Summary

9.4 Keywords

9.5 Self Assessment

9.6 Review Questions

9.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Understand file system commands
- Discuss Environmental Variables

Introduction

In this unit, we will discuss file system commands such as etc/passwd file, etc/shadow file, etc. Also you will understand the concept of environment variables.

9.1 File System Commands

Files those hold users' information are as follows:

1. *etc/passwd* - Where the *user's* name, and other pertinent information are stored. This includes the password unless your system is using shadow passwords.
2. */etc/shadow* - Where the user's password is stored if you are using shadow passwords.
3. */etc/group* - Where group names are stored.
4. */etc/aliases* - Where the user's name is matched to a nickname for e-mail.
5. */etc/sudoers* - A list of users with special privileges along with the commands they can execute.

Files those hold accounting information are as follows:

1. `/var/log/wtmp` – Stores information about all logins and logouts.
2. `/var/run/utmp` – Stores information about who is currently on the system and is used by the `who` command.
3. `/var/log/btmp` – Used to store information about failed logins.

9.1.1 The `/etc/passwd` File

`/etc/passwd` is a text file that contains the attributes of (i.e., basic information about) each user or account on a computer running Linux or another Unix-like operating system.

The permissions for `/etc/passwd` are by default set so that it is world readable, that is, so that it can be read by any user on the system. The file can be easily read using a text editor (such as `gedit` or `vi`) or with a command such as `cat`, which is commonly used to read files, i.e.,

```
cat /etc/passwd
```

Each line in `/etc/passwd` represents a single user. The first listed is the root (i.e., administrative) account, which has complete power over every aspect of the system. This is followed by system-defined groups and accounts that are required for proper installation and update of system software. The lines at the end represent real people who use the system.

Each line contains seven attributes or fields: name, password, user ID, group ID, `gecos`, home directory and shell. Each attribute is separated from the adjacent attributes by colons but with no spaces. Colons must not be used in the attributes themselves, in order to avoid confusing the system. If there is no data for an attribute, there is no space, but, rather, two consecutive colons.

Name is the user's login name, that is the name that a user types in when logging into the system. Each such name must be unique string (i.e., sequence of characters).

The password field originally contained an encrypted login password. However, for security reasons, the encrypted passwords are now contained on another file, `/etc/shadow`, that cannot be read by ordinary users. This field now merely contains the letter `x` to indicate that a password has been assigned to the user and is required for authentication. If this field is empty, the user can log in without a password.

User ID is the user's unique numeric identification number, which is used by the system for access control. Zero is reserved for the root account, and one through 99 are reserved for other predefined accounts. 100 through 999 are available for ordinary users and groups.

Group ID specifies the user's principal group identification number. This is usually the same as the user ID.

Named for historical reasons, `gecos` contains general information about the user that is not needed by the system, most commonly the user's real name. This field can be empty, in which case there is no space between the two delimiting colons. Alternatively, it can contain multiple entries, each separated by a comma.

Home directory is the full path (i.e., the location relative to the root directory) of the user's home directory. This is the directory that the user is first in when logging into the system and which contains programs and configuration files specific to that user.

Shell is the full path of the default shell for the user. A shell is a program that provides a text-only user interface and whose main purpose is to execute commands typed in by a user and display the results. The default shell on Linux is `bash`, whose absolute path is `/bin/bash`.

Notes

Although `/etc/passwd` can, by default, be read by any user, it can be changed only by the root user. The main, and safest, way that it is changed is through the use of commands such as `adduser` and `userdel`, which are employed by the root account to add users to and remove users from the system, respectively. It can also be changed by direct editing with a text editor; however, as is the case with other system configuration files, great care should be exercised, as slight errors can cause serious problems, and a backup copy should be made in advance.

With shadow passwords, the `/etc/passwd` file contains account information, and looks like this:

```
smithj:x:561:561:Joe Smith:/home/smithj:/bin/bash
```

Each field in a `passwd` entry is separated with `:` colon characters, and are as follows:

1. Username, up to 8 characters. Case-sensitive, usually all lowercase
2. An `x` in the password field. Passwords are stored in the `/etc/shadow` file.
3. Numeric user id. This is assigned by the `adduser` script. Unix uses this field, plus the following group field, to identify which files belong to the user.
4. Numeric group id. Red Hat uses group id's in a fairly unique manner for enhanced file security. Usually the group id will match the user id.
5. Full name of user. I'm not sure what the maximum length for this field is, but try to keep it reasonable (under 30 characters).
6. User's home directory. Usually `/home/username` (e.g. `/home/smithj`). All user's personal files, web pages, mail forwarding, etc. will be stored here.
7. User's "shell account". Often set to `/bin/bash` to provide access to the bash shell (my personal favorite shell).

Perhaps you do not wish to provide shell accounts for your users. You could create a script file called `/bin/sorrysh`,



Example: That would display some kind of error message and log the user off, and then set this script as their default shell.

9.1.2 The `/etc/shadow` File

`etc/shadow` file contains encrypted password as well as other information such as account or password expiration values, etc. The `/etc/shadow` file is readable only by the root account and is therefore less of a security risk.

While some other Linux distributions forces you to install the Shadow Password Suite in order to use the shadow format, Red Hat makes it simple. The `/etc/shadow` file contains password and account expiration information for users, and looks like this:

```
smithj:Ep6mckrOLChF.:10063:0:99999:7:::
```

As with the `passwd` file, each field in the shadow file is also separated with `:` colon characters, and are as follows:

1. Username, up to 8 characters. Case-sensitive, usually all lowercase. A direct match to the username in the `/etc/passwd` file.
2. Password, 13 character encrypted. A blank entry (eg. `::`) indicates a password is not required to log in (usually a bad idea), and a `!*` entry (eg. `::*`) indicates the account has been disabled.

3. The number of days (since January 1, 1970) since the password was last changed.
4. The number of days before password may be changed (0 indicates it may be changed at any time).
5. The number of days after which password must be changed (99999 indicates user can keep his or her password unchanged for many, many years).
6. The number of days to warn user of an expiring password (7 for a full week).
7. The number of days after password expires that account is disabled.
8. The number of days since January 1, 1970 that an account has been disabled.
9. A reserved field for possible future use.

Notes

*Note*

All fields are separated by a colon (:) symbol. It contains one entry per line for each user listed in /etc/passwd file.

9.1.3 The /etc/group File

In Linux systems, multiple users can be categorized into groups. Users on Linux systems are assigned to one or more groups for following reasons:

To share files or other resource with a small number of users:

1. Ease of user management
2. Ease of user monitoring
3. Group membership is perfect solution for large Linux installation.
4. Group membership gives you or your user special access to files and directories or devices which are permitted to that group

In Linux systems, every user must be a member of at least one group, which is identified by the numeric GID of the user's entry in /etc/passwd. This group is referred to as the primary group ID. A user may be listed as members of additional groups in the relevant groups entry in the /etc/group; the IDs of these groups are referred to as supplementary group IDs.

etc/group is a text file which defines the groups to which users belong under Linux and UNIX operating system. Under Unix / Linux multiple users can be categorized into groups. Unix file system permissions are organized into three classes, user, group, and others. The use of groups allows additional abilities to be delegated in an organized fashion, such as access to disks, printers, and other peripherals. This method, amongst others, also enables the Superuser to delegate some administrative tasks to normal users.

The /etc/group file contains the database that lists every group on your computer and its corresponding GID. Its format is similar to the format used by the /etc/passwd file.

It stores group information or defines the user groups i.e. it defines the groups to which users belong. There is one entry per line, and each line has the format (all fields are separated by a colon (:))

Consider following /etc/group line:

```
cdrom:x: 24:vivek, student13, raj
```


group_name: It is the name of group. If you run ls -l command, you will see this name printed in the group field.

Notes

Password: Generally password is not used, hence it is empty/blank. It can store encrypted password. This is useful to implement privileged groups.

Group ID (GID): Each user must be assigned a group ID. You can see this number in your /etc/passwd file.

Group List: It is a list of user names of users who are members of the group. The user names, must be separated by commas.

 <p><i>Task</i></p>	Examine all the advantages of the /etc/group File.
--	--

9.2 Environment Variables

All the programs that run under Linux are called as processes. Processes run continuously in Linux and you can kill or suspend different processes using various commands. When you start a program a new process is created. This process runs within what is called an environment. This particular environment would be having some characteristics which the program/process may interact with. Every program runs in its own environment. You can set parameters in this environment so that the running program can find desired values when it runs.

The bash environment variables are commands that are simply expected to be there. As long as bash has the right pointers, it will fulfill your commands quickly.

Setting a particular parameter is as simple as typing VARIABLE=value. This would set a parameter by the name VARIABLE with the value that you provide.

To see a list of the environment variables that are already set on your machine, type the following:

```
$ env
```

This would produce a long list. Just go through the list before reading the next part of the article. Linux by default sets many environment variables for you. You can modify the values of most of these variables. A few of the variables that are set are:

HOME=/home/ramesh

This would set the home directory to /home/ramesh. This is perfect in case your login name is ramesh and you have been given a directory named /home/ramesh . In case you don't want this to be your home directory but some other one you could indicate so by typing the new directory name. The HOME directory is always the directory that you are put in when you login.

There are many advantages of using the HOME variable. You can always reach your home directory by only typing ' cd ' at the prompt, irrespective of which directory you are presently within. This would immediately transfer you to your HOME directory. Besides in case you write scripts that have \$HOME present in them to refer to the current HOME directory, these scripts can be used by other users as well since \$HOME in their case would refer to their home directories.

PATH=/usr:/bin:/usr/local/bin:.

This is a very important environment variable. This sets the path that the shell would be looking at when it has to execute any program. It would search in all the directories that are present in the above line. Remember that entries are separated by a ' : ' . You can add any number of directories to this list. The above 3 directories entered is just an example.

The last entry in the PATH command is a ' . ' (period). This is an important addition that you could make in case it is not present on your system. The period indicates the current directory

in Linux. That means whenever you type a command, Linux would search for that program in all the directories that are in its PATH. Since there is a period in the PATH, Linux would also look in the current directory for program by the name (the directory from where you execute a command). Thus whenever you execute a program which is present in the current directory (maybe some scripts you have written on your own) you don't have to type a './programname'. You can only type 'programname' since the current directory is already in your PATH.

Remember that the PATH variable is a very important variable. In case you want to add some particular directory to your PATH variable and in case you try typing the following:

```
PATH =/newdirectory
```

This would replace the current PATH value with the new value only. What you would want is to append the new directory to the existing PATH value. For that to happen you should type:

```
PATH=$PATH:/newdirectory
```

This would add the new directory to the existing PATH value. Always a \$VARIABLE is substituted with the current value of the variable.

```
PS1=boss
```

PS1 is the shell prompt. It defines what you want your shell prompt to look like. By default it looks like a '\$' in bash shell. The above case would replace the default '\$' with a new 'boss'. Hence an ls command would look something like:

```
boss> ls
```

All your commands would now be typed at a 'boss' prompt instead of a '\$' prompt.

```
SHELL=/bin/bash
```

This tells where the program that represents your shell is to be found. In case you typed /bin/ksh in the above, then your bash shell would be replaced with the ksh shell (korn shell). So in case you are not happy with the bash shell, you could replace the bash with some other shell.

```
LOGNAME=ramesh
```

The LOGNAME is automatically set for you as the same as your login name. This variable is used in case you want to use your own login name in any script. This is the simplest way of getting your login name from within a script. Thus in case you use \$LOGNAME in any script the script would work for all users since the LOGNAME always holds the name of the current user.

A good use is in case you have been given a temporary directory to work with and to make temporary files then you would want to delete the files that you created. You could use a command with the \$LOGNAME in it to locate files that were created by you and then you could pass the result of this command to a rm command. This would be a neat way to delete all the files at one go, rather than find them one at a time.

There are more environment variables than the ones that are mentioned here. But most users would find the ones given here to be useful.



Note

There are more environment variables than the ones that are mentioned here. But most users would find the ones given here to be useful.

Notes

File Listings

The ls command, used to view lists of files in any directory to which a user has execute permission, has many interesting options. For example:

```
$ ls -liah *
22684 -rw-r--r--      1 blucher  users   952 Dec 28 18:43 .profile
19942 -rw-r--r--      1 scalish  users   30 Jan  3 20:00 test2.out
   925 -rwxr-xr-x      1 scalish  users   378 Sep  2 2002 test.sh
```

The listing above shows 8 columns:

1. The first column indicates the inode of the file, because we used the `-li` option. The remaining columns are normally displayed with the `-l` option.
2. The second column shows the file type and file access permissions.
3. The third shows the number of links, including directories.
4. The fourth and fifth columns show the owner and the group owner of the files. Here, the owner "blucher" belongs to the group "users".
5. The sixth column displays the file size with the units displayed, rather than the default number of bytes, because we used the `-lh` option.
6. The seventh column shows the date, which looks like three columns consisting of the month, day and year or time of day.
7. The eighth column has the filenames. Use of `-a` in the option list causes the list of hidden files, like `.profile`, to be included in the listing.

Here are some of the most commonly used ls flags:

- `-a` Lists all files, including hidden ones.
- `-l` Displays the file list in long format, including file details like size, time stamp, and owner.
- `-F` Adds a slash after the name for directories, an asterisk for executables, and an at sign (@) for linked files.
- `-r` Reverses the sort order (alphabetic or time).
- `-t` Sorts the list by the time each file was created.
- `-R` will the subdirectories recursively, which means it will show all the directories and files within the specified directory.
- `-s` will also show you the size of the files (in blocks, not bytes)
- `-h` will show the size in "human readable format" (i.e. 4K, 16M, 1G etc). Of course you must use this option in conjunction with the `-s` option.



Case Study

Windows File Systems

The configuration of my system is as follows: Pentium III, 40 GB hard disk, Samsung CD-RW and D-link external modem. My system is very slow and a lot of applications have expired due to shareware and demo software. I formatted my system about three years ago and would like to reformat it now. I know there are three types of formats: fat, fat32, and ntfs that are possible. Could you explain what they mean and which one I ought to opt for? Also, what operating system should I install?

The newer operating systems such as Windows 2000 and Windows XP support three file systems for formatting the hard disk. They are FAT, FAT32 and NTFS file systems.

Basically, a file system is a system for organising directories and files, generally in terms of how it is implemented in the disk operating system.

FAT: The fat (file allocation table) file system is an old, out-dated file system also called FAT16. Old operating systems such as MS-DOS, Windows 3.1 and Windows 95 (prior to OSR2) used FAT file systems. Windows 2000 supports the maximum size of fat file system, 4 GB only. So it is advisable not to use FAT file system.

FAT32: Operating systems such as Windows 95 OSR2, Windows 98, Me, 2000, and Windows XP support fat32 partitions. The fat32 file system is more efficient than fat because it supports larger partition sizes. For example, Windows 2000 will format FAT32 partitions up to 32GB in size. If you would like to use dual boot with Windows 98 or Windows Me with later OS such as Windows 2000/XP/2003, then you will have to use FAT32 instead of NTFS file systems because Windows 98/Me will not support NTFS file systems. You can convert the FAT32 file system to NTFS file system anytime. After converting to NTFS you cannot revert back to the FAT32 file system.

To convert a FAT or FAT32 file system to NTFS, please do the following: click Start - Run - type "convert C: /fs:ntfs" and click ok. Here C refers to the c drive to convert.

NTFS: The Windows NT File System (NTFS) is the latest file system supported by Windows 2000/XP/2003. It is a highly reliable and recoverable file system. The additional features of NTFS file systems are that they support NTFS compression (compression attribute), "Encrypting File System (EFS)" that is Encrypting attribute, Disk quotas, mounting volume as directory, local file and folder security. It formats NTFS partitions up to 2 TB. If you would like to use only one operating system (or dual boot without Windows 98/Me) then it is advisable always to use the NTFS file system.

Regarding the Operating System, Windows 98 is obsolete. Microsoft itself has withdrawn support for Windows 98. You can install either Windows 2000 or Windows XP. If you are installing Windows 2000, please make sure service pack 4 is installed and for Windows XP service pack 2 is essential. It is advisable to use the latest operating system Windows XP with service packs and required Windows updates.

9.3 Summary

/etc/passwd is a text file that contains the attributes of (i.e., basic information about) each user or account on a computer running Linux or another Unix-like operating system. etc/shadow file contains encrypted password as well as other information such as account or password expiration values, etc. In Linux systems, every user must be a member of at least one group, which is identified by the numeric GID of the user's entry in /etc/passwd. All the programs that run under Linux are called as processes. Processes run continuously in Linux and you can kill or suspend different processes using various commands.

Notes

9.4 Keywords

Group List: It is a list of user names of users who are members of the group.

ls command: It is used to view lists of files in any directory to which a user has execute permission.

9.5 Self Assessment

Fill in the blanks:

1. The for /etc/passwd are by default set so that it is world readable, that is, so that it can be read by any user on the system.
2. User ID is the user's unique numeric number, which is used by the system for access control.
3. In Linux systems, every user must be a member of at least one group, which is identified by the of the user's entry in /etc/passwd.
4. etc/group is a which defines the groups to which users belong under Linux and UNIX operating system.
5. Processes run in Linux and you can kill or suspend different processes using various commands.
6. The directory is always the directory that you are put in when you login.
7. All the programs that run under Linux are called as
8.specifies the user's principal group identification number.
9. Unix file system permissions are organized into three classes, user,, and others.
10. Setting a particular parameter is as simple as typing VARIABLE=.....
11. The last entry in the PATH command is a
12. Thecommand is used to view lists of files in any directory to which a user has execute permission
13. The /etc/shadow file is readable only by theaccount and is therefore less of a security risk.
14. The default shell on Linux is....., whose absolute path is /bin/bash.
15. Each line in /etc/passwd represents auser.

9.6 Review Questions

1. In your opinion, what are the permissions for /etc/passwd so that it is world readable?
2. Do you think that the /etc/group file contains the database that lists every group on your computer and its corresponding GID? State your view with reasons.
3. User ID is the user's unique numeric identification number, which is used by the system for access control. Comment
4. Examine the concept of Environment Variables
5. What are the most commonly used ls flags?
6. Elucidate the file system commands used to hold users' information.

7. Explicate the files used to hold accounting information.
8. Analyze different fields used in a passwd entry.
9. Identify the main factors that are to be considered while sharing files or other resource with a small number of users.
10. The ls command, used to view lists of files in any directory to which a user has execute permission, has many interesting options. Comment.

Notes

Answers: Self Assessment

- | | |
|-------------------|-------------------|
| 1. permissions | 2. identification |
| 3. numeric GID | 4. text file |
| 5. continuously | 6. HOME |
| 7. Processes | 8. Group ID |
| 9. Group | 10. Value |
| 11. '.' (period). | 12. Ls |
| 13. Root | 14. bash |
| 15. single | |

9.7 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

<http://www.freeos.com/guides/lsst/>

<http://linux.about.com/od/linux101/a/desktop11.htm>

<http://www.linux.com/news/software/applications/17997-what-is-a-linux-shell>

Unit 10: File System

CONTENTS

Objectives

Introduction

10.1 File Systems

10.1.1 Managing File Systems

10.1.2 Adding and Partitioning a Disk

10.1.3 Network File Systems

10.1.4 Quota Management

10.1.5 Enabling Disk Quotas

10.2 Anatomy of File System

10.3 File Permissions and Directories Permissions

10.3.1 Ownerships and Permissions

10.3.2 File and Directory Types

10.3.3 Change Ownership

10.3.4 Change Group

10.3.5 Change Mode

10.4 File Search Utilities

10.5 Summary

10.6 Keywords

10.7 Self Assessment

10.8 Review Questions

10.9 Further Readings

Objectives

After studying this unit, you will be able to:

- Understand meaning of file system
- Describe anatomy of file system
- Discuss file permissions and directories permissions
- Explain file search utilities

Introduction

A simple description of the UNIX system, also applicable to Linux, is this:

“On a UNIX system, everything is a file; if something is not a file, it is a process.”

This statement is true because there are special files that are more than just files (named pipes and sockets, for instance), but to keep things simple, saying that everything is a file is an acceptable generalization. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. Input and output devices, and generally all devices, are considered to be files, according to the system.

In order to manage all those files in an orderly fashion, man likes to think of them in an ordered tree-like structure on the hard disk, as we know from MS-DOS (Disk Operating System) for instance. The large branches contain more branches, and the branches at the end contain the tree's leaves or normal files.

10.1 File Systems

A file system is an organization of data and metadata on a storage device. With a vague definition like that, you know that the code required to support this will be interesting.

While it is reasonably safe to suppose that everything you encounter on a Linux system is a file, there are some exceptions.

1. **Directories:** Files that are lists of other files.
2. **Special files:** The mechanism used for input and output. Most special files are in /dev.
3. **Links:** A system to make a file or directory visible in multiple parts of the system's file tree.
4. **(Domain) sockets:** A special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.
5. **Named pipes:** Act more or less like sockets and form a way for processes to communicate with each other, without using network socket semantics.

The -l option to ls displays the file type, using the first character of each input line:

```
/Documents> ls -l
total 80
-rw-rw-r--  1 jaime  jaime  31744 Feb 21 17:56 intro Linux.doc
-rw-rw-r--  1 jaime  jaime  41472 Feb 21 17:56 Linux.doc
drwxrwxr-x  2 jaime  jaime   4096 Feb 25 11:50 course
```

This table gives an overview of the characters determining the file type:

Table 10.1: File Types in a Long List

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special file
s	Socket
p	Named pipe
b	Block device

Notes

In order not to always have to perform a long listing for seeing the file type, a lot of systems by default don't issue just `ls`, but `ls -F`, which suffixes file names with one of the characters `"/=*|@` to indicate the file type. To make it extra easy on the beginning user, both the `-F` and `--color` options are usually combined.

As a user, you only need to deal directly with plain files, executable files, directories and links. The special file types are there for making your system do what you demand from it and are dealt with by system administrators and programmers.

The first thing that most new users shifting from Windows will find confusing is navigating the Linux filesystem. The Linux filesystem does things a lot more differently than the Windows filesystem.

Everything starts from the root directory, represented by `'/'`, and then expands into sub-directories. Where DOS/Windows had various partitions and then directories under those partitions, Linux places all the partitions under the root directory by 'mounting' them under specific directories. Closest to root under Windows would be `c:`.

Under Windows, the various partitions are detected at boot and assigned a drive letter. Under Linux, unless you mount a partition or a device, the system does not know of the existence of that partition or device. This might not seem to be the easiest way to provide access to your partitions or devices but it offers great flexibility. Most files are just files, called regular files; they contain normal data, for example text files, executable files or programs, input for or output from a program and so on.

This kind of layout, known as the unified filesystem, does offer several advantages over the approach that Windows uses. Let's take the example of the `/usr` directory. This directory off the root directory contains most of the system executables. With the Linux filesystem, you can choose to mount it off another partition or even off another machine over the network. The underlying system will not know the difference because `/usr` appears to be a local directory that is part of the local directory structure. How many times have you wished to move around executables and data under Windows, only to run into registry and system errors? Try moving `c:windowssystem` to another partition or drive.



Did u know? What is named pipes?

Another point likely to confuse newbies is the use of the frontslash `'/'` instead of the backslash `'\'` as in DOS/Windows. So `c:windowssystem` would be `/c/windows/system`. Well, Linux is not going against convention here.

Unix has been around a lot longer than Windows and was the standard a lot before Windows was. Rather, DOS took the different path, using `'/'` for command-line options and `'\'` as the directory separator.

To liven up matters even more, Linux also chooses to be case sensitive. What this means that the case, whether in capitals or not, of the characters becomes very important. So this is not the same as `THIS` or `ThIs` for that matter. This one feature probably causes the most problems for newbies.

We now move on to the layout or the directory structure of the Linux filesystem. Given below is the result of a `'ls -p'` in the root directory.

`bin/ dev/ home/ lost+found/ proc/ sbin/ usr/`

`boot/ etc/ lib/ mnt/ root/ tmp/ var/`

/sbin - This directory contains all the binaries that are essential to the working of the system.

These include system administration as well as maintenance and hardware configuration programs. Find lilo, fdisk, init, ifconfig etc. here. These are the essential programs that are required by all the users. Another directory that contains system binaries is /usr/sbin.

This directory contains other binaries of use to the system administrator. This is where you will find the network daemons for your system along with other binaries that only the system administrator has access to, but which are not required for system maintenance, repair etc.

/bin - In contrast to /sbin, the bin directory contains several useful commands that are used by both the system administrator as well as non-privileged users. This directory usually contains the shells like bash, csh etc. as well as much used commands like cp, mv, rm, cat, ls.

There also is /usr/bin, which contains other user binaries. These binaries on the other hand are not essential for the user. The binaries in /bin however, a user cannot do without.

/boot - This directory contains the system.map file as well as the Linux kernel. LILO places the boot sector backups in this directory.

/dev - This is a very interesting directory that highlights one important characteristic of the Linux filesystem - everything is a file or a directory. Look through this directory and you should see hda1, hda2 etc., which represent the various partitions on the first master drive of the system. /dev/cdrom and /dev/fd0 represent your CDROM drive and your floppy drive. This may seem strange but it will make sense if you compare the characteristics of files to that of your hardware. Both can be read from and written to. Take /dev/dsp, for instance. This file represents your speaker device. So any data written to this file will be re-directed to your speaker. Try 'cat /etc/lilo.conf > /dev/dsp' and you should hear some sound on the speaker. That's the sound of your lilo.conf file! Similarly, sending data to and reading from /dev/ttyS0 (COM 1) will allow you to communicate with a device attached there - your modem.

/etc - This directory contains all the configuration files for your system. Your lilo.conf file lies in this directory as does hosts, resolv.conf and fstab. Under this directory will be X11 sub-directory which contains the configuration files for X. More importantly, the /etc/rc.d directory contains the system startup scripts. This is a good directory to backup often. It will definitely save you a lot of re-configuration later if you re-install or lose your current installation.

/home - Linux is a multi-user environment so each user is also assigned a specific directory which is accessible only to them and the system administrator. These are the user home directories, which can be found under /home/username. This directory also contains the user specific settings for programs like IRC, X etc.

/lib - This contains all the shared libraries that are required by system programs. Windows equivalent to a shared library would be a DLL file.

/lost+found - Linux should always go through a proper shutdown. Sometimes your system might crash or a power failure might take the machine down. Either way, at the next boot, a lengthy filesystem check using fsck will be done. Fsck will go through the system and try to recover any corrupt files that it finds. The result of this recovery operation will be placed in this directory.

/mnt - This is a generic mount point under which you mount your filesystems or devices. Mounting is the process by which you make a filesystem available to the system. After mounting your files will be accessible under the mount-point. This directory usually contains mount points or sub-directories where you mount your floppy and your CD. You can also create additional mount-points here if you want. There is no limitation to creating a mount-point anywhere on your system but convention says that you do not litter your file system with mount-points.

/opt - This directory contains all the software and add-on packages that are not part of the default installation. Generally you will find KDE and StarOffice here. Again, this directory is not used very often as it's mostly a standard in Unix installations.

Notes


/proc - This is a special directory on your system.

/root - We talked about user home directories earlier and well this one is the home directory of the user root. This is not to be confused with the system root, which is directory at the highest level in the filesystem.

/tmp - This directory contains mostly files that are required temporarily. Many programs use this to create lock files and for temporary storage of data. On some systems, this directory is cleared out at boot or at shutdown.

/usr - This is one of the most important directories in the system as it contains all the user binaries. X and its supporting libraries can be found here. User programs like telnet, ftp etc. are also placed here. */usr/doc* contains useful system documentation. */usr/src/linux* contains the source code for the Linux kernel.

/var - This directory contains spooling data like mail and also the output from the printer daemon. The system logs are also kept here in */var/log/messages*. You will also find the database for BIND in */var/named* and for NIS in */var/yp*.



Note The files recovered are not likely to be complete or make much sense but there always is a chance that something worthwhile is recovered.

10.1.1 Managing File Systems


In Linux, as it is for Unix, the separate file systems the system may use are not accessed by device identifiers (such as a drive number or a drive name) but instead they are combined into a single hierarchical tree structure that represents the file system as one whole single entity. Linux adds each new file system into this single file system tree as it is mounted.

All file systems, of whatever type, are mounted onto a directory and the files of the mounted file system cover up the existing contents of that directory. This directory is known as the mount directory or mount point. When the file system is unmounted, the mount directory's own files are once again revealed.

When disks are initialized (using *fdisk*, say) they have a partition structure imposed on them that divides the physical disk into a number of logical partitions. Each partition may hold a single file system, for example an EXT2 file system.

File systems organize files into logical hierarchical structures with directories, soft links and so on held in blocks on physical devices. Devices that can contain file systems are known as block devices. The IDE disk partition */dev/hda1*, the first partition of the first IDE disk drive in the system, is a block device.

The Linux file systems regard these block devices as simply linear collections of blocks, they do not know or care about the underlying physical disk's geometry. It is the task of each block device driver to map a request to read a particular block of its device into terms meaningful to its device; the particular track, sector and cylinder of its hard disk where the block is kept.



Task "File systems organize files into logical hierarchical structures with directories, soft links and so on held in blocks on physical devices." Comment

A file system has to look, feel and operate in the same way no matter what device is holding it. Moreover, using Linux's file systems, it does not matter (at least to the system user) that these different file systems are on different physical media controlled by different hardware controllers. The file system might not even be on the local system, it could just as well be a disk remotely mounted over a network link. Consider the following example where a Linux system has its root file system on a SCSI disk:

A	E	boot	etc	lib	opt	tmp	usr
C	F	cdrom	fd	proc	root	var	sbin
D	bin	dev	home	mnt	lost+found		

Neither the users nor the programs that operate on the files themselves need know that /C is in fact a mounted VFAT file system that is on the first IDE disk in the system. In the example (which is actually my home Linux system), /E is the master IDE disk on the second IDE controller. It does not matter either that the first IDE controller is a PCI controller and that the second is an ISA controller which also controls the IDE CDROM. We can dial into the network where we work using a modem and the PPP network protocol using a modem and in this case we can remotely mount my Alpha AXP Linux system's file systems on /mnt/remote.

The files in a file system are collections of data. A file system not only holds the data that is contained within the files of the file system but also the structure of the file system. It holds all of the information that Linux users and processes see as files, directories soft links, file protection information and so on. Moreover it must hold that information safely and securely, the basic integrity of the operating system depends on its file systems. Nobody would use an operating system that randomly lost data and files.

Minix, the first file system that Linux had is rather restrictive and lacking in performance.

Its filenames cannot be longer than 14 characters (which is still better than 8.3 filenames) and the maximum file size is 64MBytes. 64Mbytes might at first glance seem large enough but large file sizes are necessary to hold even modest databases. The first file system designed specifically for Linux, the Extended File system, or EXT, was introduced in April 1992 and cured a lot of the problems but it was still felt to lack performance.

So, in 1993, the Second Extended File system, or EXT2, was added. Stephen Tweedie first revealed that he was working on extending ext2 in Journaling the Linux ext2fs Filesystem in 1998 paper and later in a February 1999 kernel mailing list posting and the filesystem was merged with the mainline Linux kernel in November 2001 from 2.4.15 onward. Its main advantage over ext2 is journaling which improves reliability and eliminates the need to check the file system after an unclean shutdown. Its successor is ext4. The ext4 or fourth extended filesystem is a journaling file system developed as the successor to ext3. It was born as a series of backward compatible extensions to add 64-bit storage limits and other performance improvements to ext3.

An important development took place when the EXT file system was added into Linux. The real file systems were separated from the operating system and system services by an interface layer known as the Virtual File system, or VFS.

All of the details of the Linux file systems are translated by software so that all file systems appear identical to the rest of the Linux kernel and to programs running in the system. Linux's Virtual File system layer allows you to transparently mount the many different file systems at the same time.

The Linux Virtual File system is implemented so that access to its files is as fast and efficient as possible. It must also make sure that the files and their data are kept correctly. These two requirements can be at odds with each other. The Linux VFS caches information in memory from each file system as it is mounted and used. VFS allows Linux to support many, often very different, file systems, each presenting a common software interface to the VFS.

Notes

A lot of care must be taken to update the file system correctly as data within these caches is modified as files and directories are created, written to and deleted. If you could see the file system's data structures within the running kernel, you would be able to see data blocks being read and written by the file system. Data structures, describing the files and directories being accessed would be created and destroyed and all the time the device drivers would be working away, fetching and saving data.

The most important of these caches is the Buffer Cache, which is integrated into the way that the individual file systems access their underlying block devices. As blocks are accessed they are put into the Buffer Cache and kept on various queues depending on their states. The Buffer Cache not only caches data buffers, it also helps manage the asynchronous interface with the block device drivers.



Did u know? In how many ways VFS allows Linux to support file system?

10.1.2 Adding and Partitioning a Disk

Most people have a vague knowledge of what partitions are, since every operating system has the ability to create or remove them. It may seem strange that Linux uses more than one partition on the same disk, even when using the standard installation procedure, so some explanation is called for.

One of the goals of having different partitions is to achieve higher data security in case of disaster. By dividing the hard disk in partitions, data can be grouped and separated. When an accident occurs, only the data in the partition that got the hit will be damaged, while the data on the other partitions will most likely survive.

This principle dates from the days when Linux didn't have journaled file systems and power failures might have lead to disaster. The use of partitions remains for security and robustness reasons, so a breach on one part of the system doesn't automatically mean that the whole computer is in danger. This is currently the most important reason for partitioning. A simple example: a user creates a script, a program or a web application that starts filling up the disk. If the disk contains only one big partition, the entire system will stop functioning if the disk is full. If the user stores the data on a separate partition, then only that (data) partition will be affected, while the system partitions and possible other data partitions keep functioning.

Mind that having a journaled file system only provides data security in case of power failure and sudden disconnection of storage devices. This does not protect your data against bad blocks and logical errors in the file system. In those cases, you should use a RAID (Redundant Array of Inexpensive Disks) solution.

Partition Layout and Types

There are two kinds of major partitions on a Linux system:

1. **Data partition:** normal Linux system data, including the root partition containing all the data to start up and run the system; and
2. **Swap partition:** expansion of the computer's physical memory, extra memory on hard disk.

Most systems contain a root partition, one or more data partitions and one or more swap partitions. Systems in mixed environments may contain partitions for other system data, such as a partition with a FAT or VFAT file system for MS Windows data.

Most Linux systems use `fdisk` at installation time to set the partition type. As you may have noticed during the exercise from previous unit, this usually happens automatically. On some occasions, however, you may not be so lucky. In such cases, you will need to select the partition type manually and even manually do the actual partitioning. The standard Linux partitions have number 82 for swap and 83 for data, which can be journaled (`ext3`) or normal (`ext2`, on older systems). The `fdisk` utility has built-in help, should you forget these values.

Apart from these two, Linux supports a variety of other file system types, such as the relatively new Reiser file system, JFS, NFS, FATxx and many other file systems natively available on other (proprietary) operating systems.

The standard root partition (indicated with a single forward slash, `/`) is about 100-500 MB, and contains the system configuration files, most basic commands and server programs, system libraries, some temporary space and the home directory of the administrative user. A standard installation requires about 250 MB for the root partition.

Swap space (indicated with `swap`) is only accessible for the system itself, and is hidden from view during normal operation. Swap is the system that ensures, like on normal UNIX systems, that you can keep on working, whatever happens. On Linux, you will virtually never see irritating messages like Out of memory, please close some applications first and try again, because of this extra memory. The swap or virtual memory procedure has long been adopted by operating systems outside the UNIX world by now.

Using memory on a hard disk is naturally slower than using the real memory chips of a computer, but having this little extra is a great comfort.

Linux generally counts on having twice the amount of physical memory in the form of swap space on the hard disk. When installing a system, you have to know how you are going to do this.



Example: An example on a system with 512 MB of RAM:

1. **1st possibility:** one swap partition of 1 GB
2. **2nd possibility:** two swap partitions of 512 MB
3. **3rd possibility:** with two hard disks: 1 partition of 512 MB on each disk.

The last option will give the best results when a lot of I/O is to be expected.

Read the software documentation for specific guidelines. Some applications, such as databases, might require more swap space. Others, such as some handheld systems, might not have any swap at all by lack of a hard disk. Swap space may also depend on your kernel version.

The kernel is on a separate partition as well in many distributions, because it is the most important file of your system. If this is the case, you will find that you also have a `/boot` partition, holding your kernel(s) and accompanying data files.

The rest of the hard disk(s) is generally divided in data partitions, although it may be that all of the non-system critical data resides on one partition, for example when you perform a standard workstation installation. When non-critical data is separated on different partitions, it usually happens following a set pattern:

1. a partition for user programs (`/usr`)
2. a partition containing the users' personal data (`/home`)
3. a partition to store temporary data like print- and mail-queues (`/var`)
4. a partition for third party and extra software (`/opt`)

Notes

Once the partitions are made, you can only add more. Changing sizes or properties of existing partitions is possible but not advisable.


The division of hard disks into partitions is determined by the system administrator. On larger systems, he or she may even spread one partition over several hard disks, using the appropriate software. Most distributions allow for standard setups optimized for workstations (average users) and for general server purposes, but also accept customized partitions. During the installation process you can define your own partition layout using either your distribution specific tool, which is usually a straight forward graphical interface, or fdisk, a text-based tool for creating partitions and setting their properties.

A workstation or client installation is for use by mainly one and the same person. The selected software for installation reflects this and the stress is on common user packages, such as nice desktop themes, development tools, client programs for E-mail, multimedia software, web and other services. Everything is put together on one large partition, swap space twice the amount of RAM is added and your generic workstation is complete, providing the largest amount of disk space possible for personal use, but with the disadvantage of possible data integrity loss during problem situations.

On a server, system data tends to be separate from user data. Programs that offer services are kept in a different place than the data handled by this service. Different partitions will be created on such systems:

1. a partition with all data necessary to boot the machine
2. a partition with configuration data and server programs
3. one or more partitions containing the server data such as database tables, user mails, an ftp archive etc.
4. a partition with user programs and applications
5. one or more partitions for the user specific files (home directories)
6. one or more swap partitions (virtual memory)

Servers usually have more memory and thus more swap space. Certain server processes, such as databases, may require more swap space than usual; see the specific documentation for detailed information. For better performance, swap is often divided into different swap partitions.

 <i>Task</i>	Explain the major partitions of linux system
--	--

Mount Points

All partitions are attached to the system via a mount point. The mount point defines the place of a particular data set in the file system. Usually, all partitions are connected through the root partition. On this partition, which is indicated with the slash (/), directories are created. These empty directories will be the starting point of the partitions that are attached to them. An example, given a partition that holds the following directories:

videos/ cd-images/ pictures/

We want to attach this partition in the filesystem in a directory called /opt/media. In order to do this, the system administrator has to make sure that the directory /opt/media exists on the system. Preferably, it should be an empty directory. Then, using the mount command, the administrator can attach the partition to the system. When you look at the content of the formerly empty directory /opt/media, it will contain the files and directories that are on the

mounted medium (hard disk or partition of a hard disk, CD, DVD, flash card, USB or other storage device).

During system startup, all the partitions are thus mounted, as described in the file `/etc/fstab`. Some partitions are not mounted by default, for instance if they are not constantly connected to the system, such like the storage used by your digital camera. If well configured, the device will be mounted as soon as the system notices that it is connected, or it can be user-mountable, i.e. you don't need to be system administrator to attach and detach the device to and from the system.

On a running system, information about the partitions and their mount points can be displayed using the `df` command (which stands for disk full or disk free). In Linux, `df` is the GNU version, and supports the `-h` or human readable option which greatly improves readability.



Note Commercial UNIX machines commonly have their own versions of `df` and many other commands. Their behavior is usually the same, though GNU versions of common tools often have more and better features.

The `df` command only displays information about active non-swap partitions. These can include partitions from other networked systems, like in the example below where the home directories are mounted from a file server on the network, a situation often encountered in corporate environments.

```
> df -h

Filesystem      Size  Used Avail Use% Mounted on
/dev/hda8       496M  183M  288M  39% /
/dev/hda1       124M   8.4M  109M   8% /boot
/dev/hda5       19G   15G   2.7G  85% /opt
/dev/hda6       7.0G   5.4G   1.2G  81% /usr
/dev/hda7       3.7G   2.7G   867M  77% /var
fsl:/home       8.9G   3.7G   4.7G  44% /.automount/fsl/root/home
```

10.1.3 Network File Systems

The Network File System (NFS) was originally developed by SUN Microsystems as a protocol that allowed communications between different computing environments. The NFS enables a Linux workstation to mount an exported share from the server into its own filesystem, thus giving the user and the client the appearance that the sub filesystem belongs to the client; it provides a seamless network mount point.

NFS consists of at least two main parts: a server and one or more clients. The client remotely accesses the data that is stored on the server machine. In order for this to function properly a few processes have to be configured and running.

The server has to be running the following daemons:

Daemon	Description
<code>nfsd</code>	The NFS daemon which services requests from the NFS clients.
<code>mountd</code>	The NFS mount daemon which carries out the requests that <code>nfsd</code> passes on to it.
<code>rpcbind</code>	This daemon allows NFS clients to discover which port the NFS server is using.

Notes

The client can also run a daemon, known as `nfsiod`. The `nfsiod` daemon services the requests from the NFS server. This is optional, and improves performance, but is not required for normal and correct operation.

NFS allows hosts to mount partitions on a remote system and use them as though they are local file systems. This allows the system administrator to store resources in a central location on the network, providing authorized users continuous access to them.

Linux uses a combination of kernel-level support and continuously running daemon processes to provide NFS file sharing, however, NFS support must be enabled in the Linux kernel in order to function. NFS uses Remote Procedure Calls (RPC) to route requests between clients and servers, meaning that the portmap service must be enabled and active at the proper runlevels for NFS communication to occur. Working with portmap, the following processes ensure that a given NFS connection is allowed and may proceed without error:

1. **`rpc.mountd`** – The running process that receives the mount request from an NFS client and checks to see if it matches with a currently exported file system.
2. **`rpc.nfsd`** – The process that implements the user-space components of the NFS service. It works with the Linux kernel to meet the dynamic demands of NFS clients, such as providing additional server threads for NFS clients to use.
3. **`rpc.lockd`** – A daemon that is not necessary with modern kernels. NFS file locking is now done by the kernel. It is included with the `nfs-utils` package for users of older kernels that do not include this functionality by default.
4. **`rpc.statd`** – Implements the Network Status Monitor (NSM) RPC protocol. This provides reboot notification when an NFS server is restarted without being gracefully brought down.
5. **`rpc.rquotad`** – An RPC server that provides user quota information for remote users.

Not all of these programs are required for NFS service. The only services that must be enabled are `rpc.mountd`, `rpc.nfsd`, and `portmap`. The other daemons provide additional functionality and should only be used if the server environment requires them.

NFS version 2 uses the User Datagram Protocol (UDP) to provide a stateless network connection between the client and server. NFS version 3 can use UDP or TCP running over an IP. The stateless UDP connection minimizes network traffic, as the NFS server sends the client a cookie after the client is authorized to access the shared volume. This cookie is a random value stored on the server's side and is passed along with RPC requests from the client. The NFS server can be restarted without affecting the clients and the cookie will remain intact.

NFS only performs authentication when a client system attempts to mount a remote file system. To limit access, the NFS server first employs TCP wrappers. TCP wrappers reads the `/etc/hosts.allow` and `/etc/hosts.deny` files to determine if a particular client should be permitted or prevented access to the NFS server.

After the client is granted access by TCP wrappers, the NFS server refers to its configuration file, `/etc/exports`, to determine whether the client can mount any of the exported file systems. After granting access, any file and directory operations are sent to the server using remote procedure calls.



Task “NFS only performs authentication when a client system attempts to mount a remote file system.” Comment

Configuring NFS

Notes

NFS configuration is a relatively straightforward process. The processes that need to be running can all start at boot time with a few modifications to your `/etc/rc.conf` file.

On the NFS server, make sure that the following options are configured in the `/etc/rc.conf` file:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

`mountd` runs automatically whenever the NFS server is enabled.

On the client, make sure this option is present in `/etc/rc.conf`:

```
nfs_client_enable="YES"
```

Each line in `/etc/exports` specifies a file system to be exported and which machines have access to that file system. Along with what machines have access to that file system, access options may also be specified. There are many such options that can be used in this file but only a few will be mentioned here. You can easily discover other options by reading over the exports manual page.

Here are a few example `/etc/exports` entries:

The following examples give an idea of how to export file systems, although the settings may be different depending on your environment and network configuration. For instance, to export the `/cdrom` directory to three example machines that have the same domain name as the server (hence the lack of a domain name for each) or have entries in your `/etc/hosts` file. The `-ro` flag makes the exported file system read-only. With this flag, the remote system will not be able to write any changes to the exported file system.

```
/cdrom -ro host1 host2 host3
```

The following line exports `/home` to three hosts by IP address. This is a useful setup if you have a private network without a DNS server configured. Optionally the `/etc/hosts` file could be configured for internal hostnames. The `-alldirs` flag allows the subdirectories to be mount points. In other words, it will not mount the subdirectories but permit the client to mount only the directories that are required or needed.

```
/home -alldirs 10.0.0.2 10.0.0.3 10.0.0.4
```

The following line exports `/a` so that two clients from different domains may access the file system. The `-maproot=root` flag allows the root user on the remote system to write data on the exported file system as root. If the `-maproot=root` flag is not specified, then even if a user has root access on the remote system, he will not be able to modify files on the exported file system.

```
/a -maproot=root host.example.com box.example.org
```

In order for a client to access an exported file system, the client must have permission to do so. Make sure the client is listed in your `/etc/exports` file.

In `/etc/exports`, each line represents the export information for one file system to one host. A remote host can only be specified once per file system, and may only have one default entry.



Example: Assume that `/usr` is a single file system. The following `/etc/exports` would be invalid:

```
# Invalid when /usr is one file system
/usr/src client
/usr/ports client
```

Notes

One file system, /usr, has two lines specifying exports to the same host, client. The correct format for this situation is:

```
/usr/src /usr/ports client
```

The properties of one file system exported to a given host must all occur on one line. Lines without a client specified are treated as a single host. This limits how you can export file systems, but for most people this is not an issue.

The following is an example of a valid export list, where /usr and /exports are local file systems:

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root client01
/usr/src /usr/ports client02
# The client machines have root and can mount anywhere
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root client01 client02
/exports/obj -ro
```

The mountd daemon must be forced to recheck the /etc/exports file whenever it has been modified, so the changes can take effect. This can be accomplished either by sending a HUP signal to the running daemon:

```
# kill -HUP `cat /var/run/mountd.pid`
```

or by invoking the mountd rc script with the appropriate parameter:

```
# /etc/rc.d/mountd onereoad
```

Alternatively, a reboot will make FreeBSD set everything up properly. A reboot is not necessary though. Executing the following commands as root should start everything up.

On the NFS server:

```
# rpcbind
# nfsd -u -t -n 4
# mountd -r
```

On the NFS client:

```
# nfsiod -n 4
```

Now everything should be ready to actually mount a remote file system. In these examples the server's name will be server and the client's name will be client. If you only want to temporarily mount a remote file system or would rather test the configuration, just execute a command like this as root on the client:

```
# mount server:/home /mnt
```

This will mount the /home directory on the server at /mnt on the client. If everything is set up correctly you should be able to enter /mnt on the client and see all the files that are on the server.

If you want to automatically mount a remote file system each time the computer boots, add the file system to the /etc/fstab file. Here is an example:

```
server:/home /mnt nfs rw 0 0
```



Note The `/etc/exports` file specifies which file systems NFS should export (sometimes referred to as “share”).

10.1.4 Quota Management

This feature of Linux allows the system administrator to allocate a maximum amount of disk space a user or group may use. It allows to limit the amount of disk space and/or the number of files a user can use. It can be flexible in its adherence to the rules assigned and is applied per filesystem. The default Linux Kernel which comes with Redhat and Fedora Core comes with quota support compiled in.

Disk quotas are implemented on a per-file system basis. In other words, it is possible to configure quotas for `/home` (assuming `/home` is on its own file system), while leaving `/tmp` without any quotas at all.

Quotas can be set on two levels:

1. For individual users
2. For individual groups

This kind of flexibility makes it possible to give each user a small quota to handle “personal” file (such as email, reports, etc.), while allowing the projects they work on to have more sizable quotas (assuming the projects are given their own groups).

In addition, quotas can be set not just to control the number of disk blocks consumed, but also to control the number of inodes. Because inodes are used to contain file-related information, this allows control over the number of files that can be created.

But before we can implement quotas, we should have a better understanding of how they work. The first step in this process is to understand the manner in which disk quotas are applied. There are three major concepts that you should understand prior to implementing disk quotas:

Hard Limit

The hard limit defines the absolute maximum amount of disk space that a user or group can use. Once this limit is reached, no further disk space can be used.

Soft Limit

The soft limit defines the maximum amount of disk space that can be used. However, unlike the hard limit, the soft limit can be exceeded for a certain amount of time. That time is known as the grace period.

Grace Period

The grace period is the time during which the soft limit may be exceeded. The grace period can be expressed in seconds, minutes, hours, days, weeks, or months, giving the system administrator a great deal of freedom in determining how much time to give users to get their disk usage below their soft limit.

The quota tools package usually needs to be installed, it contains the command line tools. The quotas are not limited per default (set to 0). The limits are set with `edquota/setquota` for single user. A quota can be also duplicated to many users.

Notes



Caution

The file structure is different between the quota implementations, but the principle is the same: the values of blocks and inodes can be limited.

Only change the values of soft and hard. If not specified, the blocks are 1k. Users can check their quota by simply typing `quota`. Root can check all quotas. Activate the user quota in the `fstab` and remount the partition. If the partition is busy, either all locked files must be closed, or the system must be rebooted.

Creating disk quotas frequently isn't enough. You also have to manage the process by reviewing the quota needs of each user and adjusting them according to the policies of your company.

You'll need to make Linux scan its hard disks periodically to check for exceeded quotas.

The Syntax is

```
quota [ -guv | q ]
quota [ -uv | q ] user
quota [ -gv | q ] group
```

The Options are

- g Print group quotas for the group of which the user is a member.
- u Print user quotas (this is the default)
- v Verbose, will display quotas on filesystems where no storage is allocated.
- q Print a more terse message, containing only information on filesystems where usage is over quota.

Specifying both `-g` and `-u` displays both the user quotas and the group quotas (for the user).

Only the super-user may use the `-u` flag and the optional user argument to view the limits of other users. Non-super-users can use the `-g` flag and optional group argument to view only the limits of groups of which they are members.

The `-q` flag takes precedence over the `-v` flag.

Quota reports the quotas of all the filesystems listed in `/etc/fstab`. For filesystems that are NFS-mounted a call to the `rpc.rquotad` on the server machine is performed to get the information. If `quota` exits with a non-zero status, one or more filesystems are over quota.

Related Files are

`quota.user` located at the filesystem root with user quotas

`quota.group` located at the filesystem root with group quotas

`/etc/fstab` to find filesystem names and locations



Did u know? How can you manage the process by reviewing the quota?

10.1.5 Enabling Disk Quotas

In order to use disk quotas, you must first enable them. This process involves several steps:

1. Modifying `/etc/fstab`
2. Remounting the file system(s)

3. Running quotacheck
4. Assigning quotas

Let us look at these steps in more detail.

Modifying /etc/fstab

You can modify /etc/fstab configuration file which is used to list all disk partitions which are required to be mounted automatically whenever the system boots. You can add usrquota option and this option is mentioned in fstab main pages. Using the text editor of your choice, simply add the usrquota and/or grpquota options to the file systems that require quotas:

The file /etc/fstab looked like

```
LABEL=/home /home ext3 defaults 1 2
```

After editing /etc/fstab it should be

```
LABEL=/home /home ext3 defaults,usrquota 1 2
```

Remounting the File System

At this point you must remount each file system whose fstab entry has been modified. You may be able to simply umount and then mount the file system(s) by hand, but if the file system is currently in use by any processes, the easiest thing to do is to reboot the system.

After editing /etc/fstab file it needs to view the file for /home. This could be done using this command.

```
# mount -o remount /home
```

user exit command to get out from single user mode.

```
# exit
```

Running quotacheck

Once you have verified that the partitions have been remounted with user and group quotas enabled you will need to build a table of the current disk space usage. This can be done using the "quotacheck" command. The quotacheck command examines quota-enabled file systems, building a table of the current disk usage for each one. This table is then used to update the operating system's copy of disk usage. In addition, the file system's disk quota files are updated (or created, if they do not already exist).

Once the "quotacheck" utility has finished building the disk space usage table you may notice two files in the "/home" directory called: "aquota.group" and "aquota.user". These two files contain information on how much space is being used and by whom it is being used by.

Once the quotas have been enabled you can set and enforce quotas for users and groups.

```
quotacheck -avug
```

The options used in this example direct quotacheck to:

Check all quota-enabled, locally-mounted file systems (-a)

Display status information as the quota check proceeds (-v)

Check user disk quota information (-u)

Notes

Check group disk quota information (-g)

Once quotacheck has finished running, you should see the quota files corresponding to the enabled quotas (user and/or group) in the root directory of each quota-enabled file system

Now we are ready to begin assigning quotas.

Assigning Quotas

The mechanics of assigning disk quotas are relatively simple. The edquota program is used to edit a user or group quota. The command to manage quotas is “edquota”. The “edquota” command uses the “-u” qualifier to modify users quotas.

edquota -u damian

Once you have set up a user quota it is the same process for setting up a group quota. The “edquota” command issued with the “-g” qualifier allows you to modify group quotas

edquota -g users

edquota uses a text editor (which can be selected by setting the EDITOR environment variable to the full pathname of your preferred editor) to display and change the various settings.

 <i>Task</i>	Describe the various steps of enabling the disk quotas.
---	---

10.2 Anatomy of File System

When it comes to file systems, Linux® is the Swiss Army knife of operating systems. Linux supports a large number of file systems, from journaling to clustering to cryptographic. Linux is a wonderful platform for using standard and more exotic file systems and also for developing file systems.

This article explores the virtual file system (VFS)—sometimes called the virtual filesystem switch—in the Linux kernel and then reviews some of the major structures that tie file systems together.

The VFS keeps track of the currently-supported file systems, as well as those file systems that are currently mounted.

VFS

File systems can be dynamically added or removed from Linux using a set of registration functions. The kernel keeps a list of currently-supported file systems, which can be viewed from user space through the /proc file system. This virtual file also shows the devices currently associated with the file systems. To add a new file system to Linux, register_filesystem is called. This takes a single argument defining the reference to a file system structure (file_system_type), which defines the name of the file system, a set of attributes, and two superblock functions. A file system can also be unregistered.

Registering a new file system places the new file system and its pertinent information onto a file_systems list. This list defines the file systems that can be supported. You can view this list by typing cat /proc/filesystems at the command line.



Note

The VFS acts as the root level of the file-system interface.

10.3 File Permissions and Directories Permissions

10.3.1 Ownerships and Permissions

The Linux operating system differs from other computing environments in that it is not only a multitasking system but it is also a multi-user system as well.

It means that more than one user can be operating the computer at the same time. While your computer will only have one keyboard and monitor, it can still be used by more than one user. For example, if your computer is attached to a network, or the Internet, remote users can log in via telnet or ssh (secure shell) and operate the computer. In fact, remote users can execute X applications and have the graphical output displayed on a remote computer. The X Windows system supports this.

The multi-user capability of Linux is not a recent “innovation,” but rather a feature that is deeply ingrained into the design of the operating system. Linux is originated from UNIX. The environment in which UNIX was created, this makes perfect sense. Years ago before computers were “personal;” they were large, expensive, and centralized. In order to make this practical, a method had to be devised to protect the users from each other. After all, you could not allow the actions of one user to crash the computer, nor could you allow one user to interfere with the files belonging to another user.

Linux uses the same permissions scheme as Unix. Each file and directory on your system is assigned access rights for the owner of the file, the members of a group of related users, and everybody else. Rights can be assigned to read a file, to write a file, and to execute a file (i.e., run the file as a program).

A typical university computer system consisted of a large mainframe computer located in some building on campus and terminals were located throughout the campus, each connected to the large central computer. The computer would support many users at the same time.



Did u know? What are the basic permissions to the user to access the file?

To see the permission settings for a file, we can use the ls command as follows:

```
$ ls -l some_file
```

```
-rw-rw-r-- 1 me me 1097374 Sep 26 18:48 some_file
```

We can determine a lot from examining the results of this command:

1. The file “some_file” is owned by user “me”
2. User “me” has the right to read and write this file
3. The file is owned by the group “me”
4. Members of the group “me” can also read and write this file
5. Everybody else can read this file

Notes

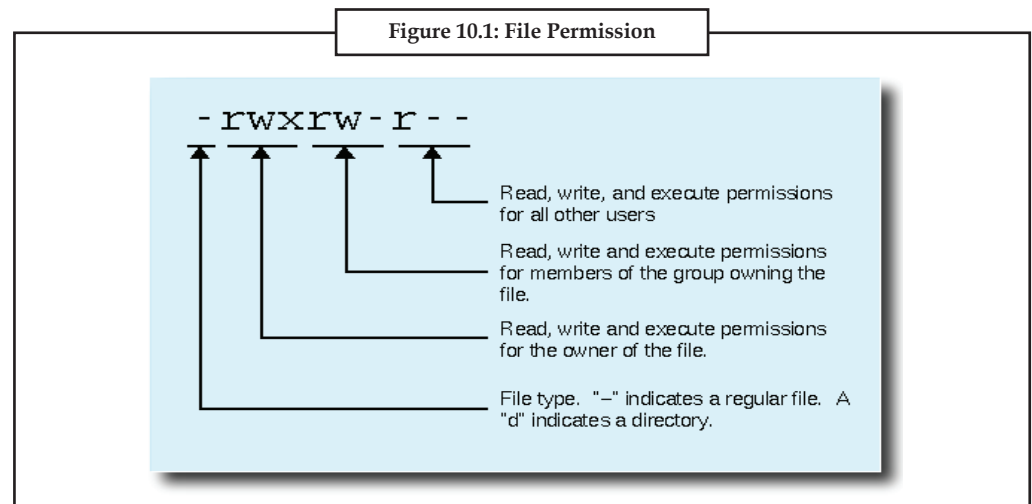
Let's try another example. We will look at the bash program which is located in the /bin directory:

```
$ ls -l /bin/bash
-rwxr-xr-x  1 root    root      316848 Feb 27  2000 /bin/bash
```

Here we can see:

1. The file "/bin/bash" is owned by user "root"
2. The superuser has the right to read, write, and execute this file
3. The file is owned by the group "root"
4. Members of the group "root" can also read and execute this file
5. Everybody else can read and execute this file

In the Figure 10.1, we see how the first portion of the listing is interpreted. It consists of a character indicating the file type, followed by three sets of three characters that convey the reading, writing and execution permission for the owner, group, and everybody else.

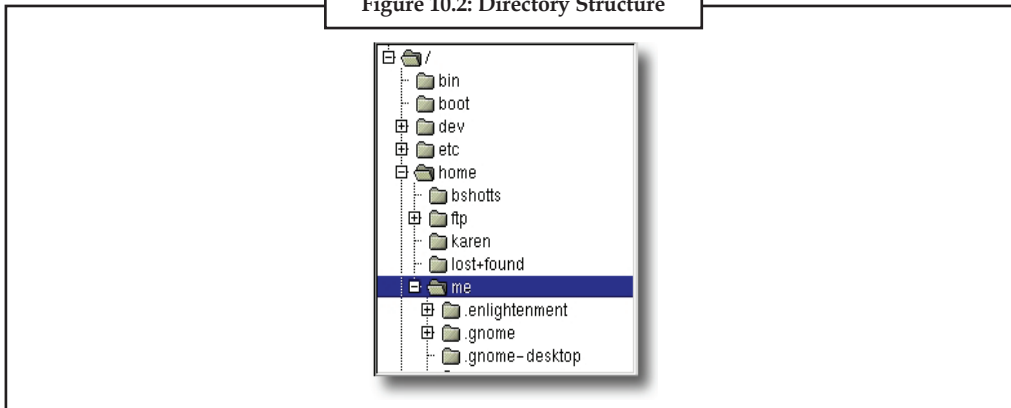


10.3.2 File and Directory Types

Like that legacy operating system, the files on a Linux system are arranged in what is called a hierarchical directory structure. This means that they are organized in a tree-like pattern of directories (called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and subdirectories which contain more files and subdirectories and so on.

Most graphical environments today include a file manager program to view and manipulate the contents of the file system. Often you will see the file system represented like this:

Figure 10.2: Directory Structure



One important difference between the legacy operating system and Unix/Linux is that Linux does not employ the concept of drive letters. While drive letters split the file system into a series of different trees (one for each drive), Linux always has a single tree. Different storage devices may contain different branches of the tree, but there is always a single tree.

**Task**

Analyse most graphical environments today include a file manager program to view and manipulate the contents of the file system.

Since a command line interface cannot provide graphic pictures of the file system structure, it must have a different way of representing it. Think of the file system tree as a maze, and you are standing in it. At any given moment, you stand in a single directory. Inside that directory, you can see its files and the pathway to its parent directory and the pathways to the subdirectories of the directory in which you are standing.

The directory you are standing in is called the working directory. To find the name of the working directory, use the `pwd` command.

When you first log on to a Linux system, the working directory is set to your home directory. This is where you put your files. On most systems, your home directory will be called `/home/your_user_name`, but it can be anything according to the whims of the system administrator.

To list the files in the working directory, use the `ls` command.


To change your working directory (where you are standing in the maze) you use the `cd` command. To do this, type `cd` followed by the pathname of the desired working directory. A pathname is the route you take along the branches of the tree to get to the directory you want. Pathnames can be specified in one of two different ways; absolute pathnames or relative pathnames. Let's deal with absolute pathnames first.

An absolute pathname begins with the root directory and follows the tree branch by branch until the path to the desired directory or file is completed. For example, there is a directory on your system in which programs are installed for the X window system. The pathname of the directory is `/usr/X11R6/bin`. This means from the root directory (represented by the leading slash in the pathname) there is a directory called "usr" which contains a directory called "X11R6" which contains a directory called "bin".

Now we can see that we have changed the current working directory to `/usr/X11R6/bin` and that it is full of files. Notice how your prompt has changed? As a convenience, it is usually set up to display the name of the working directory.

Notes

Where an absolute pathname starts from the root directory and leads to its destination, a relative pathname starts from the working directory. To do this, it uses a couple of special symbols to represent relative positions in the file system tree. These special symbols are “.” (dot) and “..” (dot dot).



Note The “.” symbol refers to the working directory and the “..” symbol refers to the working directory’s parent directory.

10.3.3 Change Ownership

The chown command changes the ownership of a file or files. This command is a useful method that root can use to shift file ownership from one user to another. An ordinary user may not change the ownership of files, not even her own files.

```
chown [Options]... NewOwner File...
chown [Options]... :Group File...
chown [Options]... --reference=RFILE File...
```

chown changes the user and/or group ownership of each given file, according to its first non-option argument, which is interpreted as follows. If only a user name (or numeric user ID) is given, that user is made the owner of each given file, and the files’ group is not changed. If the user name is followed by a colon or dot and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well. If the colon or dot and group are given, but the user name is omitted, only the group of the files is changed; in this case, chown performs the same function as chgrp.

OPTIONS:

- c
--changes
Verbosely describe the action for each FILE whose ownership actually changes.
- dereference
Do not act on symbolic links themselves but rather on what they point to.
- f
--silent
--quiet
Do not print error messages about files whose ownership cannot be changed.
- h
--no-dereference
Act on symbolic links themselves instead of what they point to. This is the default. This mode relies on the ‘lchown’ system call. On systems that do not provide the ‘lchown’ system call, ‘chown’ fails when a file specified on the command line is a symbolic link. By default, no diagnostic is issued for symbolic links encountered during a recursive traversal, but see ‘--verbose’.
- reference=FILE
Use the user and group of the reference FILE instead of an explicit NewOwner value.

-R

--recursive

Recursively change ownership of directories and their contents.

-v

--verbose

Verbosely describe the action (or non-action) taken for every FILE. If a symbolic link is encountered during a recursive traversal on a system without the 'lchown' system call, and '--no-dereference' is in effect, then issue a diagnostic saying neither the symbolic link nor its referent is being changed.

If a colon or dot but no group name follows the user name, that user is made the owner of the files and the group of the files is changed to that user's login group.



Did u know? Why the ordinary user cannot change the ownership of files?

10.3.4 Change Group

chgrp changes the group ownership of each given File to Group (which can be either a group name or a numeric group id) or to the group of an existing reference file.

chgrp [Options]... {Group | --reference=File} File...

-c

--changes

like verbose but report only when a change is made

--dereference

affect the referent of each symbolic link, rather than the symbolic link itself (this is the default)

-h

--no-dereference

affect each symbolic link instead of any referenced file (useful only on systems that can change the ownership of a symlink)

--no-preserve-root do not treat '/' specially (the default)

--preserve-root

fail to operate recursively on '/'

-f

--silent,

--quiet

suppress most error messages

--reference=RFILE

use RFILE's group rather than the specifying GROUP value

-R

--recursive

Notes operate on files and directories recursively

-v
--verbose
output a diagnostic for every file processed

The following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one takes effect.


-H
if a command line argument is a symbolic link to a directory, traverse it

-L
traverse every symbolic link to a directory encountered

-P
do not traverse any symbolic links (default)

--help
display this help and exit

--version
output version information and exit

 <i>Task</i>	Describe the syntax of changing the group.
--	--

10.3.5 Change Mode

The `chmod` ('change mode') command is used to change the modes (access permissions) associated with a file. The change is specified in the form:

```
bash$ chmod modes file
```

A mode is a three-item sequence which represents who, operation and permission.

who is one of:

1. *u user* – the owner of the file
2. *g group* – the group to which the owner belongs
3. *others* – everyone else
4. *a all* – user, group and others

Operation is one of:

1. + add the specified permission
2. - subtract the specified permission
3. = assign the specified permission, ignoring whatever may have been set before.

Permission is specified by one or more of:

1. r read

2. w write
3. x execute

In the following examples, the `chmod` command is given together with the old and new modes for the files concerned.

Here is a table of numbers that covers all the common settings. The ones beginning with “7” are used with programs (since they enable execution) and the rest are for other kinds of files.

Value	Meaning
777	<i>(rwxrwxrwx)</i> : No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	<i>(rwxr-xr-x)</i> : The file’s owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	<i>(rwx-----)</i> : The file’s owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	<i>(rw-rw-rw-)</i> : All users may read and write the file.
644	<i>(rw-r--r--)</i> : The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read but only the owner may change.
600	<i>(rw-----)</i> : The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

The `chmod` command can also be used to control the access permissions for directories. In most ways, the permissions scheme for directories works the same way as with files. However, the execution permission is used in a different way. It provides control for access to file listing and other things. Here are some useful settings for directories:

Value	Meaning
777	<i>(rwxrwxrwx)</i> : No restrictions on permissions. Anybody may list files, create new files in the directory and delete files in the directory. Generally not a good setting.
755	<i>(rwxr-xr-x)</i> : The directory owner has full access. All others may list the directory but cannot create files nor delete them. This setting is common for directories that you wish to share with other users.
700	<i>(rwx-----)</i> : The directory owner has full access. Nobody else has any rights. This setting is useful for directories that only the owner may use and must be kept private from others.



Example:

Given a file ‘file1’ with the default access modes (read and write for the user), add read mode for the group:

```
bash$ chmod g+r file1
```

```
user group others
```

Before: `rw- --- ---`

After: `rw- r-- ---`

Remove read and write access for everyone else from file ‘file2’ (showing that you can use multiple permissions):

```
bash$ chmod o-rw file2
```

```
user group others
```

Before: `rw- rw- rw-`

After: `rw- rw- ---`

Notes Give everyone (owner, group and others) read, write and execute permissions on file 'file3':
 bash\$ chmod a=rwx file3
 user group others
 Before: -irrelevant-
 After: rwx rwx rwx

10.4 File Search Utilities

File Management and Manipulation

1. cp - copy files and directories
2. mv - move or rename files and directories
3. rm - remove files and directories
4. mkdir - create directories

These four commands are among the most frequently used Linux commands. They are the basic commands for manipulating both files and directories.

Now, to be frank, some of the tasks performed by these commands are more easily done with a graphical file manager. With a file manager, you can drag and drop a file from one directory to another, cut and paste files, delete files, etc. So why use these old command line programs?

The answer is power and flexibility. While it is easy to perform simple file manipulations with a graphical file manager, complicated tasks can be easier with the command line programs. For example, how would you copy all the HTML files from one directory to another, but only copy files that did not exist in the destination directory or were newer than the versions in the destination directory? Pretty hard with with a file manager. Pretty easy with the command line:

```
$ cp -u *.html destination
```

Wildcards

Before we begin with our commands, let's talk about a shell feature that makes these commands so powerful. Since the shell uses filenames so much, it provides special characters to help you rapidly specify groups of filenames. These special characters are called wildcards. The table below lists the wildcards and what they select:

Summary of wildcards and their meanings.

Wildcard	Meaning
*	Matches any characters
?	Matches any single character
[characters]	Matches any character that is a member of the set characters. The set of characters can be expressed as a range of characters. (For example, [A-Z] represents all uppercase letters)
[!characters]	Matches any character that is not a member of the set characters

Using wildcards, it is possible to construct very sophisticated selection criteria for filenames. Here are some examples of patterns and what they match:

Examples of wildcard matching.

Pattern	Matches
*	All filenames
C*	All filenames that begin with the character "g"
B*.txt	All filenames that begin with the character "b" and end with the characters ".txt"
Data???	Any filename that begins with the characters "Data" followed by exactly 3 more characters
[abc]*	Any filename that begins with "a" or "b" or "c" followed by any other characters
[A-Z]*	Any filename that begins with an uppercase letter. This is an example of a range.
BACKUP. [0-9][0-9][0-9]	Another example of ranges. This pattern matches any filename that begins with the characters "BACKUP." followed by exactly 3 numerals.
[!a-z]*	Any filename that does not begin with a lowercase letter.

You can use wildcards with any command that accepts filename arguments.

cp

The cp program copies files and directories. In it's simplest form, it copies a single file:

```
$ cp file1 file2
```

It can also be used to copy multiple files to a different directory:

```
$ cp file1 file2 file3 directory
```

Other useful examples of cp and its options include:

Examples of the cp command

Command	Results
cp file1 file2	Copies the contents of file1 into file2. If file2 does not exist, it is created, otherwise, file2 is overwritten with the contents of file1.
cp -i file1 file2	Like above however, since the "-i" (interactive) option is specified, if file2 exists, the user is prompted before it is overwritten with the contents of file1.
cp file1 dir1	Copy the contents of file1 (into a file named file1) inside of directory dir1.
cp -R dir1 dir2	Copy the contents of the directory dir1. If directory dir2 does not exist, it is created. Otherwise, it creates a directory named dir1 within directory dir2.

mv

The mv command performs two different functions depending on how it is used. It will either move one or more files to a different directory, or it will rename a file or directory. To rename a file, it is used like this:

```
$ mv filename1 filename2
```

To move files to a different directory:

```
$ mv file1 file2 file3 directory
```

Notes

Examples of mv and its options include:

Command	Results
mv file1 file2	If file2 does not exist, then file1 is renamed file2. If file2 exists, its contents are replaced with the contents of file1.
mv -i file1 file2	Like above however, since the "-i" (interactive) option is specified, if file2 exists, the user is prompted before it is overwritten with the contents of file1.
mv file1 file2 file3 dir1	The files file1, file2, file3 are moved to directory dir1. dir1 must exist or mv will exit with an error.
mv dir1 dir2	If dir2 does not exist, then dir1 is renamed dir2. If dir2 exists, the directory dir1 is created within directory dir2.

rm

The rm command deletes (removes) files and directories.

\$ rm file

It can also be used to delete a directory:

\$ rm -r directory

Examples of rm and its options include:

Command	Results
rm file1 file2	Delete file1 and file2.
rm -i file1 file2	Like above however, since the "-i" (interactive) option is specified, the user is prompted before each file is deleted.
rm -r dir1 dir2	Directories dir1 and dir2 are deleted along with all of their contents.

Be careful with rm. Linux does not have an undelete command. Once you delete a file with rm, it's gone. You can inflict terrific damage on your system with rm if you are not careful, particularly with wildcards.

Before you use rm with wildcards, try this helpful trick: construct your command using ls instead. By doing this, you can see the effect of your wildcards before you delete files. After you have tested your command with ls, recall the command with the up-arrow key and then substitute rm for ls in the command.

mkdir

The mkdir command is used to create directories. To use it, you simply type:

\$ mkdir directory



Note Wildcards allow you to select filenames based on patterns of characters.

Process Manipulation

As with any multitasking operating system, Linux executes multiple, simultaneous processes. Well, they appear simultaneous anyway. Actually, a single processor computer can only execute one process at a time but the Linux kernel manages to give each process its turn at the processor and each appears to be running at the same time.

There are several commands that can be used to control processes. They are:

1. *ps* – list the processes running on the system
2. *kill* – send a signal to one or more processes (usually to “kill” a process)
3. *jobs* – an alternate way of listing your own processes
4. *bg* – put a process in the background
5. *fg* – put a process in the foreground

While it may seem that this subject is rather obscure, it can be very practical for the average user who mostly works with the graphical user interface. You might not know this, but most if not all of the graphical programs can be launched from the command line. Here’s an example: there is a small program supplied with the X Windows system called *xload* which displays a graph representing system load. You can execute this program by typing the following:

```
$ xload
```

Notice that the small *xload* window appears and begins to display the system load graph. Notice also that your prompt did not reappear after the program launched. The shell is waiting for the program to finish before control returns to you. If you close the *xload* window, the *xload* program terminates and the prompt returns.

Now in order to make life a little easier, we are going to launch the *xload* program again, but this time we will put it in the background so that the prompt will return. To do this we execute *xload* like this:

```
$ xload &
```

```
[1] 1223
```

```
$
```

In this case, the prompt returned because the process was put in the background.

Now imagine that you forgot to use the “&” symbol to put the program into the background. There is still hope. You can type control-z and the process will be suspended. The process still exists but is idling. To resume the process in the background type the *bg* command (short for background). Here is an example:

```
$ xload
```

```
[2]+ Stopped xload
```

```
$ bg
```

```
[2]+ xload &
```

Now that we have a process in the background, it would be helpful to display a list of the processes we have launched. To do this, we can use either the *jobs* command or the more powerful *ps* command.

```
$ jobs
```

```
[1]+ Running xload &
```

```
$ ps
```

```
PID TTY TIME CMD
```

```
1211 pts/4 00:00:00 bash
```

```
1246 pts/4 00:00:00 xload
```

```
1247 pts/4 00:00:00 ps
```

```
$
```

Notes

Suppose that you have a program that becomes unresponsive, how do you get rid of it? You use the kill command, of course. Let's try this out on xload. First you need to identify the process you want to kill. You can use either jobs or ps to do this. If you use jobs you will get back a job number. With ps, you are given a process id (PID). We will do it both ways:

```
$ xload &
[1] 1292
$ jobs
[1]+  Running xload &
$ kill %1
$ xload &
[2] 1293
[1] Terminated xload
$ ps
PID TTY TIME CMD
1280 pts/5 00:00:00 bash
1293 pts/5 00:00:00 xload
1294 pts/5 00:00:00 ps
$ kill 1293
[2]+ Terminated xload
$
```

While the kill command is used to "kill" processes, its real purpose is to send signals to processes. Most of the time the signal is intended to tell the process to go away, but there is more to it than that. Programs (if they are properly written) listen for signals from the operating system and respond to them, most often to allow some graceful method of terminating.



Example:

A text editor might listen for any signal that indicates that the user is logging off or that the computer is shutting down. When it receives this signal, it saves the work in progress before it exits. The kill command can send a variety of signals to processes. Typing:

```
kill -l
```

will give you a list of the signals it supports. Most are rather obscure, but several are useful to know:

Signal #	Name	Description
1	SIGHUP	Hang up signal. Programs can listen for this signal and act (or not act) upon it.
15	SIGTERM	Termination signal. This signal is given to processes to terminate them. Again, programs can process this signal and act upon it. You can also issue this signal directly by typing control-c in the terminal window where the program is running. This is the default signal sent by the kill command if no signal is specified.
9	SIGKILL	Kill signal. This signal causes the immediate termination of the process by the Linux kernel. Programs cannot listen for this signal.

Now let's suppose that you have a program that is hopelessly hung (Netscape, maybe) and you want to get rid of it. Here's what you do:

1. Use the `ps` command to get the process id (PID) of the process you want to terminate.
2. Issue a kill command for that PID.
3. If the process refuses to terminate (i.e., it is ignoring the signal), send increasingly harsh signals until it does terminate.

```
$ ps x
PID TTY STAT TIME COMMAND
2931 pts/5 SN 0:00 netscape
$ kill -SIGTERM 2931
$ kill -SIGKILL 2931
```

In the example above we used the `kill` command in the formal way. In actual practice, it is more common to do it in the following way since the default signal sent by `kill` is `SIGTERM` and `kill` can also use the signal number instead of the signal name:

```
$ kill 2931
```

Then, if the process does not terminate, force it with the `SIGKILL` signal:

```
$ kill -9 2931
```

In addition to the standard development tools used in software development projects, many other utilities are also helpful. Most of the time these utilities are not directly related to the software development process but are used as helping aids. Let us introduce to some of these utilities that are extremely helpful in the software development process. Some of the more common uses of these utilities are presented and readers are encouraged to experiment with them. If properly utilized, they may save time and increase productivity.

The `indent` utility is helpful in implementing a common programming style scheme to all source code files in a project. The `sed` utility is useful for searching and replacing text across multiple files. The `diff` command is used to display difference between two or more files. The `cscope` and `cbrowser` are used to locate symbols in a software source code tree. The `strace` and `ltrace` utilities are useful to find out system and library calls when a program is executed. GNU binary utilities are a set of utilities that is often used for different needs.

Indent Utility

One major objective of every software development project is that the code be well organized, structured and easy to understand for other members of the team. Proper indentation plays an important role in achieving these objectives. Most of the editors used in software development can do automatic indentation if they are configured to do so. The Emacs editor is the best example as it understands different programming languages and does indentation according to language rules and style. However you can come across code written by others that is poorly indented and you may like to re-indent it according to your own requirements or preferences. In large software development projects, you may try to enforce a particular coding style, but be assured that not everyone will follow it.

The `indent` programme is your tool to handle the situation and reformat all files in the project according to the style adopted for the project. You can do it by creating a script that goes into each directory of the project and reformats all files. In this section we shall explore some of its features and how to utilize it. By default, the `indent` programme uses the GNU style of coding and applies these rules to input files. It creates a backup file with original contents of the file. The reformatted file is created with the same name. The backup file ends with a tilde character `~`.

Notes

*Note*

The same scheme of creating backup files is used by the Emacs editor.

Sedutility

The sed utility is a stream editor that can be used for different file editing purposes when used as a filter. The most common task for software development purposes is the use of sed to search and replace text in source code files.

This utility reads one or more text files, makes editing changes according to a script of editing commands, and writes the results to standard output. The script is obtained from either the script operand string or a combination of the option-arguments from the `-e script` and `-f script_file` options.

The sed utility supports the XBD specification, Utility Syntax Guidelines, except that the order of presentation of the `-e` and `-f` options is significant.

The following options are supported:

-e script: Add the editing commands specified by the script option-argument to the end of the script of editing commands. The script option-argument has the same properties as the script operand, described in the OPERANDS section.

-f script_file: Add the editing commands in the file `script_file` to the end of the script.

-n: Suppress the default output (in which each line, after it is examined for editing, is written to standard output). Only lines explicitly selected for output will be written.

Multiple `-e` and `-f` options may be specified. All commands are added to the script in the order specified, regardless of their origin.

The following operands are supported:

file: A pathname of a file whose contents will be read and edited. If multiple file operands are specified, the named files will be read in the order specified and the concatenation will be edited. If no file operands are specified, the standard input will be used.

script: A string to be used as the script of editing commands. The application must not present a script that violates the restrictions of a text file except that the final character need not be a new-line character.

Stdin: The standard input will be used only if no file operands are specified.

Input Files: The input files must be text files. The `script_files` named by the `-f` option will consist of editing commands, one per line.

Diff Utility: The diff utility is another useful tool that developers may need. It is used to find out the differences between two files. If you are using CVS, differences between different versions of a file in the CVS repository can be found using the `cvs (cvs diff)` command as well. However, if you want to find out the difference between two files that are not in the CVS repository, the diff utility may be quite handy. One common use may be to find out the difference between the working copy and the backup copy of a source code file. This will enable you to find out what changes have been made in the working copy of the file. The output of the diff utility follows similar rules to those used in the CVS diff command. It displays the differences between two files, or each corresponding file in two directories.

Each set of differences is called a “diff” or “patch”. For files that are identical, diff normally produces no output; for binary (non-text) files, diff normally reports only that they are different.

Syntax is `diff [options] from-file to-file`



Caselet

File Sharing or Stealing?

The files in question contain merely the information needed to download film or music files, and nothing more. This information is, however, lifted from those who had themselves copied the files without permission from the original track.

R.K. Raghavan

There is great interest among millions of Internet users across continents in a trial that is going on in Stockholm. It is over alleged copyright violations committed by Pirate Bay, a popular Swedish Web site floated by an anti-copyright organisation Piratbyran.

Any ruling in this case could affect the fate of many other sites engaged in making available music, movies and software free of cost to the large online community that is ceaselessly on the lookout for free downloads. In these days of high cost of living, who is not tempted by freebies?

Founded in 2003, Pirate Bay started its service independent of Piratbyran in 2004. It has undoubtedly endeared itself to millions across the globe who have a huge stake in the collapse of the prosecution in this case.

According to many observers, Sweden has one of the most lax copyright laws in the world, and it was this that possibly encouraged its three founders, now on trial along with a fourth person, an investor in the site. Interestingly, the criminal case against Pirate Bay is accompanied simultaneously by a civil claim initiated by music and movie companies. The latter alone are demanding a compensation of 10.9 million euros.

Pirate Bay's case is that it does not store any pirated music, movies or other copyrighted material. What it does is merely help its customers, nearly 25 million, track down the files listed by BitTorrent which is basically a data-sharing protocol and helps transfer large files.

As one defender of the site put it: "It is at best a file pointer," meaning, thereby, that it does not store unauthorised files. Also, BitTorrent is a legal application and is extremely popular among file-sharers because it is fast and efficient unlike many others in the trade.

The files in question contain merely the information needed to download film or music files, and nothing more. This information is, however, lifted from those who had themselves copied the files without permission from the original track. There is an element of ethics that cannot be ignored here.

The three Pirate Bay operators claim they have hardly made any money. This, however, needs verification.

According to Swedish prosecutors, who were trying to debunk the site's Robin Hood image, Pirate Bay earned annual revenue of \$4.5 million, solely from advertisements. This squares with the uncorroborated information that it was trying to buy Sealand, a platform in the North Sea, close to the Suffolk coast, in order to locate its servers. When the negotiations failed, it switched on trying to buy an island. If these reports are correct, they point to an affluence that would be difficult for the four men on trial to cover up during the three-week-long proceedings.

Backed by many servers

This is not the first time that Pirate Bay has been hauled up. In 2006, the police swooped on its offices and seized a lorry-load of computers, including many servers.

Contd...

Notes

As a result the site went out of action temporarily. Within days it came back to life with additional infrastructure. As a result, it has now 30 servers all over the world, and the men behind it claim that any one server disabled because of police action will not affect its service elsewhere in the world. (Many legal pundits point out that this is one factor that could help Pirate Bay get away.

How can persons hounding Pirate Bay pursue cases in so many countries that have widely varying copyright and cyber laws?)

Also, the police raid gave huge publicity to Pirate Bay which from then on acquired a celebrity status. It became a kind of a "folk hero" as one observer put it.

Pirate Bay operators are not penitent. Nor are they worried about the outcome of the trial. They are credited with the position that the site would continue whatever be the conclusion.

What should encourage them further in their battle with Swedish law enforcement and the music/film industry is the fact that, on the second day of the trial itself, the Swedish prosecutor dropped one of the charges: copying films and music without authorisation.

The only charge that will now be pursued is that which accuses Pirate Bay of making movies or music protected by IPR accessible to the public. This is an interesting development. Pirate Bays are ecstatic at this.

Apart from upholding Pirate Bay, the prosecution volte face should encourage or at least give ideas to those who offer services similar to the former.

Further proceedings in this trial, described by some as the 'Internet Piracy Trial of the Decade', should be absorbing. This has attracted so many that there is live audio coverage of daily proceedings. This has warmed the hearts of those on trial. In the words of Mark Mulligan of Forrester Research: "After every victory, file sharing has got bigger. I see no reason why the same won't happen again this time."

The question is who is going to win?

The writer is a former CBI Director who is currently Adviser (Security) to TCS Ltd.

Source: <http://www.thehindubusinessline.in/ew/2009/02/23/stories/2009022350040200.htm>

10.5 Summary

In Linux the command line interpreter is known as the shell. Whatever you type at the command line is understood and interpreted by a program and then that program gives you an output after executing your command. bash or Bourne Again shell is the standard GNU shell. The shell associates a job with each pipeline. It keeps a table of currently executing jobs, which may be listed with the jobs command. Bash uses the job abstraction as the basis for job control. This variable controls how the shell interacts with the user and job control. All the programs that run under Linux are called as processes. Processes run continuously in Linux and you can kill or suspend different processes using various commands. Like that legacy operating system, the files on a Linux system are arranged in what is called a hierarchical directory structure. This means that they are organized in a tree-like pattern of directories (called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and subdirectories which contain more files and subdirectories and so on and so on. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. A file system is an organization of data and metadata on a storage device. Most files are just files, called regular files; they contain normal data, for example text files, executable files or programs, input for or output from a program and so on. As a user,

you only need to deal directly with plain files, executable files, directories and links. Linux places all the partitions under the root directory by 'mounting' them under specific directories. Under Linux, unless you mount a partition or a device, the system does not know of the existence of that partition or device. Devices that can contain file systems are known as block devices. All of the details of the Linux file systems are translated by software so that all file systems appear identical to the rest of the Linux kernel and to programs running in the system. Linux's Virtual File system layer allows you to transparently mount the many different file systems at the same time. NFS allows hosts to mount partitions on a remote system and use them as though they are local file systems. NFS only performs authentication when a client system attempts to mount a remote file system. Quota Management allows to limit the amount of disk space and/or the number of files a user can use.

10.6 Keywords

/bin: The bin directory contains several useful commands that are used by both the system administrator as well as non-privileged users.

/boot: This directory contains the system.map file as well as the Linux kernel.

(Domain) Sockets: A special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.

diff utility: It is used to find out the differences between two files.

/etc: This directory contains all the configuration files for your system.

BASH: Its full form is Bourne Again shell and it is the standard GNU shell.

Bashdoc: It is a tool to make documentation from bash scripts.

Data Partition: It is the normal Linux system data, including the root partition containing all the data to start up and run the system.

Directories: Files that are lists of other files.

File system: It is an organization of data and metadata on a storage device.

File: The files in a file system are collections of data.

Indent programme: It is your tool to handle the situation and reformat all files in the project according to the style adopted for the project.

Job Control Variable: This variable controls how the shell interacts with the user and job control.

Job Control: It refers to the ability to selectively stop (suspend) the execution of processes and continue (resume) their execution at a later point.

/lib: This contains all the shared libraries that are required by system programs.

Links: A system to make a file or directory visible in multiple parts of the system's file tree.

Mode: It is a three-item sequence which represents who, operation and permission.

man pages: The man pages are a user manual that is by default built into most Linux distributions (i.e., versions) and most other Unix-like operating systems during installation.

Named Pipes: It act more or less like sockets and form a way for processes to communicate with each other, without using network socket semantics.

Pipeline: It is a sequence of simple commands separated by one of the control operators '|', '&' or '&|'.

Notes

Processes: All the programs that run under Linux are called as processes.

Quota Management: It allows to limit the amount of disk space and/or the number of files a user can use.

Redirection: It simply means capturing output from a file, command, program, script, or even code block within a script and sending it as input to another file, command, program, or script.

root directory: The first directory in the Linux file system is called the root directory.

/sbin: This directory contains all the binaries that are essential to the working of the system.

sed utility: It is a stream editor that can be used for different file editing purposes when used as a filter.

/tmp: This directory contains mostly files that are required temporarily.

Special Files: The mechanism used for input and output. Most special files are in /dev.

Swap Partition: It is the expansion of the computer's physical memory, extra memory on hard disk.

Texinfo: It is a typesetting syntax used for generating documentation in both on-line and printed form (creating filetypes as dvi, html, pdf, etc., and its own hypertext format info) with a single source file.

/var: This directory contains spooling data like mail and also the output from the printer daemon.

10.7 Self Assessment

Fill in the blanks:

1. The first directory in the Linux file system is called the directory.
2. The command changes the ownership of a file or files.
3. A is a three-item sequence which represents who, operation and permission.
4. Linux places all the partitions under the root directory by them under specific directories.
5. directory contains the system.map file as well as the Linux kernel.
6. directory contains all the user binaries.
7. All file systems are mounted onto a directory which is known as the
8. The first file system that Linux had called
9. Full form of VFS is
10. Full form of NFS is
11. NFS consists of at least two main parts: and
12. Full form of NSM is
13. allow you to select filenames based on patterns of characters.
14. The program copies files and directories.
15. is used to find out the differences between two files.

10.8 Review Questions

Notes

1. What is shell? Explain it.
2. Describe Bourne Again shell.
3. What is Job control? Describe job Control Builtins.
4. Compare and contrast between job Control variables and environment variables.
5. What is PS1? How does it work?
6. What do you mean by pipeline? Explain with example.
7. What do you mean by redirection? Explain with example.
8. What is man command? How does it work?
9. Explain the kill command with example.
10. Write short note on the following topics:
 - (a) Bashdoc
 - (b) The > operator
 - (c) Texinfo
 - (d) ls command
 - (e) sed utility
11. What is file system?
12. What is Virtual File system? Explain it.
13. What do you mean by mount point? What is its significance?
14. Explain the concept of Unified filesystem in Linux.
15. What do you mean by Quota Management?
16. Write short notes on the following topics:
 - (a) Directories
 - (b) Data partition
 - (c) Network File System

Answers: Self Assessment

- | | | | |
|--------------------------------------|-------------|----------------------------|-------------|
| 1. root | 2. chown | 3. mode | 4. mounting |
| 5. /boot | 6. /usr/bin | 7. mounted file system | 8. Minix |
| 9. Virtual File System | | 10. Network File System | |
| 11. a server and one or more clients | | 12. Network Status Monitor | |
| 13. Wildcards | 14. Cp | 15. diff utility | |

Notes

10.9 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlan and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

http://tldp.org/LDP/intro-linux/html/sect_03_01.html

<http://tldp.org/LDP/Linux-Filesystem-Hierarchy/Linux-Filesystem-Hierarchy.pdf>

Unit 11: User Accounts

Notes

CONTENTS

Objectives

Introduction

11.1 Superuser

11.1.1 Who is the Super User?

11.1.2 How to Add Users?

11.1.3 How to Change Passwords?

11.1.4 How to Delete Users?

11.1.5 How to Tell the Groups to which a user Belongs?

11.1.6 How to Change the Ownership of a File?

11.1.7 Using sudo

11.1.8 User and Super User

11.2 Normal Users in Linux

11.3 Creating Groups

11.4 Summary

11.5 Keywords

11.6 Self Assessment

11.7 Review Questions

11.8 Further Readings

Objectives

After studying this unit, you will be able to:


- Discuss differences between super user and normal user
- Discuss RedHat user manager
- Understand creating groups

Introduction

Most of us need to create users accounts once in a while. System administrators frequently work with user accounts on a daily basis. The following text wants to explain how to work with (create, modify) user accounts and groups in a Linux environment.

Most of the people never bothered to understand this or they run into complicated documentation that doesn't clarify things. I will try to make this as comprehensive as possible, but in the end, you'll be the one to judge.

Notes

 <i>Note</i>	Most of you use the root account for day to day tasks but this is wrong.
--	--

The easiest way to add a new user is to use the useradd command like this: useradd bart. We have now created a new user called bart. All basic requirements are met at this point. We would probably want to assign a password for that user and this is done with the command passwd bart.

Everything starts with the /etc/passwd file. Here, you can see all the accounts that exist on your Linux system and for each one, there are several fields that describe various stuff. By default, your /etc/passwd file has several entries that are actually users for programs that need to control processes or need "special" access to the filesystem. You'll also see there the root account and perhaps some user accounts that were created at installation time or after that.

11.1 Superuser

In Linux and Unix like computer operating systems, root is the conventional name of the user who has all rights or permissions (to all files and programs) in all modes (single- or multi-user).

The etymology of the term may be that root is the only user account with permission to modify the root directory of a Unix system.

Superuser login - How to become Superuser in Linux

Under Linux (and other Unixish oses) you use command called su. It is used is used to become another user during a login session or to login as super user. If Invoked without a username, su defaults to becoming the super user. It is highly recommend that you use argument - to su command. It is used to provide an environment similar to what the user root would expect had the user logged in directly. Type su command as follows:
\$ su -Output:

Password: <TYPE ROOT PASSWORD>

#

Once you typed the root user password, you become super or root user.

All users on a Unix system are the same, except one. That user is called the **SuperUser** and has complete control of the system. This user can kill any of the running processes and access all of the files (including the device files, so root is the one who configures the hardware), as opposed to ordinary users who can only mess with their own processes and files. Many other **Operating Systems** have a similar notion of one or more super-privileged accounts. On Unix the **SuperUser** is called root (think of a tree here), because this account alone has sufficient permissions to manipulate the files and directories involved in the creation of new user accounts. (You can change that of course, but you really don't want to.) If you are logged in as a regular user and you want to "become root", you use the command (which stands for switch user and can change you to any user).

Before we proceed, it would be best to cover some basic user administration topics that will be very useful in later unit.

Adding Users

Notes

One of the most important activities in administering a Linux box is the addition of users. Here you'll find some simple examples to provide a foundation for future unit. It is not intended to be comprehensive, but is a good memory refresher. You can use the command `man useradd` to get the help pages on adding users with the `useradd` command or the `man usermod` to become more familiar with modifying users with the `usermod` command.



Did u know? What are permissions to the root users?

11.1.1 Who is the Super User?

The super user with unrestricted access to all system resources and files in Linux is the user named `root`. This user has a user ID, of 0 which is universally identified by Linux applications as belonging to a user with supreme privileges. You will need to log in as user `root` to add new users to your Linux server.



Note When installing Ubuntu Linux systems, you are prompted to create a primary user that is not `root`. A `root` user is created but no password is set, so you initially cannot log in as this user.

11.1.2 How to Add Users?

Adding users takes some planning; read through these steps below before starting:

1. Arrange your list of users into groups by function. In this example there are three groups "parents", "children" and "soho".

Parents	Children	Soho
Paul	Alice	Accounts
Jane	Derek	Sales

2. Add the Linux groups to your server:

```
[root@bigboy tmp]# groupadd parents
[root@bigboy tmp]# groupadd children
[root@bigboy tmp]# groupadd soho
```

3. Add the Linux users and assign them to their respective groups

```
[root@bigboy tmp]# useradd -g parents paul
[root@bigboy tmp]# useradd -g parents jane
[root@bigboy tmp]# useradd -g children derek
[root@bigboy tmp]# useradd -g children alice
[root@bigboy tmp]# useradd -g soho accounts
[root@bigboy tmp]# useradd -g soho sales
```


If you don't specify the group with the `-g`, RedHat/Fedora Linux creates a group with the same name as the user you just created; this is also known as the User Private Group Scheme. When each new user first logs in, they are prompted for their new permanent password.

Notes

- Each user's personal directory is placed in the /home directory. The directory name will be the same as their user name.

```
[root@bigboy tmp]# ll /home
drwxr-xr-x  2 root    root      12288 Jul 24 20:04 lost+found
drwx----- 2 accounts soho      1024 Jul 24 20:33 accounts
drwx----- 2 alice   children 1024 Jul 24 20:33 alice
drwx----- 2 derek   children 1024 Jul 24 20:33 derek
drwx----- 2 jane    parents  1024 Jul 24 20:33 jane
drwx----- 2 paul    parents  1024 Jul 24 20:33 paul
drwx----- 2 sales   soho     1024 Jul 24 20:33 sales

[root@bigboy tmp]#
```



Task Explain the procedure of adding the Linux users and assigning them to their respective groups.

11.1.3 How to Change Passwords?

You need to create passwords for each account. This is done with the passwd command. You are prompted once for your old password and twice for the new one.

- User root changing the password for user paul.

```
[root@bigboy root]# passwd paul
Changing password for user paul.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@bigboy root]#
```

- Users might wish to change their passwords at a future date. Here is how unprivileged user paul would change his own password.

```
[paul@bigboy paul]$ passwd
Changing password for paul
Old password: your current password
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password: your new password
Re-enter new password: your new password
Password changed.
[paul@bigboy paul]$
```

11.1.4 How to Delete Users?

Notes

The `userdel` command is used to remove the user's record from the `/etc/passwd` and `/etc/shadow` used in the login process. The command has a single argument, the username.

```
[root@bigboy tmp]# userdel paul
```

There is also an optional `-r` switch that additionally removes all the contents of the user's home directory. Use this option with care. The data in a user's directory can often be important even after the person has left your company.

```
[root@bigboy tmp]# userdel -r paul
```

11.1.5 How to tell the Groups to which a user Belongs?

Use the `groups` command with the username as the argument.

```
[root@bigboy root]# groups paul
```

```
paul : parents
```

```
[root@bigboy root]#
```

11.1.6 How to Change the Ownership of a File?

You can change the ownership of a file with the `chown` command. The first argument is the desired username and group ownership for the file separated by a colon (`:`) followed by the filename. In the next example we change the ownership of the file named `test.txt` from being owned by user `root` and group `root` to being owned by user `testuser` in the group `users`:

```
[root@bigboy tmp]# ll test.txt
```

```
-rw-r--r-- 1 root root 0 Nov 17 22:14 test.txt
```

```
[root@bigboy tmp]# chown testuser:users test.txt
```

```
[root@bigboy tmp]# ll test.txt
```

```
-rw-r--r-- 1 testuser users 0 Nov 17 22:14 test.txt
```

```
[root@bigboy tmp]#
```



Caution

You can also use the `chown` command with the `-r` switch for it to do recursive searches down into directories to change permissions.

11.1.7 Using `sudo`

If a server needs to be administered by a number of people it is normally not a good idea for them all to use the root account. This is because it becomes difficult to determine exactly who did what, when and where if everyone logs in with the same credentials. The `sudo` utility was designed to overcome this difficulty.

The `sudo` utility allows users defined in the `/etc/sudoers` configuration file to have temporary access to run commands they would not normally be able to due to file permission restrictions. The commands can be run as user "root" or as any other user defined in the `/etc/sudoers` configuration file. The privileged command you want to run must first begin with the word `sudo` followed by the command's regular syntax.

When running the command with the `sudo` prefix, you will be prompted for your regular password before it is executed. You may run other privileged commands using `sudo` within a

Notes

five-minute period without being re-prompted for a password. All commands run as sudo are logged in the log file /var/log/messages.



Did u know? How can we run the privileged command?



Example: Using sudo is relatively simple as we can see from these examples.

Temporarily Gaining root Privileges

In this example, user bob attempts to view the contents of the /etc/sudoers file, which is an action that normally requires privileged access. Without sudo, the command fails:

```
[bob@bigboy bob]$ more /etc/sudoers
/etc/sudoers: Permission denied

[bob@bigboy bob]$
```

Bob tries again using sudo and his regular user password and is successful:

```
[bob@bigboy bob]$ sudo more /etc/sudoers
Password:
...
...

[bob@bigboy bob]$
```

The details of configuring and installing sudo are covered in later sections.

Becoming Root for a Complete Login Session

The su command allows a regular user to become the system's root user if they know the root password. A user with sudo rights to use the su command can become root, but they only need to know their own password, not that of root as seen here.

```
someuser@u-bigboy:~$ sudo su -
Password:
root@u-bigboy:~#
```

Some systems administrators will use sudo to grant root privileges to their own personal user account without the need to provide a password.

Later sections describe how to disable sudo su ability and also how to use sudo without password prompts.

Downloading and Installing the sudo Package

Fortunately the package is installed by default by RedHat/Fedora which eliminates the need to anything more in this regard.

The visudo Command

The visudo command is a text editor that mimics the vi editor that is used to edit the /etc/sudoers configuration file. visudo uses the same commands as the vi text editor. The visudo command must run as user root and should have no arguments:

```
[root@aqua tmp]# visudo
```



Task

“It is not recommended that you use any other editor to modify your sudo parameters because the sudoers file isn’t located in the same directory on all versions of Linux.” Comment

The /etc/sudoers File

The /etc/sudoers file contains all the configuration and permission parameters needed for sudo to work. There are a number of guidelines that need to be followed when editing it with visudo. General /etc/sudoers Guidelines

There are some general guidelines when editing this file:

1. Groups are the same as user groups and are differentiated from regular users by a % at the beginning. The Linux user group “users” would be represented by %users.
2. You can have multiple usernames per line separated by commas.
3. Multiple commands also can be separated by commas. Spaces are considered part of the command.
4. The keyword ALL can mean all usernames, groups, commands and servers.
5. If you run out of space on a line, you can end it with a back slash (\) and continue on the next line.
6. sudo assumes that the sudoers file will be used network wide, and therefore offers the option to specify the names of servers which will be using it in the servername position in. In most cases, the file is used by only one server and the keyword ALL suffices for the server name.
7. The NOPASSWD keyword provides access without prompting for your password.



Example: This section presents some simple examples of how to do many commonly required tasks using the sudo utility.

Granting all Access to Specific Users

You can grant users bob and bunny full access to all privileged commands, with this sudoers entry.

```
bob, bunny ALL=(ALL) ALL
```

This is generally not a good idea because this allows bob and bunny to use the su command to grant themselves permanent root privileges thereby bypassing the command logging features of sudo. The example on using aliases in the sudoers file shows how to eliminate this prob

Granting Access to Specific Users to Specific Files

This entry allows user peter and all the members of the group operator to gain access to all the program files in the /sbin and /usr/sbin directories, plus the privilege of running the command /usr/local/apps/check.pl. Notice how the trailing slash (/) is required to specify a directory location:

```
peter, %operator ALL= /sbin/, /usr/sbin, /usr/local/apps/check.pl
```

Notes

Notice also that the lack of any username entries within parentheses () after the = sign prevents the users from running the commands automatically masquerading as another user. This is explained further in the next example.

Granting Access to Specific Files as Another User

The sudo -u entry allows you to execute a command as if you were another user, but first you have to be granted this privilege in the sudoers file.

This feature can be convenient for programmers who sometimes need to kill processes related to projects they are working on. For example, programmer peter is on the team developing a financial package that runs a program called monthend as user accounts. From time to time the application fails, requiring “peter” to stop it with the /bin/kill, /usr/bin/kill or /usr/bin/pkill commands but only as user “accounts”. The sudoers entry would look like this:

```
peter ALL=(accounts) /bin/kill, /usr/bin/kill, /usr/bin/pkill
```

User peter is allowed to stop the monthend process with this command:

```
[peter@bigboy peter]# sudo -u accounts pkill monthend
```

Granting Access without Needing Passwords

This example allows all users in the group operator to execute all the commands in the /sbin directory without the need for entering a password. This has the added advantage of being more convenient to the user:

```
%operator ALL= NOPASSWD: /sbin/
```

Using Aliases in the sudoers File

Sometimes you’ll need to assign random groupings of users from various departments very similar sets of privileges. The sudoers file allows users to be grouped according to function with the group and then being assigned a nickname or alias which is used throughout the rest of the file. Groupings of commands can also be assigned aliases too.

In the next example, users peter, bob and bunny and all the users in the operator group are made part of the user alias ADMINS. All the command shell programs are then assigned to the command alias SHELLS. Users ADMINS are then denied the option of running any SHELLS commands and su:

```
Cmd_Alias    SHELLS = /usr/bin/sh, /usr/bin/csh, \  
              /usr/bin/ksh, /usr/local/bin/tcsh, \  
              /usr/bin/rsh, /usr/local/bin/zsh  
  
User_Alias   ADMINS = peter, bob, bunny, %operator  
  
ADMINS      ALL    = !/usr/bin/su, !SHELLS
```

This attempts to ensure that users don’t permanently su to become root, or enter command shells that bypass sudo’s command logging. It doesn’t prevent them from copying the files to other locations to be run. The advantage of this is that it helps to create an audit trail, but the restrictions can be enforced only as part of the company’s overall security policy.

Other Examples

Notes

You can view a comprehensive list of `/etc/sudoers` file options by issuing the command `man sudoers`.

Using syslog to Track all sudo Commands

All sudo commands are logged in the log file `/var/log/messages` which can be very helpful in determining how user error may have contributed to a problem. All the sudo log entries have the word `sudo` in them, so you can easily get a thread of commands used by using the `grep` command to selectively filter the output accordingly.

Here is sample output from a user `bob` failing to enter their correct sudo password when issuing a command, immediately followed by the successful execution of the command `/bin/more` sudoers.

```
[root@bigboy tmp]# grep sudo /var/log/messages
Nov 18 22:50:30 bigboy sudo(pam_unix)[26812]: authentication failure; logname=bob
uid=0 euid=0 tty=pts/0 ruser= rhost= user=bob
Nov 18 22:51:25 bigboy sudo: bob : TTY=pts/0 ; PWD=/etc ; USER=root ; COMMAND=/
bin/more sudoers
[root@bigboy tmp]#
```



Note

It is important to know how to add users, not just so they can log in to our system. Most server based applications usually run via a dedicated unprivileged user account, for example the MySQL database application runs as user `mysql` and the Apache Web server application runs as user `apache`. These accounts aren't always created automatically, especially if the software is installed using TAR files.

Finally, the `sudo` utility provides a means of dispersing the responsibility of systems management to multiple users. You can even give some groups of users only partial access to privileged commands depending on their roles in the organization. This makes `sudo` a valuable part of any company's server administration and security policy.

11.1.8 User and Super User

Linux puts a lot of power at your fingertips. That's the best reason to switch to Linux; it's also the most dangerous thing about the system. Linux controls how much power you can use on the computer based on your Login ID. It keeps a database of all users, and it keeps track of which user owns which files, and which users have permission to view, edit, and execute each file, folder or program. An ordinary user will not be able to do really dangerous things, like editing the user database, or deleting every file on the system.

But right now you are logged in as `root`. You are not just an ordinary user, you are `SuperUser`. (`SuperUser` is a real Unix term, synonymous with `root`.) There are no restrictions on your power. You have the ability to crash the system and make it otherwise unusable in more ways than you can imagine. As a novice it is very easy to make your system completely unusable with a single erroneous command. Believe me. I speak from experience. The first weekend after I installed Linux, I had to reinstall it FOUR times before I finally got smart and quit destroying it. And I'm a pretty savvy guy around computers, so don't think you're immune just because you know your way around a PC.

Notes

Because it is so dangerous to be logged in as root, you should never use this account unless you have to. The root account is meant to be used by the System Administrator to perform certain duties which can be destructive and therefore should only be performed by an expert. Some examples are emptying log files, mounting and unmounting file systems (more on this later under Getting to CD's and Floppies), installing or removing programs, and creating or deleting user accounts.

If you are using Mandrake Linux, you will have a tool available to perform the most common administration tasks, even when logged in as a regular user. This is called the Mandrake Control Center, which you may find on your desktop or in the Configuration menu. It will ask you for the root password when you start it for security reasons. As a result of this handy tool, you may never need to actually log in as root.

Becoming SuperUser


No phone booth needed. The obvious way is to login as root. That's the best way to do it if you plan on doing a bunch of system maintenance type stuff, but it can be a pain if you're logged in as User with an X session and 14 programs open and connected to the Internet and you just need to copy *one* file into /usr/lib so you can run this program you just downloaded. Fear not, there is a better way. Type this:

```
[user]$ su
Password: *****
[root]#
```

Bang! Just like that, you are SuperUser! A few cautions: Although you are now SuperUser, this is not a "login" shell, so your environment hasn't changed. The biggest way this will effect you is that some programs you normally run as root may appear to be missing. That's because your PATH environment variable, the list of places Linux looks for executables, does not contain /sbin or /usr/sbin. If you try to run a command like shutdown (see below) and it complains, try typing /sbin/shutdown instead. That should do it.

When you are finished with your maintenance tasks you should immediately change back to normal user mode:

```
[root]# exit
[user]$
```

	<p><i>Note</i> While you are SuperUser, your command prompt looks different. An ordinary user is prompted with the dollar sign (\$) while SuperUser gets a pound sign (#).</p>
---	---

This makes it easy to tell which mode you are in. (This is true on most distributions, but the prompts may be different on different distributions, and they can be customized.)

11.2 Normal Users in Linux

A "root" user has the power to do anything on a Linux system. A "normal user" will be assigned some of these capabilities based on what group they are in, but is generally prevented from running any commands that may affect the system outside of their home directory. Users can be granted "sudo" rights, which allows them to run the sudo command and temporarily be granted root privileges.

You need to use the sudo command which is used to execute a command as another user. It allows a permitted user to execute a command as the superuser or another user, as specified in the /etc/sudoers (config file that defines or list of who can run what) file. The sudo command allows users to do tasks on a Linux system as another user.

sudo command

sudo is more secure than su command. By default it logs sudo usage, command and arguments in /var/log/secure (Red Hat/Fedora / CentOS Linux) or /var/log/auth.log (Ubuntu / Debian Linux).

If the invoking user is root or if the target user is the same as the invoking user, no password is required. Otherwise, sudo requires that users authenticate themselves with a password by default. Once a user has been authenticated, a timestamp is updated and the user may then use sudo *without a password for a short period of time (15 minutes unless overridden in sudoers)*.

/etc/sudoers Syntax

Following is general syntax used by /etc/sudoers file:

```
USER HOSTNAME=COMMAND
```

Where,

1. **USER:** Name of normal user
2. **HOSTNAME:** Where command is allowed to run. It is the hostname of the system where this rule applies. sudo is designed so you can use one sudoers file on all of your systems. This space allows you to set per-host rules.
3. **COMMAND:** A simple filename allows the user to run the command with any arguments he/she wishes. However, you may also specify command line arguments (including wildcards). Alternately, you can specify "" to indicate that the command may only be run without command line arguments.

How do I use sudo?

Give user rokcy access to halt/shutdown command and restart Apache web server. First, Login as root user. Use visudo command edit the config file:

```
# visudo
```

Append the following lines to file:

```
rokcy localhost=/sbin/halt
rokcy dbserver=/etc/init.d/apache-perl restart
```

Save and close file . Now rokcy user can restart Apache web server by typing the following command:

```
$ sudo /etc/init.d/apache-perl restart
```

Output:

```
Password:
Restarting apache-perl 1.3 web server....
```

The sudo command has logged the attempt to the logfile /var/log/secure or /var/log/auth.log file:

```
# tail -f /var/log/auth.log
```

Notes

If rokcy want to shutdown computer he needs to type command:

```
$ sudo /sbin/halt
```

Output:

Password:

Before running a command with sudo, users usually supply their password. Once authenticated, and if the /etc/sudoers configuration file permits the user access, then the command is run. sudo logs each command run.

Examples

1. Allow jadmin to run various commands:

```
jadmin ALL=/sbin/halt, /bin/kill, /etc/init.d/httpd
```

2. Allow user jadmin to run /sbin/halt without any password i.e. as root without authenticating himself:

```
jadmin ALL= NOPASSWD: /sbin/halt
```

3. Allow user charvi to run any command from /usr/bin directory on the system dev02:

```
charvi dev02 = /usr/bin/*
```

How to let normal users shut down the computer in Linux?

Spencer Stirling

This may seem like a stupid issue - why SHOULDN'T the average user be able to turn off his/her own machine?

The answer is that Linux is inherently designed as a multiuser system. It would generally be a bad thing if any one user would be allowed to spontaneously turn off the computer while other users are working. Just imagine what hell would ensue if your webserver were taken down by some insignificant user halfway across the world.

Of course, it does seem a little overprotective if the user is physically *sitting* at the machine, since he/she could just reach over and hit the power button (don't do that!!!).

There are several schools of thought concerning how to allow a user to shut down a machine properly. The first method seems somewhat useless to me, but I put it here for completeness.

shutdown.allow

There is a file in /etc called **shutdown.allow** (and if there isn't, root can add it). This file contains a list of users (1 per line) who are allowed to shutdown the computer.

This *doesn't* mean that these users can invoke the **shutdown** (or **reboot** or **halt**) command(s). Instead, it means that an authorized user can shut down the computer by pressing ctrl+alt+del.

In order for this to occur, the ctrl+alt+del key sequence must be trapped in the /etc/inittab file. The necessary line in inittab is

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

When ctrl+alt+del is pressed, **init** checks if there is an authorized user (listed in /etc/shutdown.allow) logged into *any* virtual console, and proceeds to shutdown if this is true.

Of course, if I'm running X windows, then generally the window manager will trap ctrl+alt+del for itself, so this won't work anymore.

sudo

Notes

The program **sudo** allows normal users to execute certain root-only commands. Which users are authorized to run which commands is specified in the `/etc/sudoers` file. This should only be edited with the command **visudo**.

For example, suppose I wanted to add a group of users who are allowed to shut down the machine. So I first want to add a group called "shutdown" (run these commands while root)

```
groupadd shutdown
```

Then I need to edit the `/etc/group` file to add users to the "shutdown" group. I just tack the usernames at the end of the shutdown line, separated by commas, e.g.

```
shutdown:x:407:user1,user2,...
```

Whatever users I put there will be able to shut down the computer (so choose wisely). Now I need to configure sudo to allow members of the "shutdown" group to actually invoke the assorted shutdown commands provided in linux. Run **visudo** and add the following lines

```
%shutdown ALL=(root) NOPASSWD: /sbin/reboot
```

```
%shutdown ALL=(root) NOPASSWD: /sbin/halt
```

```
%shutdown ALL=(root) NOPASSWD: /sbin/shutdown
```

This allows the "shutdown" group to run `/sbin/reboot`, `/sbin/halt`, and `/sbin/shutdown` AS IF THEY WERE ROOT. The only caveat is that the users must run the commands with the command **sudo** in front, e.g.

```
sudo /sbin/halt
```

This is always a bit of a pain (and users never remember), so I can create the following script called `/usr/bin/reboot` (and similar scripts for **halt** and **shutdown**)

```
#!/bin/sh
```

```
sudo /sbin/reboot $*
```

Remember to make these scripts executable! To make this slightly more secure, I might want to change the ownership of these scripts to the "shutdown" group

```
chgrp shutdown /usr/bin/reboot /usr/bin/halt /usr/bin/shutdown
```

and then make them executable *only* for the group "shutdown"

```
chmod g+x /usr/bin/reboot /usr/bin/halt /usr/bin/shutdown
```

KDE shutdown

If you are running **kdm** (the kde display manager - e.g. graphical login) then the shutdown behavior can be modified in "Control Panel" (in "Administrator Mode" of course). If you are starting KDE manually (using **startx** or **startkde**) then you will have to resort to the previous "sudo" solution (and probably add a shortcut on the desktop to one of those scripts).

XFCE4 shutdown

If you are using **xfce4** then you will need to configure a few items in order to use the built-in "Reboot computer"/"Turn off computer" options available when you are logging out.

Notes

CHANGE - SEE BELOW FOR XFCE 4.2

XFCE 4.0

First, all users who are allowed to shut down the machine from **xfce4** must be listed in the /etc/xfce4/shutdown.allow file (syntax is just like in the /etc/shutdown.allow file - see above). Second, the file permissions on the **xfce4-shutdown** program must be modified. This file is usually in /usr/sbin/xfce4-shutdown, so I would type

```
chmod u+s /usr/sbin/xfce4-shutdown
```

the "u+s" argument means that the command **xfce4-shutdown** will run as though the owner (probably root) initiated it, regardless of which user actually called the program.

This should be enough to shut down the computer from xfce4.

Instructions for XFCE 4.2 The game has changed for xfce 4.2. Now you must instead allow sudo access to a program called /usr/sbin/xfsm-shutdown-helper (note: this may also be located in /usr/local/libexec/ - just use the "locate" command to find xfsm-shutdown-helper).

Using the same kind of ideas presented above in the sudo method of shutdown, I add the following line to /etc/sudoers file (using **visudo**)

```
%shutdown ALL=(root) NOPASSWD: /usr/sbin/xfsm-shutdown-helper
```

This allows the "shutdown" group to shutdown the machine.

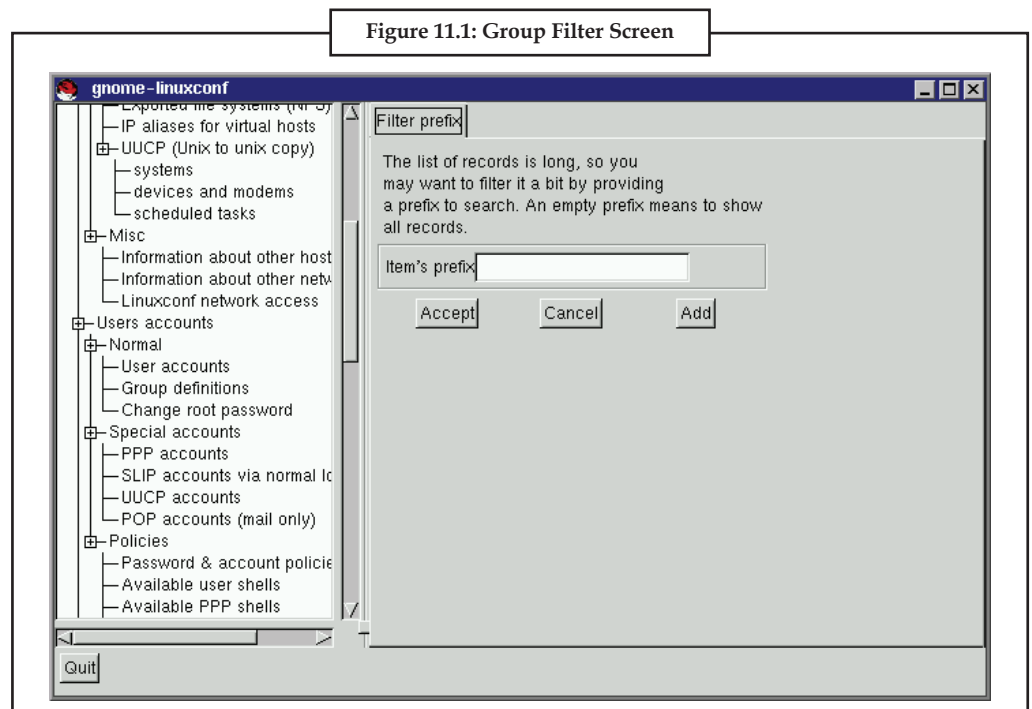
11.3 Creating Groups

To create a new group:

Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [Group definition].

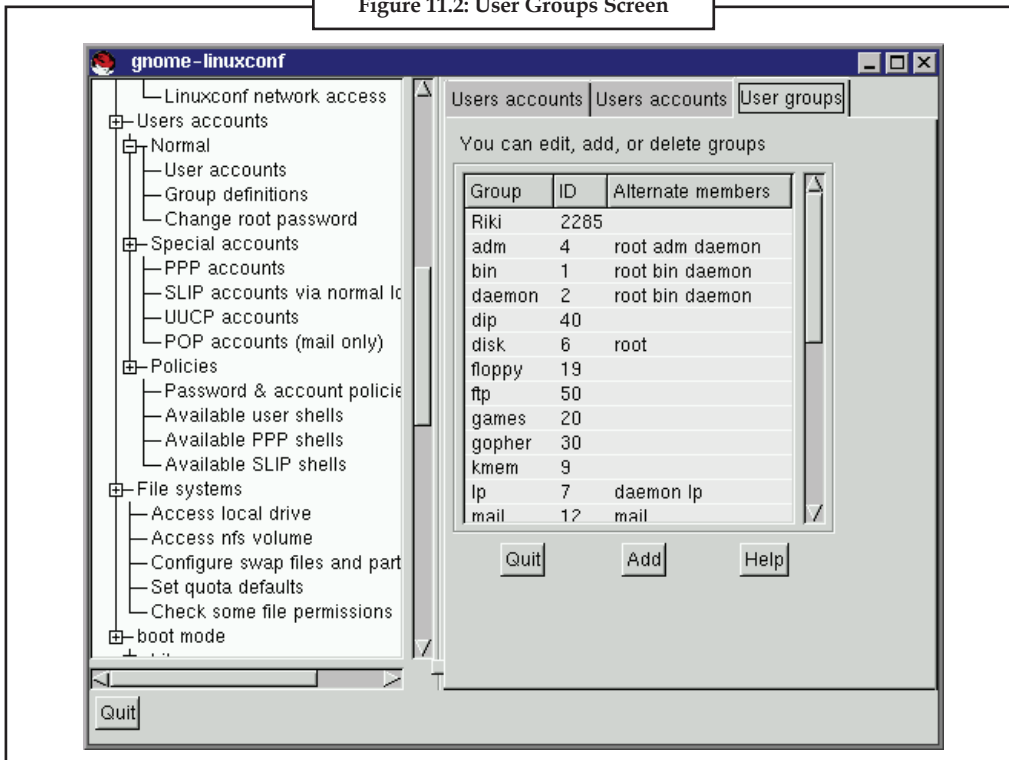
If you have more than 15 groups, you will be given the option to select the groups by providing a prefix.



You may add a group directly from this screen, or move on to the User groups screen. To move on select choice Accept with or without a prefix, to add a new group, hit choice Add.

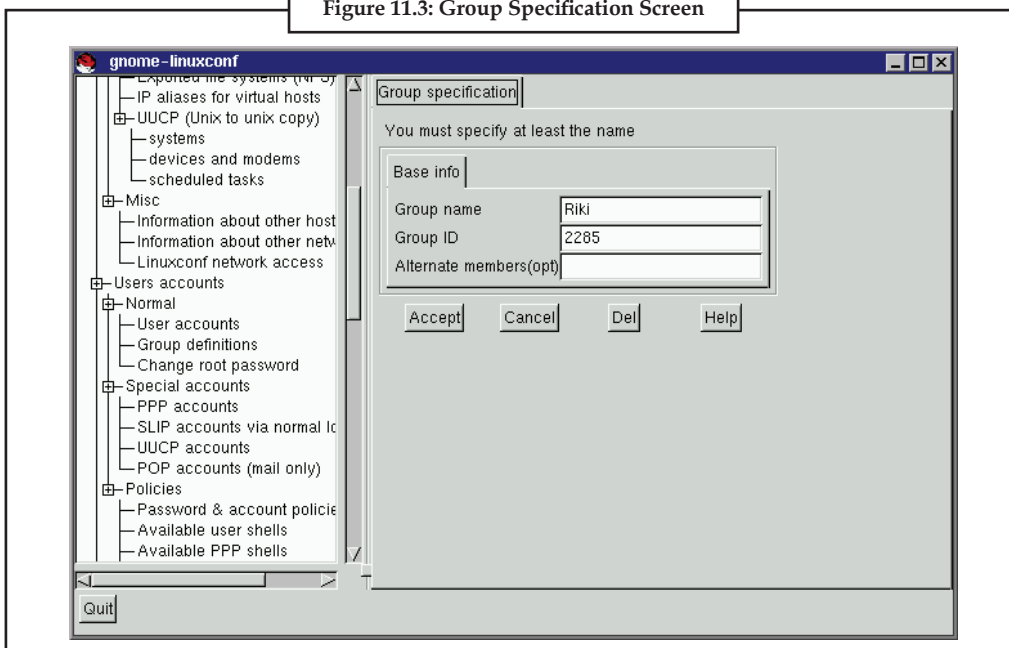
Notes

Figure 11.2: User Groups Screen



Select Add at the bottom of the User groups screen.

Figure 11.3: Group Specification Screen



Notes

Enter a group name. You may also wish to specify members of the group and can do so in the Alternate members field. The list of users should be space delimited, meaning that each username must have a space between it and the next one. When you're finished, select Accept and the group will be created.

Deleting a Group

To delete a group:

Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [Group definitions].

If you have more than 15 groups, you will be given a filter screen to narrow your choice of groups by specifying a prefix.

With or without a prefix select Accept at the bottom of the screen.

On the User groups screen select the group you wish to delete.

You'll be presented with the Group specification screen.

Select Del to delete the group. Linuxconf will then prompt you to confirm the deletion. Choose yes to delete the group.

The group's files will still remain and their respective owners will still have sole control over them. The group name will be replaced with the deleted group's ID. The files may be assigned to a new group by using the chgrp command. More information on chgrp can be found by typing the command info chgrp or man chgrp at the shell prompt. If a new group is created and the deleted group's ID is specified then the new group will have access to the deleted group's files. Don't worry, Linuxconf doesn't recycle old group numbers any more than it does old user IDs, so it won't happen by accident.

Modifying Group Membership

There are two ways to modify the list of users that belong to a group. You can either update each user account itself, or you can update the group definitions. In general, the fastest way is to update each of the group definitions. If you're planning on changing more information for each user than just the group information, then updating each user account may prove easier.

We'll start by detailing the group definitions method.

Start Linuxconf by typing linuxconf at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [Group definitions].

If you have more than 15 groups, you will be given a filter screen to narrow your choice of groups by specifying a prefix.

With or without a prefix, select Accept at the bottom of the screen.

Select the group you wish to modify. This will open the Group specification screen.

Add or remove each user from the Alternate members field. Make sure that all of the user names are separated by a space " " character.

Once you've done this select Accept which can be found at the bottom of the screen.

This will automatically update each user account with the group showing up in the Supplementary groups field if added or absent if removed.

Adding and removing groups can also be done by modifying each individual user account.

Notes

Start Linuxconf by typing `linuxconf` at the shell prompt.

Open [Config] → [Users accounts] → [Normal] → [User accounts].

If you have more than 15 accounts on the system, Linuxconf will provide you with a filter screen.

On the User accounts screen, select a user that you wish to update. You will be presented with the User information screen.

Add or remove the desired groups from the, Supplementary groups field. Each group should be separated by a space " " character.

Once you've made all the changes you'd like, select Accept at the bottom of the screen.

This will automatically update the group definitions. Repeat the process for each user.

Setuid and Setgid Programs

`setuid` and `setgid` (short for set user ID upon execution and set group ID upon execution, respectively) are Linux access rights flags that allow users to run an executable with the permissions of the executable's owner or group. They are often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task. While the assumed user id or group id privileges provided are not always elevated, at a minimum they are specific.

`setuid` and `setgid` are needed for tasks that require higher privileges than those which a common user has, such as changing his or her login password. Some of the tasks that require elevated privileges may not immediately be obvious, though — such as the ping command, which must send and listen for control packets on a network interface.

setgid()

`setgid` sets the effective group ID of the current process. If the caller is the superuser, the real and saved group ID's are also set. If the process has appropriate privileges, `setgid()` shall set the real group ID, effective group ID, and the saved set-group-ID of the calling process to `gid`.

If the process does not have appropriate privileges, but `gid` is equal to the real group ID or the saved set-group-ID, `setgid()` shall set the effective group ID to `gid`; the real group ID and saved set-group-ID shall remain unchanged.

The `setgid()` function shall not affect the supplementary group list in any way.

Any supplementary group IDs of the calling process shall remain unchanged.

Upon successful completion, 0 is returned. Otherwise, -1 shall be returned and `errno` set to indicate the error.

setuid()

If the process has appropriate privileges, `setuid()` shall set the real user ID, effective user ID, and the saved set-user-ID of the calling process to `uid`.

If the process does not have appropriate privileges, but `uid` is equal to the real user ID or the saved set-user-ID, `setuid()` shall set the effective user ID to `uid`; the real user ID and saved set-user-ID shall remain unchanged.

Notes

The setuid() function shall not affect the supplementary group list in any way.

Upon successful completion, 0 shall be returned. Otherwise, -1 shall be returned and errno set to indicate the error.



Caselet

Mobile-based e-payment can Facilitate Microfinance, says Expert

Vinson Kurian

Thiruvananthapuram, Nov. 1

Mobile telephony will be the best option to drive financial inclusion, at a time when the ills plaguing the micro finance sector threaten to turn off the tap of formal finance to the poor.

Mr Abraham Paul, a Kerala-based communications and networking expert, has created a 'Money-through-Mobile (MTM)' facility, which, he says, is an easy tool to transact virtually all form of e-payments.

This could range from utility bills, taxi/auto fares, road and bridge toll; shops and malls and payments at petrol bunks, cinema halls, shops and malls, to wages for employees and workers.

Government agencies can send welfare payments to the MTM system through banks to credit the beneficiary's MTM account, Mr Paul, a Siemens and BPL mobile veteran, with 50 years of experience in the industry, says.

Set-top Box

The MTM system can be accessed by mobile phone users of different service providers over the mobile phone network, he says.

A special purpose set top box (STB) equipped with SIMs of various mobile service providers will be placed at various vendors and service provider premises, where multiple accesses to mobile network is required to take care of the call traffic.

The STB will interface a local computer with the MTM application to enable interaction with the system via the mobile network. Applications required in the user devices can be downloaded from the system.

Mobile users can register themselves as MTM clients by making an initial deposit in their MTM account, through the local post office or at any MTM franchisee. The user can link his post office savings bank account with his or her MTM account.

KYC Information

Unique identity numbers, when ready, offer themselves as a tool for know-your-customer (KYC) information as well as for personal identification.

An MTM user can pay money into any other user account. However, withdrawal and e-transfer of money can be transacted only from his own account.

Prepaid recharge coupons marketed by the system provider could be used to recharge/refill the MTM account, as in the case of prepaid mobile phone service.

Contd...

Cash vending machines

Another powerful feature is e-transfer of money using the prepaid coupons by any land line or mobile phone, from anywhere to any client account.

Cash vending machines, identical to ATMs but featuring an STB, and provided by the system provider at franchisee points will deliver real cash to MTM users.

The system technology is identical to the pre-paid system and uses the mobile network and switching systems for access, but its function is transparent to the latter. The whole idea is to make the micro-payment banking facility available, accessible and affordable for people at the lower end of the economic strata across the country.

Excess Liquidity

The proposed system will be an exclusive technology-oriented mobile banking system that is transparent to mobile service providers, banks or other financial architecture owned by the Government.

Since MTM deals amount to being virtual e-transactions, most of the real money remains within the MTM system.

The huge excess liquidity available with MTM owner can be profitably deployed in various banking products, subject to regulatory norms.

Notes

Source: <http://www.thehindubusinessline.in/2010/11/02/stories/2010110252550700.htm>

11.4 Summary

Under Linux (and other Unixish oses) you use command called su. It is used is used to become another user during a login session or to login as super user. One of the most important activities in administering a Linux box is the addition of users. Here you'll find some simple examples to provide a foundation for future chapters. It is not intended to be comprehensive, but is a good memory refresher. The sudo utility allows users defined in the /etc/sudoers configuration file to have temporary access to run commands they would not normally be able to due to file permission restrictions. Sometimes you'll need to assign random groupings of users from various departments very similar sets of privileges.

11.5 Keywords

Normal User: A "normal user" will be assigned some of these capabilities based on what group they are in, but is generally prevented from running any commands that may affect the system outside of their home directory.

Super User: The super user with unrestricted access to all system resources and files in Linux is the user named root.

Setuid and Setgid: setuid and setgid are Linux access rights flags that allow users to run an executable with the permissions of the executable's owner or group.

11.6 Self Assessment

Fill in the blanks:

1. One of the most important activities in a Linux box is the addition of users.
2. When each new user first logs in, they are prompted for their new password.
3. The command is used to remove the user's record from the /etc/passwd and /etc/shadow used in the login process.

Notes

4. You can change the ownership of a file with the command.
5. If a server needs to be administered by a number of people it is not a good idea for them all to use the root account.
6. All commands run as are logged in the log file /var/log/messages.
7. The command allows a regular user to become the system's root user if they know the root password.
8. The command is a text editor that mimics the vi editor that is used to edit the /etc/sudoers configuration file.
9. The file contains all the configuration and permission parameters needed for sudo to work.
10. The sudo -u entry allows you to a command as if you were another user.
11. All sudo commands are in the log file /var/log/messages which can be very helpful in determining how user error may have contributed to a problem.
12. The account is meant to be used by the System Administrator to perform certain duties which can be destructive and therefore should only be performed by an expert.
13. Before a command with sudo, users usually supply their password.
14. A "root" user has the to do anything on a Linux system.
15. are needed for tasks that require higher privileges than those which a common user has.

11.7 Review Questions

1. What are the main steps to become Superuser in Linux?
2. Make a detailed critical evaluation of the Super user.
3. What do think are the limitations and advantages of the super user?
4. How do you foresee the future of super user?
5. What is the processor to add users?
6. Analyse the various ways to Change Passwords and delete users
7. Explain the ways to predict the groups of the user to which it belongs.
8. Write short note on:
 - (a) User and SuperUser.
 - (b) Creating Groups
 - (c) Deleting a Group
 - (d) Modifying Group Membership
 - (e) SetUID and SetGID Programs
9. What are the principal weaknesses of Normal Users in Linux?
10. What are major directions to let normal users shut down the computer in Linux?

Answers: Self Assessment**Notes**

- | | | | |
|------------------|--------------|-----------------------|-----------|
| 1. administering | 2. permanent | 3. userdel | 4. chown |
| 5. normally | 6. sudo | 7. su | 8. visudo |
| 9. /etc/sudoers | 10. execute | 11. logged | 12. root |
| 13. running | 14. power | 15. setuid and setgid | |

11.8 Further Readings**Books**

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.

**Online links**

http://www.faqs.org/docs/linux_admin/c2318.html

<http://news.softpedia.com/news/Managing-User-Accounts-on-Linux-36645.shtml>

Unit 12: Server Role: Linux as Web Server

CONTENTS

Objectives

Introduction

12.1 Web Server

12.2 Apache Web Server

12.3 Starting Apache

12.4 Configuring your Server for Apache

12.5 Setting up First Web Page

12.6 Summary

12.7 Keywords

12.8 Self Assessment

12.9 Review Questions

12.10 Further Readings

Objectives

After studying this unit, you will be able to:

- Know about apache web server
- Describe installing apache
- Discuss starting apache
- Explain configuring web server
- Analyse setting up first web page

Introduction

Apache is the most extensively used HTTP-server in the world today. It surpasses all free and commercial competitors on the market, and provides a myriad of features; more than the nearest competitor could give you on a UNIX variant. It is also the most used web server for a Linux system. A web server like Apache, in its simplest function, is software that displays and serves HTML pages hosted on a server to a client browser that understands the HTML code. Mixed with third party modules and programs, it can become powerful software, which will provide strong and useful services to a client browser.

12.1 Web Server

A web server is a program that runs on a host computer (also, confusingly enough, called a web server) that provides web sites. In other words, the web server program sits around awaiting requests from visitors\' web browsers for objects it has in its possession, and then sends these objects back for the visitor\'s viewing pleasure. Objects that web servers can serve include

HTML documents, plain text, images, sounds, video, and other forms of data. These objects may not necessarily exist in static form, but instead are generated on-the-fly by programs run by the server; CGI scripts are the most common of these programs.

Web servers and browsers communicate using HTTP, **H**ypertext **T**ransfer Protocol, a simple but efficient language for requesting and transmitting data over a network. Thus, you'll sometimes hear web servers referred to as HTTP servers. Web servers come in various shapes and sizes. They run under a variety of operating systems, have varying levels of power and complexity, and range in price from rather expensive to free.

Apache:

Apache is

1. **Powerful:** Apache's performance and reliability is well-known.
2. **Features-Rich:** The Apache server sports a host of features, as well as: XML support, server-side includes, powerful URL-rewriting, and virtual hosting, to name but a few. We'll be talking about some of these features in future articles.
3. **Modular:** Looking for a characteristic not implemented in the core Apache server? Chances are you will find a module that can add the functionality you need. Objects that web servers can serve include HTML documents, plain text, images, sounds, video, and other forms of data.
4. **Extensible:** Can't find a module that suits your intention? Well, as Apache is open source, you can write one yourself. In fact, you can even make changes to the inner workings of Apache. All the information you need is right there in the source code and numerous online resources. Share your patches or modules with the community by making them open source as well!
5. **Popular:** At the time of this writing, Apache holds a smidge under 60 percent of the web server marketplace. And, yes, popularity does count; help abounds and is only a mailing list or newsgroup posting away.



Did u know? What is Apache?

12.2 Apache Web Server

The Apache Web server, for those of you who haven't heard of it, is debatably the most popular Web server in use on the Internet today. While Microsoft contends that its Internet Information Server (IIS) is making huge gains, it's still struggling in many ways against Apache. Why?

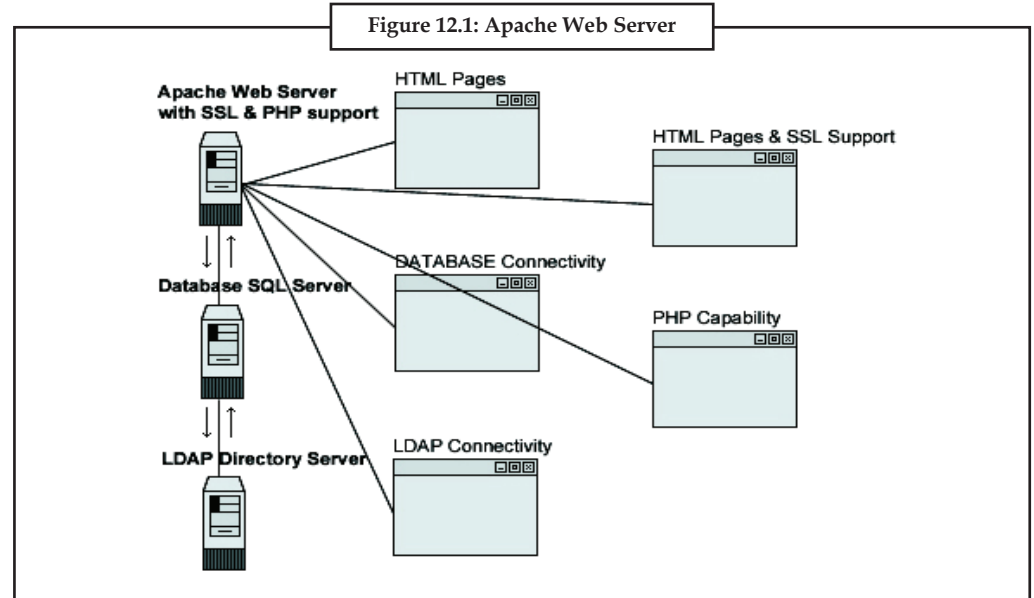
For beginners, you don't have to be running Windows to run Apache. It was first developed on the various Unix/Linux/BSD platforms, and then recently ported to Win32. Internet Information Server, while a very good Web server on the NT platform, is trapped in the "Windows-only" world. While IIS has many handy features, not everyone wants to run NT for their Web server's OS.

One more reason for Apache's widespread acceptance is its overall stability. While you can slow down an Apache Web server (especially if you run tons of PERL/CGI scripts on it), you can rarely, if ever, kill one. The Apache Web server service is near bombproof.

Finally, it's relatively fast. I say "relatively" as it's relative to what you're doing with it. If you're hosting a Web site with mostly static content, Apache is a fireball. If of desktop-centric Linux such as Caldera, it's rare that you won't have the Apache server installed. During your Linux installation, if you see an option for "Web services" makes sure to select it so that Apache will

Notes

be installed. If you're new to the Unix/Linux/BSD world, I should warn you about something. There are two types of installation packages - source and binary. If you're new to the Unix world, try to download a binary format. A binary is the fully compiled version of the application that's ready to be installed on your system.



A source package is just that, it's the source code to the application. This way you have to compile it into an executable program all by yourself. One often messes up while compiling an application. If you're fortunate, you'll find your error immediately. If you're unlucky, it could be hours, days or weeks before you find out, and then you have to spend time re-compiling it again to fix your mistakes. Do yourself a favor when first learning Apache - find the binary package for your OS. You can learn the finer points of compilation later.



Task "If you're hosting a Web site with mostly static content, Apache is a fireball."
 Comment

12.3 Starting Apache

Those of you running the RedHat Linux supply may want to take advantage of RedHat's RPM ("RedHat Package Manager") system. Almost identical to a binary, an RPM is further customized to play nicely with other RPMs and provide a consistent interface to installing, updating, and removing binaries. They often entail a loss of flexibility and clarity – for instance, it's not readily apparent where the contents of some packages will end up. That said, for Linux newcomers or when installing a small standard component, RPMs are simple and reliable.

Bear in mind that an Apache RPM may previously be installed on your system depending on how

Linux was formerly installed on your computer. To find out, at the shell prompt, type:

```
rpm
-qa | grep apache
```

If you see something like apache-1.3.9xxx, an Apache RPM has already been installed and you can skip on to

\“Starting Apache\”.

If you don't have an Apache RPM, you must obtain one. RedHat 6.x

Onwards ships with apache-1.x.x-x.i386.rpm in the RedHat/RPMS directory on the installation CD. Or, point your Web browser at you throw tons of CGI scripts at it, while making database calls at the same time, you're going to slow it down. Though much of the slowdown will come from your scripts themselves, and not Apache.

These installation instructions assume

1. Commands are Unix-compatible.
2. The source path is /var/tmp, other paths are possible.
3. Installations were tested on Red Hat Linux 6.1 and 6.2.
4. All steps in the installation will happen in super-user account root.
5. Apache version number is 1.3.12
6. Mod_SSL version number is 2.6.4-1.3.12
7. Mod_Perl version number is 1.24
8. Mod_PHP version number is 4.0.0

Following table enlists the required action points for installing the Apache Web server

Apache Homepage: http://www.apache.org/	Mod_Perl Homepage: http://perl.apache.org/
Apache FTP Site: 63.211.145.10	Mod_Perl FTP Site: 63.211.145.10
You must be sure to download: apache_1.3.12.tar.gz	You must be sure to download: mod_perl-1.24.tar.gz
Mod_SSL Homepage: http://www.modssl.org/	Mod_PHP Homepage: http://www.php.net/
Mod_SSL FTP Site: 129.132.7.171	You must be sure to download: php-4.0.0.tar.gz
You must be sure to download: mod_ssl-2.6.4-1.3.12.tar.gz	

And don't forget that these are the basics if you are following the steps described by us exactly.

1. OpenSSL should be previously installed on your system if you want Apache and SSL encryption support.
2. PostgreSQL supposed to be already installed on your system if you want Apache and PostgreSQL database connectivity support.
3. MM should be already installed on your system if you want Apache and MM high-performance RAM-based session cache support.
4. OpenLDAP should be already installed on your system if you want Apache and LDAP directory connectivity support.
5. IMAP & POP should be already installed on your system if you want Apache and IMAP & POP capability.

Before you decompress the tarballs, It is a good plan to make a list of files on the system before you install Apache, and one afterwards, and then compare them using diff to find out what file it placed where. Simply run **find / * > Apache1** before and **find / * > Apache2** after you install the software, and use **diff Apache1 Apache2 > Apache-Installed** to get a list of what changed.

Notes

To compile, decompress the tarballs (tar.gz).

```
[root@deep ]/# cp apache_version.tar.gz /var/tmp
[root@deep ]/# cp mod_ssl-version-version.tar.gz /var/tmp
[root@deep ]/# cp mod_perl-version.tar.gz /var/tmp
[root@deep ]/# cp php-version.tar.gz /var/tmp
[root@deep ]/# cd /var/tmp/
[root@deep ]/tmp# tar xzpf apache_version.tar.gz
[root@deep ]/tmp# tar xzpf mod_ssl-version-version.tar.gz
[root@deep ]/tmp# tar xzpf mod_perl-version.tar.gz
[root@deep ]/tmp# tar xzpf\ php-version.tar.gz
```

Apache Command Line

You may require to run Apache from the command line infrequently. You can obtain a complete listing of the Apache command line choices by using the “-help” argument. For instance, you would need to open a command window and cd to the Apache directory. On UNIX, cd into the bin directory under Apache. Now run the command:

```
apache -help
```

Or on UNIX:

```
apache-lyris -help
```

With respect to the command line tools mentioned herein, it is implied that you are running in a command window (MS-DOS prompt on Windows or xterm/cmdtool or equivalent on UNIX/Linux) and are at present in the Apache directory.

UNIX Specific Tools

If you are running on Solaris or Linux, there are suitable scripts available to you to start, stop, and restart Apache. These are kept in the Apache/bin directory, directly below the Lyris directory. For example, from that directory, you would run:

```
./stop
```

or

```
./start
```

or

```
./restart
```

Depending on what you require to do.

Windows Specific Tools

If you are running on Windows, you may have need for numerous different command line tools. For example, on Windows you can start Apache in the foreground very easily by issuing the following command from the Apache directory:

```
apache
```


Running Apache by means of a terminal window is good for debugging, though it blocks that shell. You can, certainly, type Ctrl+C to kill that job, but in that case Apache won't exit cleanly. Ideally, you should open another command window and cd to the Apache directory, then run the command:

```
apache -k shutdown
```

If you are running an edition of Windows that supports services, for example Windows NT or Windows 2000, you can install Apache as a service using the following command:

```
apache -d path_to_apache -i
```

In the above example, the *path_to_apache* variable should be replaced by the path to your Apache directory. Thus, if you accepted the defaults in your install, this command will appear like:

```
apache -d C:\lyris\apache -i
```

If you would like to uninstall the Apache service, run the following command:

```
apache -u
```

If the Apache service is installed, and you want to rapidly start and stop it without using the "Services" applet, you can use the "net" utility given by Windows. The syntax of the "net" command, with respect to stopping and starting services, is:

```
net [stop | start] apache
```



Note

Most of the Apache configuration features are cross platform. That means that if you make a modification to Apache on Linux, that same adjustment, or one extremely similar, is available on other various flavors of Linux using the same syntax!

12.4 Configuring your Server for Apache

Once you've got the files, you need to tell your machine where to find everything by configuring the source files. The easiest way is to believe all the defaults and just type:

```
./configure
```

Of course, most people don't want to accept just the default choices. The most significant option is the `prefix=PREFIX` option. This specifies the directory where the Apache files will be installed. You can also set specific environment variables and modules. Some of the modules I like to have installed include:

`mod_alias` - to map different parts of the URL tree

`mod_include` - to parse Server Side Includes

`mod_mime` - to associate file extensions with its MIME-type

`mod_rewrite` - to rewrite URLs on the fly

`mod_speling` (sic) - to help your readers who might misspell URLs

`mod_ssl` - to allow for strong cryptography using SSL

`mod_userdir` - to allow system users to have their own web page directories

Notes

Build Apache

As with any source installation, you'll then require to build the installation:

```
make
make install
```

Customize Apache

Assuming that there were no troubles, you are ready to customize your Apache configuration. This really just amounts to editing the httpd.conf file. This file is located in the PREFIX/conf directory.

```
vi PREFIX/conf/httpd.conf
```

Follow the instructions in this file to edit your configuration the way you would like it. More help is available on the Apache website.

Test your Apache Server

Open a web browser on the same machine and type `http://localhost/` in the address box. You should see a page. It will say in big letters. This is excellent news, as it means your server is installed correctly.

Start Editing/Uploading Pages to your Newly Installed Apache Web Server

Once your server is up and running you can start posting pages. Have fun constructing your website.

Requirements

The following requirements exist for building Apache:

1. **Disk Space:** Make sure you have at least 50 MB of temporary free disk space obtainable. After installation Apache occupies approximately 10 MB of disk space. The actual disk space requirements will vary considerably based on your chosen configuration options and any third-party modules.
2. **ANSI-C Compiler and Build System:** Make sure you have an ANSI-C compiler installed. The GNU C compiler (GCC) from the Free Software Foundation (FSF) is not compulsory (version 2.7.2 is fine). In addition, your PATH must contain basic build tools such as make.
3. **Accurate time keeping:** Elements of the HTTP protocol are expressed as the time of day. So, it's time to examine setting some time synchronization facility on your system. Usually the ntpdate or xntpd programs are used for this purpose which are based on the Network Time Protocol (NTP). See the Usenet newsgroup `comp.protocols.time.ntp` and the NTP homepage for more details about NTP software and public time servers.
4. **Perl 5 [OPTIONAL]:** For some of the support scripts like `apxs` or `dbmmanage` (which are written in Perl) the Perl 5 interpreter is essential (versions 5.003 or newer are sufficient). If you have multiple Perl interpreters (for example, a system wide install of Perl 4, and your own install of Perl 5), you are advised to use the `--with-perl` option (see below) to make sure the correct one is used by configure. If no Perl 5 interpreter is found by the configure script, you will not be able to use the affected support scripts. Of course, you will still be able to build and use Apache 2.0.



Caution
compliant.

If you don't have GCC then at least make sure your vendor's compiler is ANSI

Customizing Apache for Windows

Apache is configured by the files in the conf subdirectory. These are the same files make use to configure the Unix version, but there are a few different directives for Apache on Windows. See the directive index for all the available directives.

The main differences in Apache for Windows are:

1. Because Apache for Windows is multithreaded, it does not use a separate process for each request, as Apache does on Unix. Instead there are generally only two Apache processes running: a parent process, and a child which handles the requests. Within the child process each request is handled by a separate thread.

The process management directives are also different:

- (a) *MaxRequestsPerChild*: Like the Unix directive, this manage that how many requests a single child process will serve before exiting. However, unlike on Unix, a single process serves all the requests at once, not just one. If this is set, it is recommended that a very high number is used. The recommended default, *MaxRequestsPerChild* 0, causes the child process to never exit.
- (b) *ThreadsPerChild*: This directive is new. It informs the server how many threads it should use. This is the maximum number of connections the server can handle at once, so be sure to set this number high enough for your site if you get a lot of hits. The recommended default is *ThreadsPerChild* 50.

2. The directives that admit filenames as arguments must use Windows filenames instead of Unix ones. However, because Apache uses Unix-style names internally, you must use forward slashes, not backslashes. Drive letters can be used; if omitted, the drive with the Apache executable will be assumed.
3. While filenames are usually case-insensitive on Windows, URLs are still treated internally as case-sensitive before they are mapped to the filesystem. For example, the `<Location>`, `Alias`, and `ProxyPass` directives all use case-sensitive arguments. For this reason, it is particularly important to use the `<Directory>` directive when attempting to limit access to content in the filesystem, since this directive applies to any content in a directory, regardless of how it is accessed. If you wish to assure that only lowercase is used in URLs, you can use something like:

```
RewriteEngine On
```

```
RewriteMap lowercase int:tolower
```

```
RewriteCond %{REQUEST_URI} [A-Z]
```

```
RewriteRule (.*) ${lowercase:$1} [R,L]
```

4. Apache for Windows contains the capability to load modules at runtime, without recompiling the server. If Apache is compiled normally, it will install a number of optional modules in the `\Apache2\modules` directory. To activate these or other modules, the new `LoadModule` directive must be used. For example, to activate the status module, use the following (in addition to the status-activating directives in `access.conf`):

```
LoadModule status_module modules/mod_status.so
```

Information on creating loadable modules is also available.

Notes

5. Apache can also load ISAPI (Internet Server Application Programming Interface) extensions (i.e. internet server applications), such as those used by Microsoft IIS and other Windows servers. Apache **cannot** load ISAPI Filters.
6. When running CGI scripts, the technique Apache uses to find the interpreter for the script is configurable using the ScriptInterpreterSource directive.
7. Since it is often hard to manage files with names like .htaccess in Windows, you may find it useful to change the name of this per-directory configuration file using the AccessFilename directive.
8. Some errors during Apache startup are logged into the Windows event log when running on Windows NT. This mechanism acts as a backup for those situations where Apache cannot even access the normally used error.log file.



Note You can view the Windows event log by using the Event Viewer application on Windows NT 4.0, and the Event Viewer MMC snap-in on newer versions of Windows.

Running Apache as a Console Application

Running Apache as a service is typically the recommended way to use it, but it is sometimes easier to work from the command line (on Windows 9x running Apache from the command line is the recommended way due to the lack of reliable service support.)

To run Apache from the command line as a console application, use the following command:

```
httpd
```

Apache will execute, and will stay running until it is stopped by pressing Control-C. You can also run Apache via the shortcut Start Apache in Console placed to Start Menu → Programs → Apache HTTP Server 2.0.xx → Control Apache Server during the installation. This will open a console window and start Apache inside it. If you don't have Apache installed as a service, the window will remain visible until you stop Apache by pressing Control-C in the console window where Apache is running in. The server will exit in a few seconds. However, if you do have Apache installed as a service, the shortcut starts the service. If the Apache service is running already, the shortcut doesn't do anything.

You can tell a running Apache to stop by opening another console window and entering:

```
httpd -k shutdown
```

This should be favored over pressing Control-C because this lets Apache end any current operations and clean up gracefully. You can also tell Apache to restart. This forces it to reread the configuration file. Any operations in progress are allowed to complete without interruption. To restart Apache, use:

```
httpd -k restart
```

Note for people recognizable with the Unix version of Apache: these commands provide a Windows equivalent to kill -TERM *pid* and kill -USR1 *pid*. The command line option used, -k, was chosen as a reminder of the kill command used on Unix.

If the Apache console window closes instantly or unexpectedly after startup, open the Command Prompt from the Start Menu → Programs. Change to the folder to which you installed Apache, type the command `apache`, and read the error message. Then change to the logs folder, and

review the error.log file for configuration mistakes. If you accepted the defaults when you installed Apache, the commands would be:

```
c:
cd "\\Program Files\Apache Group\Apache2\bin"
httpd
```

Then wait for Apache to stop, or press Control-C. Then enter the following:

```
cd ..\logs
more < error.log
```

When working with Apache it is essential to know how it will find the configuration file. You can specify a configuration file on the command line in two ways:

1. `-f` specifies an absolute or relative path to a particular configuration file:

```
httpd -f "c:\my server files\anotherconfig.conf"
```

or

```
httpd -f files\anotherconfig.conf
```
2. `-n` specifies the installed Apache service whose configuration file is to be used:

```
httpd -n "MyServiceName"
```

In both of these cases, the proper `ServerRoot` should be set in the configuration file.

If you don't indicate a configuration file with `-f` or `-n`, Apache will use the file name compiled into the server, such as `conf\httpd.conf`. This built-in path is relative to the installation directory. You can verify the compiled file name from a value labelled as `SERVER_CONFIG_FILE` when invoking Apache with the `-V` switch, like this:

```
httpd -V
```

Apache will then try to determine its `ServerRoot` by trying the following, in this order:

A `ServerRoot` directive via the `-C` command line switch.

The `-d` switch on the command line.

Current working directory.

A registry entry which was produced if you did a binary installation. The server root compiled into the server. This is `/apache` by default, you can verify it by using `apache -V` and looking for a value labelled as `HTTPD_ROOT`.

Throughout the installation, a version-specific registry key is created in the Windows registry. The location of this key depends on the type of the installation. If you chose to install Apache for all users, the key is located under the `HKEY_LOCAL_MACHINE` hive, like this (the version numbers will of course vary between different versions of Apache:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Apache\2.0.43
```

Likewise, if you chose to install Apache for the current user only, the key is located under the `HKEY_CURRENT_USER` hive, the contents of which are dependent of the user currently logged on:


```
HKEY_CURRENT_USER\SOFTWARE\Apache Group\Apache\2.0.43
```

This key is compiled into the server and can facilitate you to test new versions without affecting the current version. Of course, you must take care not to install the new version in the same directory as another version.

Notes

If you did not do a binary install, Apache will in some situation complain about the missing registry key. This warning can be ignored if the server was otherwise able to find its configuration file.

The value of this key is the ServerRoot directory which consist the conf subdirectory. When Apache starts it reads the httpd.conf file from that directory. If this file contains a ServerRoot directive which contains a different directory from the one obtained from the registry key above, Apache will forget the registry key and use the directory from the configuration file. If you copy the Apache directory or configuration files to a new location it is vital that you update the ServerRoot directive in the httpd.conf file to reflect the new location.

 <i>Task</i>	Describe the various ways of specifying configuration file on the command line.
--	---

Running Apache as a Service

Apache can be run as a service on Windows NT. There is some highly experimental support for related behavior on Windows 9x. You can install Apache as a service automatically during the installation. If you chose to install for all users, the installation will make an Apache service for you. If you specify to install for yourself only, you can manually register Apache as a service after the installation. You have to be a member of the Administrators group for the service installation to succeed.

To be able to manage an Apache service with the monitor, you have to first install the service (either automatically via the installation or manually).

You can install Apache as a Windows NT service as follows from the command prompt at the Apache bin subdirectory:

```
httpd -k install
```

If you need to specify the name of the service you wish to install, use the following command. You have to do this if you have several different service installations of Apache on your computer.

```
httpd -k install -n "MyServiceName"
```

If you need to have specially named configuration files for different services, you must use this:

```
httpd -k install -n "MyServiceName" -f "c:\files\my.conf"
```

If you use the first command without any special parameters except -k install, the service will be called Apache2 and the configuration will be understood to be conf\httpd.conf.

Removing an Apache service is easy. Just use:

```
httpd -k uninstall
```

The specific Apache service to be uninstalled can be specified by using:

```
httpd -k uninstall -n "MyServiceName"
```

Normal starting, restarting and shutting down of an Apache service is generally done via the Apache Service Monitor, by using commands like NET START Apache2 and NET STOP Apache2 or via normal Windows service management. Before starting Apache as a service by any means, you should test the service's configuration file by using:

```
httpd -n "MyServiceName" -t
```

You can control an Apache service by its command line switches, too. To start an installed Apache service you'll use this:

```
httpd -k start
```

To stop an Apache service via the command line switches, use this:

```
httpd -k stop
```

or

```
httpd -k shutdown
```

You can also restart a running service and force it to revise its configuration file by using:

```
httpd -k restart
```

By default, all Apache services are indexed to run as the system user (the LocalSystem account). The LocalSystem account has no privileges to your network via any Windows-secured mechanism, including the file system, named pipes, DCOM, or secure RPC. It has, however, wide privileges locally.



Did u know? Is Apache comes with a value called the Apache Service Monitor?

Never grant any network privileges to the LocalSystem account! If you need Apache to be capable to access network resources, create a separate account for Apache as noted below.

You may want to create a separate account for running Apache service(s). Specially, if you have to access network resources via Apache, this is strongly recommended.

Create a normal domain user account, and be sure to learn its password.

Grant the newly-created user a freedom of Log on as a service and Act as part of the operating system. On Windows NT 4.0 these privileges are granted via User Manager for Domains, but on Windows 2000 and XP you probably want to use Group Policy for propagating these settings. You can also manually set these via the Local Security Policy MMC snap-in.

Verify that the created account is a member of the Users group.

Grant the account read and execute (RX) rights to all document and script folders (htdocs and cgi-bin for example).

Grant the account change (RWXD) rights to the Apache logs directory.

Grant the account read and execute (RX) rights to the Apache.exe binary executable.

It is generally a good practice to grant the user the Apache service runs as read and execute (RX) access to the whole Apache2 directory, except the logs subdirectory, where the user has to have at least change (RWXD) rights.

If you allow the account to log in as a user and as a service, then you can log on with that account and test that the account has the privileges to perform the scripts, read the web pages, and that you can start Apache in a console window. If this works, and you have followed the steps above, Apache should execute as a service with no problems.

Error code 2186 is a good sign that you need to review the "Log On As" configuration for the service, since Apache cannot access a required network resource. Also, pay close attention to the privileges of the user Apache is configured to run as.

As starting Apache as a service you may encounter an error message from the Windows Service Control Manager. For example, if you try to start Apache by using the Services applet in the Windows Control Panel, you may get the following message:

Notes

Could not start the Apache2 service on \\COMPUTER

Error 1067; The process terminated unexpectedly.

You will get this common error if there is any problem with starting the Apache service. In order to see what is really causing the problem you should follow the instructions for Running Apache for Windows from the Command Prompt.

There is some support for Apache on Windows 9x to behave in a related manner as a service on Windows NT. It is highly experimental. It is not of production-class reliability, and its future is not guaranteed. It can be mostly regarded as a risky thing to play with - proceed with caution!

There is some distinction between the two kinds of services you should be aware of:

Apache will make effort to start and if successful it will run in the background. If you run the command

```
httpd -n "MyServiceName" -k start
```

via a shortcut on your desktop, for example, then if the service starts effectively, a console window will flash up but it immediately disappears. If Apache detects any errors on startup such as incorrect entries in the httpd.conf configuration file, the console window will remain visible. This will display an error message which will be useful in tracking down the cause of the problem.

Windows 9x does not support NET START or NET STOP commands. You must control the Apache service on the command prompt via the -k switches.

Apache and Windows 9x offer no support for running Apache as a detailed user with network privileges. In fact, Windows 9x offers no security on the local machine, either. This is the simple reason because of which the Apache Software Foundation never endorses use of a Windows 9x -based system as a public Apache server. The primitive support for Windows 9x exists only to assist the user in developing web content and learning the Apache server, and perhaps as an intranet server on a secured, private network.

You can also use the Apache Service Monitor to manage Windows 9x pseudo-services.



Note

Once you have established that Apache runs correctly as a console application you can install, control and uninstall the pseudo-service with the same commands as on Windows NT.

Testing the Installation

After starting Apache (either in a console window or as a service) it will be paying attention on port 80 (unless you changed the Listen directive in the configuration files or installed Apache only for the current user). To connect to the server and access the default page, launch a browser and enter this URL:

```
http://localhost/
```

Apache should react with a welcome page and a link to the Apache manual. If nothing happens or you get an error, look in the error.log file in the logs subdirectory. If your host is not connected to the net, or if you have serious problems with your DNS (Domain Name Service) configuration, you may have to use this URL:

```
http://127.0.0.1/
```


If you ensue to be running Apache on an alternate port, you need to explicitly put that in the URL:

```
http://127.0.0.1:8080/
```

One time your basic installation is working, you should configure it properly by editing the files in the conf subdirectory. Because Apache **cannot** share the similar port with another TCP/IP application, you may need to stop, uninstall or reconfigure certain other services before running Apache. These conflicting services include other WWW servers and some firewall implementations. If you change the configuration of the Windows NT service for Apache, first attempt to start it from the command line to make sure that the service starts with no errors.



Did u know? How can we test the installation of apache?

12.5 Setting up First Web Page

Here are the basics of setting up a web site. The whole process from getting a domain name to your first page actually showing up on the World Wide Web.

1. **Get a Domain Name:** Decide on a domain name and check to see if it is obtainable with an Internet registrar. Registrars are all very alike, but for the purposes of this tutorial I'll use www.GoDaddy.com. Currently a .com domain name costs about \$10 per year. (other domain name possibilities are .net, .org, .info, .biz, etc.) Once at www.GoDaddy.com you'll have to generate an account. From this account you will be able to purchase and manage your domain name. Once your domain name is registered, you will need to identify a name server that corresponds with your hosting service. You will see a "Nameserver" icon that will take you to a page to specify your name server. (this will be explained further in Step Three).

Accepting the necessary components that make up a great domain name will help you coming up with and buying a name that will give you the majority mileage to help you meet your specific online objectives. The basic individuality of a good name are pertinent, catchy, easily remembered and able to self-promote. Choosing the right name for your online objectives take time and much thought. Endurance and perseverance are the names of the game here. Because many great and popular names may have already been taken up, it could be a challenge finding an ideal name that is available.

Set aside some time to allow manually to brainstorm probable names for your web site. This is important, as appropriate names rarely come 'just like that'. Determine what you propose to use your domain for - whether it's for business or personal purposes - then come up with a list of relevant names that are short, sweet and catchy. The name you lastly end up with should be easily remembered by as many people as possible and have the ability to draw targeted visitors to your site. Building a healthy flow of targeted traffic is a key objective of any domain name. This flow of traffic is especially important if you are using your name for an online business as this pool of targeted visitors could end up becoming your online customers and contribute significantly to the sales of your products offered on your web site.

Once you've suggest a list of probable names, you then have to look for a suitable domain registration company. These companies are more commonly known as registrars and can help you get a domain name simply (if you choose the right registrar, that is). Registrars like Active-Domain.com are not only reliable but boast capable and friendly customer service as well as a good track record with their customers (past and present).

While there are frequent registrars' contribution tremendously low rates for helping you get domain name, it is vital that you do not act on impulse and sign yourself up with the cheapest registrar obtainable. You want to look for a registrar that is not only reliable and

Notes

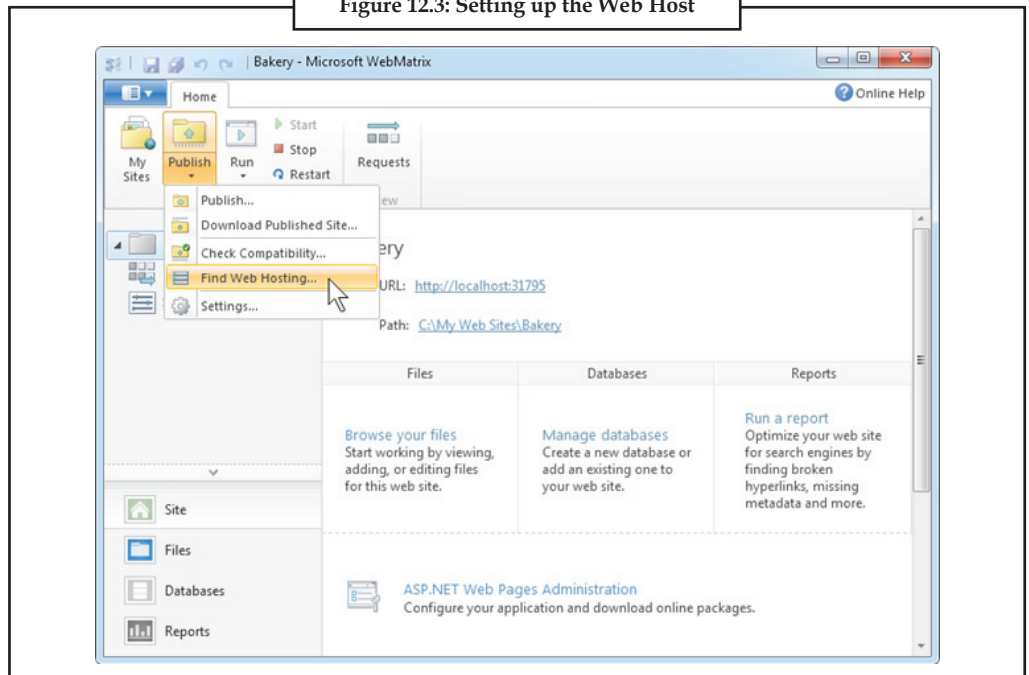
recognized but one that can meet all your specific domain needs. Be careful when choosing a registrar to help you get domain, as you will be ‘working’ with that registrar for as long as the registration period you choose. Finding a company that is dependable and recognized will also give you a peace of mind that your domain needs are well taken care of.

Figure 12.2



2. **Set up a Web Hosting Account:** You have many choices when it comes to web site hosting. If you are just starting out and have a site that will not have millions of visitors per month, you can get a good hosting account that includes e-mail service for about \$10 per month. For the purposes of this tutorial I will use Dreamhost.com. Once at Dreamhost, select a hosting plan and create an account. Once you create this account they will e-mail the web address to your web hosting account control panel. Here you will be able to set up everything you need to get started.

Figure 12.3: Setting up the Web Host



3. **Configure your Web Hosting Account:** In your control panel you will see an option call "Manage Domains." Select this option and at the top you will see an "Add New Domain" link. This is where you type in the domain name you registered at GoDaddy.com. The other options on this page can just be left as they are for most basic web sites. You will be asked the name of the folder you want to store you content in on the web server, you should just name it something that makes sense related to your domain name. Now that your hosting account is set up, you need to make sure that your domain registrar knows who is hosting your site. You do this by going back into your registrar account and specifying a name server. Click on the "Nameserver" icon and enter at least to nameservers on the page. Dreamhost's nameservers are ns1.dreamhost.com and ns2.dreamhost.com. (Note: it may take a day or two for your new domain name to work throughout the Internet. This is called domain name propagation)
4. **Make a Home Page and send it to your Web Host:** At this point you need to create your home page. There are many easy ways to make this page, but the key is to name it index.htm . This is the default home page name that most web hosts use. So if the domain name you set up is www.yoursite.com - then your index.htm page will show up at www.yoursite.com . If you name your file mypage.htm, then it will appear at www.yoursite.com/mypage.htm. So using the name index.htm will not require your site visitors to type in anything past the .com part of your domain name.



Task

"Once you create this account they will e-mail the web address to your web hosting account control panel." Comment



Caselet

Website using PHP

I am using a PC with XP sp2 installed on it, and with 128 MB RAM. I am learning Php (server scripting language). What must I install in order to execute scripts on my PC?

RAVI TEJA

To build a Web site using PHP, basically an HTTP server, PHP and Database are required. Some HTTP servers available are Apache and Microsoft IIS. While using PHP, most people choose apache for HTTP server and MySQL for database because all are free or open source software. You can install these components (PHP, Apache, MySql) either individually or a third-party component, say, EasyPHP, that integrates PHP, Apache and MySQL.

The integrated packages, such as EasyPHP, LAMP, etc, are means to get everything up and running in one step.

If you would like to install and use as individual packages, please follow these steps: Download the apache server from the following apache URL: <http://httpd.apache.org/download.cgi> Win32 Source: httpd-2.2.6-win32-srcr2.zip

To download PHP setup binaries, please look up the URL: <http://www.php.net/downloads.php>.

The following are Windows binaries required for the PHP 5.2.4 version. PHP 5.2.4 zip package [9,702Kb] PHP 5.2.4 installer [22,002Kb] PECL 5.2.4 Win32 binaries [4,363Kb]

Contd...

Notes

PHP 5.2.4 Non-thread-safe Win32 binaries [9,608Kb] PECL 5.2.4 Non-thread-safe Win32 binaries [4,110Kb]

To install MySQL, navigate to the mysql site, <http://dev.mysql.com/downloads/mysql/5.0.html> Windows downloads Windows Essentials (x86) ver.5.0.4522.9M Windows ZIP/Setup.EXE (x86) ver.5.0.4542.4M Without installer (unzip in C:\) ver.5.0.4550.0M

If you would like to install and use as a single package, you can download EasyPHP, which integrates PHP interpreter, the MySQL database program, the Apache web server, and the PHPMyAdmin database administration program.

Source: <http://www.thehindubusinessline.in/ew/2007/10/15/stories/2007101550150401.htm>

12.6 Summary

Web servers and browsers communicate using HTTP, Hypertext Transfer Protocol, a simple but effective language for requesting and transmitting data over a network. Thus, you'll sometimes hear web servers referred to as HTTP servers. Web servers come in various shapes and sizes. The Apache Web server, for those of you who haven't heard of it, is arguably the most popular Web server in use on the Internet today. While Microsoft contends that its Internet Information Server (IIS) is making huge gains, it's still struggling in many ways against Apache. Apache is configured by the files in the conf subdirectory. These are the same files used to configure the Unix version, but there are a few different directives for Apache on Windows. See the directive index for all the available directives.

12.7 Keywords

HTML: Hypertext Markup Language

HTTP: Hypertext Transfer Protocol

IIS: Internet Information Server

mod_alias: To map different parts of the URL tree

mod_include: To parse Server Side Includes

mod_mime: To associate file extensions with its MIME-type

mod_rewrite: To rewrite URLs on the fly

mod_speling: To help your readers who might misspell URLs

mod_ssl: To allow for strong cryptography using SSL

mod_userdir: To allow system users to have their own web page directories

PREFIX: This option specifies the directory where the Apache files will be installed.

Web server: It is a program that runs on a host computer that serves up web sites.

12.8 Self Assessment

Fill in the blanks:

1. Apache is the most widely used in the world today.
2. A is a program that runs on a host computer that serves up web sites.
3. The is arguably the most popular Web server in use on the Internet today.

4. option specifies the directory where the Apache files will be installed.
5. The actual requirements will vary considerably based on your chosen configuration options and any third-party modules.
6. Objects that web servers can serve include, plain text, images, sounds, video, and other forms of data.
7. During your, if you see an option for “Web services” makes sure to select it so that Apache will be installed.
8. Usually the programs are used for this purpose which are based on the Network Time Protocol (NTP).
9. It is particularly important to use the directive when attempting to limit access to content in the filesystem.
10. If Apache is compiled normally, it will a number of optional modules in the\Apache2\modules directory.
11. Apache should react with a and a link to the Apache manual.
12. If the closes instantly or unexpectedly after startup, open the Command Prompt from the Start Menu → Programs.
13. Some errors during are logged into the Windows event log when running on Windows NT.
14. Apache **cannot** share the similar with another TCP/IP application.
15. Grant the newly-created user a freedom of on as a service and Act as part of the operating system.

Notes

12.9 Review Questions

1. What are the differences between web server and apache web server?
2. Explain the installation and configuration of your server for apache.
3. What are the requirements for building apache?
4. Is apache different for Windows? Explain
5. Discuss the running of apache as a console application and as a service.
6. Explain how the Apache server sports a host of features?
7. “Apache is a fireball.” Comment
8. Discuss how we can set the first web page.
9. In your opinion running Apache as a service is the recommended way to use it. Why or why not?
10. How can we customize apache for windows?

Answers: Self Assessment

1. HTTP-server 2. web server 3. Apache Web server
4. PREFIX 5. disk space 6. HTML documents
7. Linux installation 8. ntpdate or xntpd 9. <Directory>

Notes

- | | | |
|--------------------|------------------|---------------------------|
| 10. Install | 11. welcome page | 12. Apache console window |
| 13. Apache startup | 14. Port | 15. Log |

12.10 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlan and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

<http://www.yolinux.com/TUTORIALS/LinuxTutorialWebSiteConfig.html>

<http://www.faqs.org/docs/securing/chap29sec239.html>

Unit 13: FTP Server

Notes

CONTENTS

Objectives

Introduction

13.1 Meaning

13.2 FTP (File Transfer Protocol)

13.3 Start FTP Server

13.4 Testing FTP Server

13.5 Using FTP

13.6 Using FTP Clients to Test Anonymous Read Access

13.7 Summary

13.8 Keywords

13.9 Self Assessment

13.10 Review Questions

13.11 Further Readings

Objectives

After studying this unit, you will be able to:

- Understand meaning of FTP server
- Discuss FTP Protocol
- Explain installing vsftpd FTP server
- Discuss starting FTP server
- Describe testing FTP server
- Know about using FTP server
- Understand using FTP clients to test anonymous Read access

Introduction

Vsftpd is a GPL approved FTP server for UNIX systems, including Linux. It is secure and extremely fast. It is stable. Don't take my word for it, though. Below, we will see evidence supporting all three assertions. We will also see a list of a few important sites which are happily using vsftpd. This demonstrates vsftpd is a mature and trusted solution.

If your major requirement from an FTP server is one of the following things then yes, vsftpd is probably the FTP server you are looking for.

1. Security
2. Performance
3. Stability

Notes

The only reason you might favor a different FTP server to vsftpd is if you really need the configurability of one of the more bloated FTP servers. Having said this, note that vsftpd caters for the vast majority of use cases. Even if vsftpd appears to be missing a feature, it is often satisfied by an external component such as PAM or xinetd / tcp_wrappers. In this regard, vsftpd is being a small modular component in the proper spirit of UNIX. Finally, consider moving to vsftpd even if it means sacrificing some whacky feature of your current FTP server. The security, performance and stability gains are waiting for you.

13.1 Meaning

Commonly when one thinks of the Internet, the first thing that comes to mind is “surfing” from one website to another. Being able to go from website to another, and view the contents is indeed the reason that the Internet is as popular as it is today, and growing bigger every year. If we set web surfing aside though, just what do we have left in terms of actual usage going on whilst on the Internet? Well one of the activities that takes place is the downloading of data files, movies, anti-virus updates, and the such. What these acts have in common is one protocol, namely the FTP protocol or File Transfer Protocol.

It should be noted that FTP also observes the client/server model. Unlike HTTP though, where there is a clear cut winner for web browsers and web servers, no such program can make the same claim as it relates to FTP. There is a large selection of FTP clients and servers out there today. It is worth noting that your version of Windows come with a built-in FTP client.

FTP itself uses the TCP transport protocol exclusively, or in other words, it never uses UDP for its transport needs. Typically an application layer protocol will use one or the other. One notable exception to that is DNS or Domain Name System. FTP also is odd in the fact that it uses two ports to accomplish its task. It typically uses port 20 for data transfer and port 21 to listen to commands. Though having data transferred over port 20 is not always the case as it can also be a different port as well. That is where the confusing part for many people comes into play. There are two modes to FTP, namely active and passive mode. These two modes are initiated by the FTP client, and then acted upon by the FTP server.

So just how does active and passive FTP work anyways? Well it all starts with the FTP client initiating a connection with the FTP server on its port 21. Port 21 is where the server is listening for commands issued to it, and in turn, which it will respond to. So we will assume that the TCP/IP handshake is complete, and as normal the client has done all of this on an ephemeral port. At this point the client begins to listen on it's ephemeral port + 1, and sends the PORT N+1 command to the server on its port 21 i.e. if the ephemeral port in use by the client is 1026, then it would listen on port 1027. Once this is done the data transfer port (port 20) on the FTP server would initiate a connection to the FTP client's ephemeral port plus 1, as indicated above. This is pretty much how an active FTP session is conducted by both the client and server. Though if the client has a firewall in place, this whole communication process will come to a grinding halt. The clients firewall would drop what it considers to be an unsolicited communication attempt on the ephemeral port plus one port for the data transfer. The way that FTP gets around this problem is by using passive FTP.



Did u know? Is FTP uses UDP for its transport needs?

Passive Approach

By using the passive mode of FTP or as it appears in the ASCII content of a packet “PASV”, FTP was able to neatly sidestep the firewall issue on the client side. It was done in the following fashion: The FTP client, let's say the built in FTP client that comes with a win32 operating system, will start up two connections to the FTP server. We need to keep in mind as well that both

connections that are initiated by the client are using ephemeral ports themselves, as it should be. By opening two connections, or sockets with the FTP server, the client is able to resolve the issue of its firewall denying access to the FTP server initiating contact on one of the client's high ephemeral ports.

One of the connections opened by the client will contact the FTP server on port 21, and issue it the PASV (passive) command, vice the normal PORT command when using active FTP. Now what happens is that the FTP server opens an ephemeral port and issues the PORT command to the FTP client. With this in hand the client then starts a connection back to the server port for the data transfer. It is a rather nifty way to deal with the aforementioned issue of Active FTP and client firewalls.

Much like some other application layer protocols, FTP has its own set of status and error codes. Just like HTTP, these numerical values will tell you what is going on with an established session. Also much like HTTP status and error codes they are broken down into five groups. It is always handy to have a link to a breakout of these nearby if you are investigating some traffic issues. Well with that said, what would an article about a protocol be without a packet showing it in action! Without further ado let's take a look at one of them.



Task

Analyze how does active and passive FTP work?

13.2 FTP (File Transfer Protocol)

File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers. FTP works on a client/server model. The server component is called an *FTP daemon*. It constantly listens for FTP requests from remote clients. When a request is received, it manages the the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

1. Anonymous
2. Authenticated

In the Anonymous mode, remote clients can access the FTP server by using the default user account called 'anonymous' or 'ftp' and transfer an email address as the password. In the Authenticated mode a user must have an account and a password. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

File Transfer Protocol (FTP), a standard Internet protocol, is the simplest way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol (SMTP), which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. FTP is commonly used to transfer Web page files from their creator to the computer that acts as their server for everyone on the Internet. It's also commonly used to download programs and other files to your computer from other servers.

As a user, you can use FTP with a simple command line interface (for example, from the Windows MS-DOS Prompt window) or with a commercial program that offers a graphical user interface. Your Web browser can also make FTP requests to download programs you select from a Web page. Using FTP, you can also update (delete, rename, move, and copy) files at a server. You need

Notes

to logon to an FTP server. However, publicly available files are easily accessed using anonymous FTP. The most common use for FTP is to download files from the Internet.

Basic FTP support is usually provided as part of a suite of programs that come with TCP/IP. However, any FTP client program with a graphical user interface usually must be downloaded from the company that makes it.

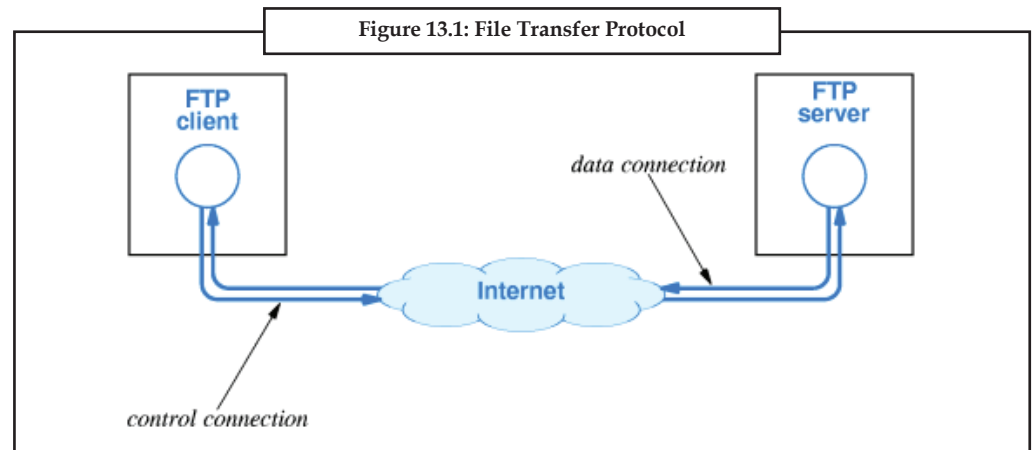
FTP (File Transfer Protocol) is the simplest and most secure way to exchange files over the Internet. Whether you know it or not, you most likely use FTP all the time.

The most common use for FTP is to download files from the Internet. Because of this, FTP is the backbone of the MP3 music craze, and vital to most online auction and game enthusiasts. In addition, the ability to transfer files back-and-forth makes FTP essential for anyone creating a Web page, amateurs and professionals alike.

When downloading a file from the Internet you're actually transferring the file to your computer from another computer over the Internet. This is why the T (transfer) is in FTP. You may not know where the computer is that the file is coming from but you most likely know its URL or Internet address.

An FTP address looks a lot like an HTTP, or Website, address except it uses the prefix ftp:// instead of http://.

Most often, a computer with an FTP address is dedicated to receive an FTP connection. Just as a computer that is setup to host Web pages is referred to as a Web server or Website, a computer dedicated to receiving an FTP connection is referred to as an FTP server or FTP site.



Did u know? What is the most common use for FTP?

vsftpd - FTP Server Installation

vsftpd is an FTP daemon accessible in Ubuntu. It is easy to install, set up, and maintain. To install **vsftpd** you can run the following command:

```
sudo apt-get install vsftpd
```

vsftpd - FTP Server Configuration

Notes

You can edit the vsftpd configuration file, `/etc/vsftpd.conf`, to modify the default settings. By default only anonymous FTP is allowed. If you wish to disable this option, you should change the following line:

```
anonymous_enable=YES
```

to

```
anonymous_enable=NO
```

By default, local system users are not permitted to login to FTP server. To change this setting, you should uncomment the following line:

```
#local_enable=YES
```

By default, users are allowed to download files from FTP server. They are not allowed to upload files to FTP server. To modify this setting, you should uncomment the following line:

```
#write_enable=YES
```

Also, by default, the anonymous users are not allowed to upload files to FTP server. To change this setting, you should uncomment the following line:

```
#anon_upload_enable=YES
```

The configuration file consists of many configuration parameters. The information about each parameter is exists in the configuration file. Alternatively, you can refer to the man page, **man 5 vsftpd.conf** for details of each parameter.

Once you configure **vsftpd** you can start the daemon. You can run following command to run the **vsftpd** daemon:

```
sudo /etc/init.d/vsftpd start
```

vsftpd (Very Secure FTP Daemon) is an FTP server for UNIX-like systems, as well as CentOS / RHEL / Fedora and other Linux distributions. It supports IPv6, SSL, locking users to their home directories and many other advanced features.



Caution You can edit the vsftpd configuration file, `/etc/vsftpd.conf`, to modify the default settings.

vsftpd Features

VSFTPD offer security, performance and constancy over other servers. A quick list of vsftpd features:

1. Virtual IP configurations
2. Virtual users
3. Run as standalone or inetd / xinetd operation
4. Per-user configuration
5. Bandwidth throttling
6. Per-source-IP configurability
7. Per-source-IP limits

Notes

8. IPv6 ready
9. Encryption support through SSL integration
10. And much more.

13.3 Start FTP Server

File Transfer Protocol, or FTP, is a robust, secure way of sharing files via a client-server model. As a Unix variant, Linux is an ideal platform for an FTP server. A popular, and very secure, FTP server available for almost all Linux systems is vsftpd (Very Secure FTP). Installing and setting up vsftpd in a Linux environment is a simple process.

1. Download and install the Linux package vsftpd. The exact procedure will vary depending on the type and version of Linux you are running. In most cases, for example with Red Hat, Suse or Ubuntu, vsftpd is a stable package that is included in the repositories that are used to update your version of Linux. In many Linux installations, vsftpd is installed by default, but not started.
2. Configure VSFTP. The exact location and name of the configuration file will vary slightly depending on the version of Linux you are using. In Ubuntu, the configuration file is “vsftpd.conf.” Edit this file to change the following settings: *ftpd_banner: Edits the message that is displayed when someone connects to the server. *listen: Enables vsftpd to run in stand alone mode. *xferlog_enable: Enables a detailed log of uploads and downloads. *connect_from_port_20: Defines whether PORT style connections use port 20 on the server. *hide_ids: Hides user IDs, listing all group users as “ftp.” *max_client: Defines the maximum number of connections allowed in standalone mode. *max_per_ip: Defines the highest number of connections allowed from a particular IP in standalone mode. *anon_root: Defines the root directory for anonymous FTP users. *anonymous_enable: Enables anonymous users. *anon_mkdir_write_enable: Allows anonymous users to create directories. For this option to work, the server must have anonymous uploads enabled, and the ftp user and group must have write permissions on the directory where the new directories will be created.
3. Once the file vsftpd.conf has been edited to your satisfaction, save the file.
4. Stop the vsftpd server by typing “sudo /etc/init.d/vsftpd stop.” You should see a message confirming that the server has stopped.
5. Restart the vsftpd server with the new configuration data in /etc/vsftpd.conf by typing “sudo /etc/init.d/vsftpd start.” You should see a message confirming that the server has started.



Task

Analyse the steps to install the vsftpd in Linux environment.

13.4 Testing FTP Server

Type the following command:

```
$ ftp localhost
$ ftp ftp.server.com
$ ftp 202.54.1.1
```

From Servage Wiki

Jump to: navigation, search

You can easily test a FTP connection from the command line in order to rule out any connection errors. This is done using `ftp`, which can be found from the command line in Windows or Linux.

Running a FTP Test on Windows 2000/XP/Vista

1. Go to Start (start button on the lower left of the desktop)
2. Choose Run
3. Type: `cmd`
4. This should bring up a DOS prompt. Once there, enter: `dir > file.txt` (to create a test file)
5. Type: `ftp ftp.servage.net`
6. Type: `yoursecretuser`
7. Type: `yoursecretpassword`
8. Type: `put file.txt` (You should see a User OK/Logged in response)
9. Exit session by entering: `quit`

Running a FTP Test using Linux Ubuntu/MacOSX

1. Start a terminal session and enter: `dir > file.txt` (to create a test file)
2. Type: `ftp ftp.servage.net`
3. Type: `yoursecretuser`
4. Type: `yoursecretpassword`
5. Type: `put file.txt` (You should see a User OK/Logged in response)
6. Exit session by entering: `quit`

With this test you can verify if the error is local (i.e. CuteFTP, FileZilla or other choice of client) or server-side. If this process was successful, but your FTP program is still not working properly, the most common reason it does not work include:

Application settings (like Firewall)

Conflicting software, virus, adware, spyware, etc.

Corrupted application

Router problems (internal Firewall)

Usability issues/improper settings. Verify the proper use of the given FTP program via its 'help' menu or contacting the software vendor's customer support

To test our new ftp server, open a new terminal and login as a normal user. Login to the ftp server by typing `ftp` at the command prompt. See an example below:

```
luzar@slackware:~$ ftp
```

```
ftp> open
```

```
(to) example.org
```

```
Connected to example.org.
```

Notes

If you can connect to the ftp server, that means you successfully setup vsftpd ftp server in Slackware Linux. Depends on how you configure vsftpd.conf file, you must provide username, whether it's local user account or anonymous. You can find more about ftp client commands at Linux ftp tutorial - FTP client guides.



Did u know? To test our new ftp server, open a new terminal and login as a normal user.


13.5 Using FTP

FTP is an acronym for File Transfer Protocol. It represents a widely-used, method for transferring files between two computers. One of the computers must be set up as a host or server. The other, known as the client, is the computer that you are using.

There are two basic types of operations:

1. When you retrieve a file from the server, you are performing a "get" operation. This is also known as "downloading."
2. When you post a file to the server, your are performing a "put" operation. This is also known as "uploading."

When you connect to an FTP server, you must send a username and password. In many cases the username "anonymous" may be used to mean "I am a visitor." Even if you are visiting and FTP site and are not prompted for a username, the FTP server will assign the anonymous username to you. Most FTP servers ask which username you would like to use.



Task

When you connect to an FTP server, you must send a username and password.

Comment

Types of FTP "Programs"

There are a number of programs to help transfer files using FTP. Technical people tend to use a command line program built into most operating systems, and it has a simple name: 'ftp'. But in order to keep network sniffers from reading your password when you connect to our ftp server, we use a secure ftp server. This means that you must use a secure ftp client that uses "SFTP", or secure file transfer protocol.

There are a number of popular freeware FTP programs such as WinSCP, CuteFTP, FileZilla, Fugu, and SmartFTP. In addition there are commercially available programs for those with too much money. For class purposes, FileZilla is a great program to use because it is free to use and runs on Windows, Macs, and Linux environments.

Side note: another way to access files is to use your web browser. However, there are lots of traces of your username and password that will be left behind, so refrain from using this method.

Using FileZilla to Transfer Files

Finally, we're talking about transferring files. Whether you're trying to get data from the server for an exercise, or trying to put your write-up into your student directory, the process is pretty much the same. There are only a few steps to this process. After a couple of uses, you'll be a pro./p>

1. Start FileZilla. Ok, that one's pretty easy. You'll see a list of your folders and files on the left side of the program window.

2. Make a connection with the file server. You can enter the server information in the “Quick connect bar” at the top of the screen.
 - (a) Host: **sftp://major.wustl.edu**
 - (b) Username: *your username*
 - (c) Password: *your password*
 - (d) Port: **22**
3. Now that you’re logged in, you’ll see a list of folders and files on the server. Double-click on a folder to open that folder and see what’s inside. (This works for both lists--your computer on the left side, and the server on the right side.) Use this method to navigate both systems so you can see the file(s) you want to transfer on one side, and you can see the destination folder on the other side.
4. Now, just drag and drop the file(s) you want to transfer. There will be a status of the transfer at the bottom of the screen that shows the progress of your transfer.
5. All done! Just log out and continue on your merry way.

Notes

Describe the using FileZilla to Transfer Files

File Transfer Protocol (FTP) servers are one of the oldest and most reliable methods of file sharing between computers. FTP servers and clients are available for all major platforms, including Unix, Linux, Macintosh and Windows. FTP servers offer advantages in reliability, security and efficiency, and offer good customization and configuration options. Access to FTP servers requires an FTP client. Paid and open source clients, both command line and graphical clients, are available for virtually all computing platforms.

Command Line Clients

1. Type the command “ftp” followed by the name of the server. For example, “ftp server.com,” where “server” is the name of the server you wish to access.
2. Type your user name.
3. Type your password.
4. Type “ls” to get a list of files available on the server.
5. Type “cd directory” to change to a subdirectory. “Directory” is the name of the subdirectory. For example, type “cd images” to change the directory to a directory named “images.”
6. Type “get filename.extension” to download a file. For example, type “get ReadMe.html” to download the file “ReadMe.html.”
7. Type “put filename.extension” to upload a file. For example, type “put ReadMe.html” to upload the file “ReadMe.html.”
8. Type “exit” to close the command line FTP client.

Graphical Clients

1. Open your graphical client.
2. Enter the location of the remote server that you wish to access in the “host” or “server” text field, depending on the client you are using. In graphical clients, FTP servers are usually entered as “ftp.server.com,” where “server.com” is the domain name of the server.


Notes

3. Enter your user name and password.
4. Click "Login" to log in to the server.
5. Browse, in the server pane, to the remote file you wish to download. In the pane of the window for the local computer, browse to the location where you wish to store the downloaded file.
6. Right-click on the file and select "Download" or "Put" to download the file.
7. Browse, in the local computer pane, to the file on your local computer that you wish to upload. In the server pane, browse to the location where you would like the file to be stored on the remote computer.
8. Right-click on the file and select "Upload" or "Put" to upload the file.
9. Exit the client to close the client and log out of the remote server.

13.6 Using FTP Clients to Test Anonymous Read Access

Anonymous FTP is the choice of Web sites that need to exchange files with numerous unknown remote users. Common uses include downloading software updates and MP3s and uploading diagnostic information for a technical support engineers' attention. Unlike regular FTP where you login with a preconfigured Linux username and password, anonymous FTP requires only a username of anonymous and your email address for the password. Once logged in to a VSFTPD server, you automatically have access to only the default anonymous FTP directory (/var/ftp in the case of VSFTPD) and all its subdirectories.

Many computersystems throughout the Internet offer files through anonymous FTP. This means that you can access a machine without having to have an account on that machine (i.e. you don't have to be an official user of the system). These anonymous FTP servers contain software, documents of various sorts, files for configuring networks, graphic images, songlyrics and all sorts of other information. Archives for electronic mailing lists are often stored on and are available through anonymous FTP. An enormous amount of information is stored on these machines and is ready for anyone who's seeking it.



Note All this is subject to change, it is a privilege and the person responsible for the machine can shut it down at any time without you being able to do anything about it.

Commands

All the normal FTP commands may be used to retrieve files. Some FTP commands are the same on different computers, but others are not. Also, some of the ftp sites offer custom commands like getting a directory with one command, 'regetting' a file or searching of directories. Read the relevant README files on the site itself for the 'special access features'.

Usually, FTP will list the commands if you type 'help' or type a question mark (?). Also, your computer's help command may have information about FTP. Try 'man ftp', 'man ftpd', 'help ftp', 'ftp /?', 'ftp -?' or 'ftp /h' (all these to be typed without quotes).

Some useful commands available on most systems include:

- ascii Switch to ascii mode. Ascii mode is the default mode and used for transferring text files
- binary Switch to binary mode. For transferring binary files like .ZIP files, .Z files and the like
- cd Change the directory on the remote computer

Notes

```

dir      List the files in the current directory on the remote computer.
ls       Same as 'dir', but shows less information sometimes.
get      Copy a file from the remote computer to yours
hash     Puts a '#' on the screen for every <number> of bytes transferred. <number>
         is 1024 in some cases, 2048 in others but is between 1024 and 4096 in most
         cases.
         Check the ftp 'help' function for more info on the number for your
         clientprogram.
help     Gives help on the use of commands within the ftp program
lcd      Change the directory on your computer (the 'l' is for local)
lpwd     Shows the present working directory (pwd) on your computer (the 'l' is
         for local). Note: this may not work on all machines. On a Unix machine,
         try !pwd if lpwd doesn't work.
mget     Copy multiple files from the remote computer to yours
pwd      Shows the present working directory (pwd) on the remote computer

```



Task

"Using FTP Clients to test anonymous read access." Comment

Procedure

Anonymous FTP is a facility offered by many machines on the Internet. This permits you to log in with the user name 'anonymous' or the user name 'ftp'. When prompted for a password, type your e-mail address -- it's not necessary, but it's a courtesy for those sites that like to know who is making use of their facility. Be courteous. Some sites require a valid e-mail address, others don't.

You can then look around and retrieve files. (Most anonymous ftp sites do not permit people to store files). Note that when you retrieve files, you have to know where the files are going to end up on your machine. This is where the 'lpwd' command comes in handy. Also note that when you have transferred a file that you want to use on your PC, but you run ftp from a Unix machine (or a similar mainframe or network machine), you will have to ftp the file from the Unix machine to your PC first (this is assuming that you can't ftp to outside your company or campus from your PC, otherwise you could have gotten the file directly to the PC).

This may sound silly, but sometimes people don't know where their files are stored or a system administrator decides to give ftp access to only a few systems.

Typically, a directory called 'pub' is where the interesting things are stored. Some sites will have a file with a name like ls-lR, that contains a complete list of the files on that site. Otherwise, you can type ls -lR and get such a listing - for some sites, this can take a LONG time (the size of the resulting file can be anywhere between approximately 2000 bytes and 25Mb).

When retrieving non-text files, you must use binary mode, otherwise the file gets messed up. To do this, use the 'binary' command. (It's safe to set this for text files, but the result might look a bit different from an ASCII transfer)

If the site at the other end is non-Unix, you may need to use some other mode -- see the documents or README files for that site and for FTP (common other modes, are LZ for VAX Multinet servers, tenex or image for some others).

Notes

The simplest way to initiate FTP would be to give the command 'ftp <system-name>'. The <system-name> is the remote system you are connecting to, either a name like garbo.uwasa.fi, if you have an entry in /etc/hosts or are accessing a Domain Name Server or the Internet address 193.166.120.5 for Garbo. If that last sentence doesn't seem to make sense just try: ftp garbo.uwasa.fi or ftp 193.166.120.5 and look what happens. After a short wait, you will be prompted for your username. If you do not have an account on the remote system, some systems allow you to use 'anonymous'. This gives you a restricted access path (meaning that you can only run certain commands like 'dir' or 'ls' and are allowed only access to certain directories like 'pub').

You would then be prompted for a password. Some systems will tell you to send your real identity as the password. What you type doesn't matter in most cases, but it is suggested to give your e-mail address. This as a courtesy to the archive maintainers, who would like to know who's using their system. Other systems need a password of 'guest', or something similar.

DO NOT TYPE A PASSWORD THAT YOU USE ON YOUR OWN SYSTEM!

After that, you should receive the FTP prompt (usually ftp>) and have access. You can get a directory of files by giving a 'dir' command. If the remote system is Unix-based and dir does not work, try 'ls -l' for an MS-DOS like output.

On Garbo, there is a file available in the default anonymous ftp directory that explains what Garbo is and where files are located. Look for 00-index.txt or README files or some similar name.

Unix systems will all have the same directory structure, and moving around is done with the 'cd' or 'cwd' command. TOPS-20, VAX/VMS, DOS VM/CMS and other systems have a different structure, but movement is still accomplished with the 'cd' command.

VAX/VMS systems have filesystems that show as ALL CAPS and directories can be recognized as filenames ending in .DIR
e.g FAQ.DIR

Files reside on disks, denoted by NAME: e.g. NETDISK:
and a file on that disk could be denoted by:
NETDISK:[FAQ.INTERNET]FTP.FAQ

You can change to that directory by typing:
cd netdisk:[faq.internet]
but since you are generally allowed only access to one disk,
you probably can use cd [faq.internet] or type cd faq and
then cd internet

TOPS-20 has directories of the form: DISK:<DIR1.DIR2>, VAX/VMS
has directories of the form DISK:[DIR1.DIR2] (use cd [-] instead
of cd .. and cd [.DIR1] instead of cd dir1). DOS, OS/2 and
Windows NT look like Unix but have shorter directory names.
VM/CMS has mini-disks that can be accessed with the CD command:

```
cd <vm_userid>.<vaddr>
```

e.g. cd arcdsk.100

For an anonymous userid:

```
cd <userid_of_interest>.<vaddr>  
account <mini-disk_write_password>
```

e.g. cd bob.191
account bob1

Note: 'account...' may not be required if the mini-disk is not
password protected.

Notes

A lot of systems give some information about how to use the system when you login, look for that after you have typed the password (some of those messages will not be shown if you use a - as the first character in your password, some people need this because the system won't recognize them otherwise. If you have problems logging into a site, try a - as the first character).

Different systems have different organizations for their files, and the above example is the way most archives have set it up. By looking around other systems, you can learn how their files are arranged and move around much faster. Note, however, that FTP will not allow you outside the FTP 'root' directory. Moving about the entire system is not permitted. You will get 'Permission denied' messages (or plainly not receiving any message and still not be able to change to the directory).

Usually, files are grouped in archive files, so you don't have to get many small files separately. The most common archival file formats for the Internet are tar and zip. Occasionally, people use shell archives (shar) instead. Tar files are basically a bunch of files 'glued' together. Tar archives can be unpacked by running the 'tar' command on a Unix system (tar exists also for DOS, VMS and a whole bunch of other Operating Systems) -- you may want to first do a 'tar t' on the file to see what it contains before unpacking it. This means typing: 'tar t filename.tar' or 'tar tf filename.tar' and looking what the output shows. To unpack the .tar file, type: 'tar xvf filename.tar', this will create a directory called filename with the unpacked archive in it (no quotes again).

Be careful when unpacking shell archives since they have to be run through the Bourne shell to unpack them. (The simplest way is to use the unshar command).

Files are often stored compressed, because they take up less space that way -- for Unix, the most common compression 'scheme'

Notes

is the 'compress' program, indicated by a .Z suffix on the file name. Also you will find Arc, Zoo, Arj, Lzh, gzipped or Zip files, which are combined archival and compression formats (there are other archival formats as well - talk to the systems staff if you encounter them and don't know how to deal with them). For .zip files use zip and unzip (or pkzip/pkunzip), for .gz files use gzip and for .Z files use compress, which are available for all Operating Systems.

Archival and compression utilities are very handy, but can make it very difficult to 'get' a file and use it:

when you're on a DOS or VMS system for example you can't type:

```
get filename.tar.Z
```

You have to type:

```
get filename.tar.Z filename.tz
```

or something like that and then remember what you have to do to unpack the file, namely first running your version of 'compress' on it and then your version of 'tar'.

Remember this when you can't seem to transfer a file.

An interesting feature of most ftp servers in use today, is the ability to compress and decompress 'on the fly'. This means that when you want to 'get' a .Z file, but you don't have compress handy, you can type: `get filename.Z filename`

The server will then decompress the file and leave you with a plain, uncompressed file. Most servers support on-line decompression of .Z, .gz and .tar files and even 'get'ing an entire directory with 'get directoryname directoryname.tar'. Note that this can take up a huge amount of space and maybe take ages. Make sure you know what you are doing when trying this.

These are the most common file types (there are zillions more):

```
SUFFIX  FTP  TYPE
-----  --  ----
```

Notes

.arc	bin	ARChive (hardly used anymore)
.arj	bin	Arj (mostly MS-DOS)
.gif	bin	Graphics Interchange Format
.gz	bin	GNU Zip (Not compatible with Zip. Found on some sites as .z files. GNU zip is seen in combination with tar as .tgz files, maybe even as .tz files)
.hqx	asc	HQX (Mac, Mac equivalent of uuencode)
.jpg	bin	JPEG (graphics format)
.lzh	bin	LHa, LHarc, Larc
.shar	ascii	SHell ARChive (mostly Unix)
.sit	bin	Stuff-It (Mac)
.tar	bin	Tape ARChive (mostly Unix)
.uu	ascii	uuencode/uuencode (also .uue)
.Z	bin	compress (mostly Unix, seen in combination with tar as .tar.Z files)
.zip	bin	Zip (either PKZip or Zip/Unzip)
.zoo	bin	Zoo

To get a list of all file compression/archiving methods and the programs to uncompress/unarchive (on the PC, Mac, Unix, VAX/VMS, VM/CMS, Atari ST and Amiga systems), FTP to the following site and retrieve the listed file:

ftp.cso.uiuc.edu directory: /pub/doc/pcnet/compression

This could be helpful to people new to FTP that don't know how to unpack the file they have just transferred.

Also check out the Frequently Asked Questions (FAQ) Lists and other periodical postings in the news.answers group. Especially the comp.graphics, comp.compression and the different Operating Systems FAQs (Unix, VMS etc.) can be very handy. Most archiver programs are available from garbo.uwasa.fi in either /pc/arcers or /unix/arcers.

Last but not least, for novices to the Internet, I highly recommend a good book, e.g. 'The Whole Internet User's Guide and Catalog' by Ed Krol. It is written clearly and contains an enormous amount of information. Read it cover to cover, and keep it close at hand. Published by O'Reilly & Associates, it is available from many computer

bookstores or O'Reilly's worldwide distributors.
Contact the publisher at +1 707-829-0515 (USA), or send e-mail to nuts@ora.com for information.

Unfortunately, this book is already outdated, but it still offers an invaluable resource and manual for novices and more experienced Internet users. Apparently there will be a new version of this book somewhere this year, so look out for it.

Other books are 'The Internet Companion', 'Internet: Getting Started', 'Internet Unleashed', 'The Internet Guide for New Users' and a lot more. Look for them in your local bookstore.

Almost all of the information in these books is also available on-line through various documents like the MaasInfo files, Zen and the Art of the Internet, The HitchHiker's Guide to the Internet, The Big Dummy's Guide to the Internet, the on-line version of The.Internet.Companion, the Internet Resource Guide and a whole number of FAQs, RFCs and the like.

The MaasInfo and Big Dummy's Guide files appear to be the most recent of these kind of files (look for info on where to get them in the sitelisting itself or try mailing to netguide@eff.org, that should send you the Big Dummy's Guide in a number of parts). Another source for information might be the magazine Internet World, from Meckler Corp. (info@mecklermedia.com).

I'm not affiliated with any of the publishers, authors or anyone mentioned above, but I bought some of the books and like them.



Caselet

File Transfer

Is there any software to send files from one machine to another just like ftp? But it should have a provision for dialling the telephone number to which my destination system is connected. And after getting connected, I should be able to send my file. In effect, without getting connected to the Internet, I should be able to perform the file transfer operation with an ordinary telephone call. Is this possible? Please explain.

Jayesh Antony Jose

There is software available for this, the most popular being the Norton PC AnyWhere, which allows you to connect to another computer and send files by dialling through the Telephone. ((<http://www.symantec.com/pcanywhere/>)

You can also use an inbuilt utility in your O/S called HYper Terminal. You can find it in Start-Programs-Accessories- Communication-Hyper Terminal.

Source: <http://www.thehindubusinessline.in/ew/2002/01/16/stories/2002011600140404.htm>

Notes

13.7 Summary

File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers. FTP works on a client/server model. The server component is called an *FTP daemon*. The configuration file consists of many configuration parameters. The information about each parameter is available in the configuration file.

13.8 Keywords

FTP: File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers.

Vsftpd: It is a GPL licensed FTP server for UNIX systems, including Linux. It is secure and extremely fast.

13.9 Self Assessment

Fill in the blanks:

1. You can edit the configuration file, /etc/vsftpd.conf, to change the default settings.
2. File Transfer Protocol (FTP) is a TCP protocol for and downloading files between computers.
3. Vsftpd is a GPL licensed FTP server for systems, including Linux. It is secure and extremely fast.
4. FTP itself uses the exclusively.
5. Once the file has been edited to your satisfaction, save the file.
6. With this test you can verify if the error is or server-side.
7. All the normal commands may be used to retrieve files.
8. FTP is a facility offered by many machines on the Internet.
9. FTP (File Transfer Protocol) is the simplest and most way to exchange files over the Internet.
10. When you connect to an FTP server, you must send a
11. The simplest way to initiate FTP would be to give the command
12. Anonymous FTP is the choice of Web sites that need to files with numerous unknown remote users.
13. File Transfer Protocol (FTP) servers are one of the oldest and most reliable methods of file between computers.
14. In the, browse to the location where you would like the file to be stored on the remote computer.
15. An interesting feature of most ftp servers in use today, is the ability to and decompress 'on the fly'.

13.10 Review Questions

Notes

1. Explain the File Transfer protocol. Explain with an diagram.
2. What is the processor to install the FTP server?
3. Discuss the configuration FTP server.
4. Explain the main features of vsftpd.
5. "File Transfer Protocol, or FTP, is a robust." Explain
6. In your opinion File Transfer Protocol (FTP) servers are one of the oldest and most reliable methods of file sharing between computers. Why or why not?
7. How can we Installing and setting up vsftp in a Linux environment?
8. Explain what are the different types of programs used in FTP?
9. What is the processor to access the graphical clients?
10. "The most common use for FTP is to download files from the Internet". Comment

Answers: Self Assessment

- | | | |
|---------------------------|-------------------------|-----------------|
| 1. Vsftpd | 2. uploading | 3. UNIX |
| 4. TCP | 5. vsftpd.conf | 6. local |
| 7. FTP | 8. Anonymous | 9. Secure |
| 10. username and password | 11. 'ftp <system-name>' | |
| 12. Exchange | 13. Sharing | 14. server pane |
| | | 15. compress |

13.11 Further Readings



Books

Brian Ward, *How Linux Works*, No Starch Press.

Christopher Negus, *Linux Bible*, Wiley.

Dee-Ann LeBlanc and Richard K. Blum, *Linux for Dummies*.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, *Linux in a Nutshell*, O'Reilly Media.

Wale Soyinka, *Linux Administration: A Beginner's Guide*, McGraw-Hill Osborne Media.



Online links

<http://www.faqs.org/docs/securing/ftpd.html>

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch15_:_Linux_FTP_Server_Setup

Unit 14: File Server

CONTENTS

Objectives

Introduction

14.1 Overview of Samba Server

14.2 Installing Samba Server

14.2.1 Download and Install Packages

14.2.2 Samba vs NFS

14.3 Starting and Stopping of Samba Server

14.4 Configuring Samba with SWAT

14.5 Starting SWAT Service

14.5.1 Installation and Configuration

14.5.2 Using SWAT

14.6 Creating User Accounts

14.6.1 Mapping Different Usernames

14.6.2 Sharing Network Directories

14.7 Create and Configuring Samba Share on Linux

14.7.1 Creating Samba Share

14.7.2 Configuring SAMBA Share

14.8 Summary

14.9 Keywords

14.10 Self Assessment

14.11 Review Questions

14.12 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss overview of samba server
- Understand installing samba server
- Discuss starting and stopping the samba server
- Discuss samba configuration with SWAT
- Discuss starting SWAT service
- Understand adding samba user
- Discuss creating and configuring Samba share

Introduction

Notes

Samba is a strong network service for file and print sharing that works on the mainstream of operating systems available today. When well implemented by the administrator, it's faster and more secure than the native file sharing services available on Microsoft Windows machines. Samba is the protocol by which a lot of PC-related machines share files and printers, and other information, such as lists of available files and printers. Operating systems that support this natively include Windows 95/98/NT, OS/2, and Linux, and add on packages that achieve the similar thing are available for DOS, Windows, VMS, Unix of all kinds, MVS, and more.

14.1 Overview of Samba Server

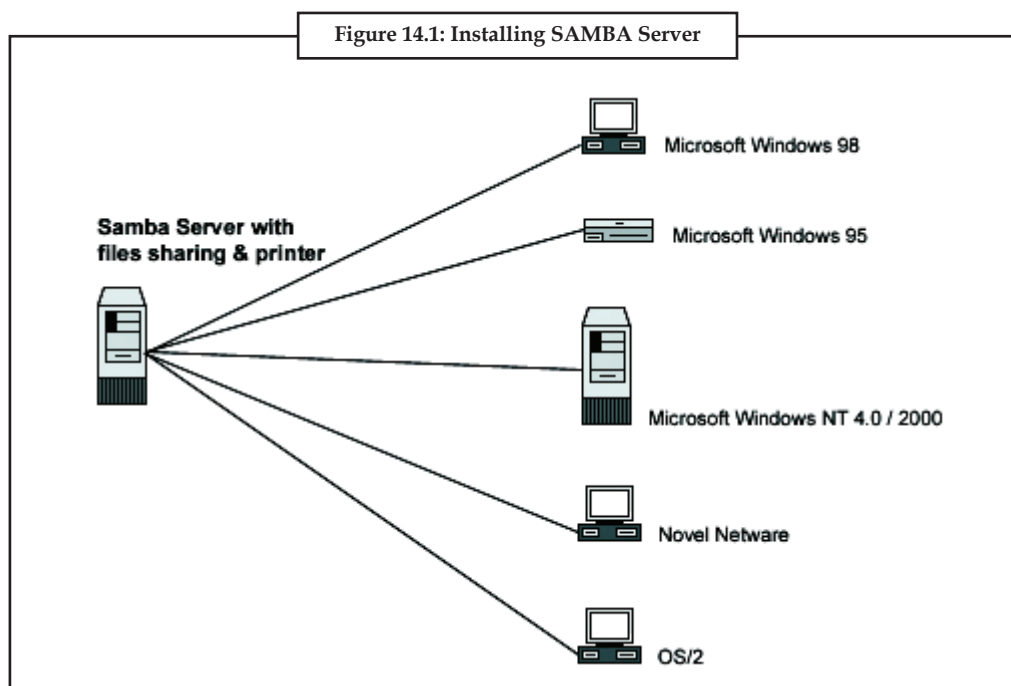
Samba is a networking tool that enables Linux to participate in Windows networks. There are two parts to Samba, one being the server which shares out files and printers for other PC's to use, and the other being the client utilities, which allow Linux to access files and printers on other Windows/Samba PCs.

Apple Macs and some Web Browsers can speak this protocol as well. Alternatives to SMB include Netware, NFS, AppleTalk, Banyan Vines, and Decnet etc. Many of these have advantages but none are public specifications and widely implemented in desktop machines by default. Samba software includes an SMB server, to provide Windows NT and LAN Manager-style file and print services to SMB clients such as Windows 95, Warp Server, smbfs and others, a NetBIOS, rfc1001/1002 name server, which amongst other things gives browsing support, an ftp-like SMB client so that you can access PC resources; disks and printers from Unix, Netware and other operating systems, and finally, a tar extension to the client for backing up PCs.



Did u know? What is SMB?

14.2 Installing Samba Server



Notes

These installation instructions assume


1. The Commands are Unix-compatible.
2. The source path is /var/tmp, other paths are feasible.
3. Installations were tested on Red Hat Linux 6.1 and 6.2.
4. All steps in the installation will happen in super-user account root.
5. Samba version number is 2.0.7

Samba is a suite of utilities that let your Linux box to share files and other resources, such as printers, with Windows boxes. This chapter describes how you can make your Linux box into a Windows Primary Domain Controller (PDC) or a server for a Windows Workgroup. Either configuration will allow everyone at home to have:

1. They have logins on all the home windows boxes whereas having their files on the Linux box appear to be located on a new Windows drive
2. Shared access to printers on the Linux box
3. Shared files accessible simply to members of their Linux user group.

Linux functionality doesn't vanish when you do this. Samba Domains and Linux share the same usernames so you can log into the Samba based Windows domain using your Linux password and immediately gain access to files in your Linux user's home directory. For added security you can make your Samba and Linux passwords different.

When it starts up, and with every client demand, the Samba daemon reads the configuration file /etc/samba/smb.conf to conclude its various modes of operation. You can make your own smb.conf using a text editor or the Web-based SWAT utility which is easier. Keep in mind, however, that if you create /etc/samba/smb.conf with a text editor then subsequently use SWAT to edit the file, you will lose all the comments you inserted with the text editor.



Task Write the configuration file to conclude its various modes of operation.

14.2.1 Download and Install Packages

Samba can do for you and your users, it's time to get your own network set up. Let's start with the installation of Samba itself on a Unix system. When dancing the samba, one learns by taking small steps. It's just the same when installing Samba; we need to teach it step by step. For illustrative purposes, we will be installing the 2.0.4 version of the Samba server on a Linux system running version 2.0.31 of the kernel. However, the installation steps are the same for all of the platforms that Samba supports. A typical installation will take about an hour to complete, including downloading the source files and compiling them, setting up the configuration files, and testing the server.

If you haven't heard of Linux yet, then you're in for a treat. Linux is a freely distributed Unix-like operating system that runs on the Intel x86, Motorola PowerPC, and Sun Sparc platforms. The operating system is relatively easy to configure, extremely robust, and is gaining in popularity.

On the other hand, if you want to download the latest version, the primary web site for the Samba software is <http://www.samba.org>. Once connected to this page, you'll see links to several Samba mirror sites across the world, both for the standard Samba web pages and sites devoted exclusively to downloading Samba. For the best performance, choose a site that is closest to your own geographic location.

The standard Samba web sites have Samba documentation and tutorials, mailing list archives, and the latest Samba news, as well as source and binary distributions of Samba. The download sites (sometimes called F T P sites) have only the source and binary distributions. Unless you specifically want an older version of the Samba server or are going to install a binary distribution, download the latest source distribution from the closest mirror site. This distribution is always named:

```
samba-latest.tar.gz
```

If you choose to use the version of Samba that is located on the CD-ROM packaged with this book, you should find the latest Samba distribution in the base directory.

Most RedHat and Fedora Linux software products are accessible in the RPM format. Downloading and installing RPMs isn't hard.

Samba is consisting of a suite of RPMs that come on the Fedora CDs. The files are named:

1. samba
2. samba-common
3. samba-client
4. samba-swat

When searching for the file, keep in mind that the RPM's filename usually starts with the RPM name followed by a version number as in `samba-client-3.0.0-15.i386`.



Did u know? Samba is consisting of a suite of RPMs that come on the Fedora CDs.

14.2.2 Samba vs NFS

Samba Server

1. Windows can connect to it natively with no installing of third-party software.
2. Support in network browsing. With or without NetBIOS.
3. Acts as a Windows NT®-style Primary Domain Controller (PDC).
4. Acts as a Backup Domain Controller (BDC) for a Samba-based PDC.
5. Acts as an Active Directory domain member server.
6. Joins a Windows NT/2000/2003 PDC.

NFS Server

1. Quicker than Samba while transferring large files over the network.
2. Simple for setup on any Linux system.
3. Supports Kerberos for authentication.
4. Has the capability to squash root so clients accessing the shares cannot access root files.
5. Uses IP-addresses to authenticate the client. And, is thus easier to setup.



Task

Explain the difference between the Samba server and NFS server.

14.3 Starting and Stopping of Samba Server

Samba basically consists of two or three daemons. A daemon is a UNIX application that runs in the background and provides services. An example of a service is the Apache Web server for which the daemon is called httpd. In the case of Samba there are three daemons, two of which are needed as a minimum.

The Samba server is made up of the following daemons:

1. **nmbd:** This daemon handles all name registration and resolution requests. It is the main vehicle involved in network browsing. It handles all UDP-based protocols. The nmbd daemon should be the first command started as part of the Samba startup process.
2. **smbd:** This daemon handles all TCP/IP-based connection services for file- and print-based operations. It also controls local authentication. It should be started immediately following the startup of nmbd.
3. **winbindd:** This daemon supposed to be started when Samba is a member of a Windows NT4 or ADS domain. It is also needed when Samba has trust relationships with another domain. The winbindd daemon will check the smb.conf file for the presence of the idmap uid and idmap gid parameters. If they are are found, winbindd will use the values specified for for UID and GID allocation. If these parameters are not specified, winbindd will start but it will not be able to allocate UIDs or GIDs.

When Samba has been packaged by an operating system dealer, the startup process is typically a custom feature of its integration into the platform as a whole. Please refer to your operating system platform administration manuals for specific information pertaining to correct management of Samba startup.

4. **SWAT:** SWAT is a Web-based interface that can be used to ease the configuration of Samba. SWAT might not be available in the Samba package that shipped with your platform, but in a separate package. If you need to build SWAT please read the SWAT man page regarding compilation, installation, and configuration of SWAT from the source code.



Did u know? What is nmbd?

To launch SWAT, just run your preferred Web browser and point it to <http://localhost:901/>. Replace localhost with the name of the computer on which Samba is running if that is a different computer than your browser.

SWAT can be used from a browser on any IP-connected machine, however be aware that connecting from a remote machine leaves your connection open to password sniffing because passwords will be sent over the wire in the clear.

To start a Samba server, type the following command in a shell prompt while logged in as root:

```
/sbin/service smb start
```

To set up a domain member server, you must first join the domain or Active Directory using the net join command before starting the smb service.

To stop the server, type the following command in a shell prompt while logged in as root:

```
/sbin/service smb stop
```

The *restart* option is a quick way of stopping and then starting Samba. This is the most reliable way to make configuration changes take effect after editing the configuration file for Samba. Note that the restart option starts the daemon even if it was not running originally.

To restart the server, type the following command in a shell prompt while logged in as root:

```
/sbin/service smb restart
```

The *condrestart* (*conditional restart*) option only starts *smb* on the condition that it is currently running. This option is useful for scripts, because it does not start the daemon if it is not running.

When the *smb.conf* file is changed, Samba automatically reloads it after a few minutes. Issuing a manual restart.

To conditionally restart the server, type the following command as root:

```
/sbin/service smb condrestart
```

A manual reload of the *smb.conf* file can be useful in case of a failed automatic reload by the *smb* service. To ensure that the Samba server configuration file is reloaded without restarting the service, type the following command as root:

```
/sbin/service smb reload
```

By default, the *smb* service does *not* start automatically at boot time. To configure Samba to start at boot time, use an initscript utility, such as */sbin/chkconfig*, */usr/sbin/ntsysv*, or the *Services Configuration*



Task

Write the two commands to start the Samba server.

You can start or stop the Samba server automatically (during boot) or manually. Starting and stopping policy is a part of the YaST Samba server configuration. To stop or start running Samba services with YaST, use System+System Services (Runlevel) and check *winbind*, *smb*, and *nmb*. From a command line, stop services required for Samba with *rcsmb stop && rcnmb stop* and start them with *rcnmb start && rcsmb start*; *rcsmb* cares about *winbind* if needed.

14.4 Configuring Samba with SWAT

Samba's configuration is stored in the *smb.conf* file, which generally resides in */etc/samba/smb.conf* or */usr/local/samba/lib/smb.conf*. You can either edit this file yourself or do it using one of the many graphical tools that are available, such as the Web-based interface SWAT, that is included with Samba.

Configuration File Syntax

The *smb.conf* file uses the similar syntax as the various old *.ini* files in Windows 3.1: Each file consists of various sections, which are started by putting the section name between brackets ([]) on a new line. Each contains zero or more key/value pairs separated by an equality sign (=). The file is just a plaintext file, so you can open and edit it with your favorite editing tool.

Each section in the *smb.conf* file symbolizes either a share or a meta-service on the Samba server. The section *[global]* is special, since it contains settings that apply to the whole Samba server. Samba supports a number of meta-services, each of which serves its own purpose. For example, the *[homes]* share is a meta-service that causes Samba to provide a personal home share for each user. The *[printers]* share is a meta-service that establishes print queue support and that specifies the location of the intermediate pool directory into which print jobs are received from Windows clients prior to being dispatched to the UNIX/Linux print spooler.

Notes

Notes

The printers meta-service will origin every printer that is either specified in a printcap file, via the lpstat, or via the CUPS API, to be published as a shared print queue. The printers stanza in the smb.conf file can be set as not browseable. If it is set to be browseable, then it will be visible as if it is a share. That makes no sense given that this meta-service is responsible only for making UNIX system printers available as Windows print queues. If a comment parameter is specified, the value of it will be displayed as part of the printer name in Windows Explorer browse lists.

Each section of the smb.conf file that states a share, or a meta-service, is called a stanza. The global stanza specifies settings that affect all the other stanzas in the smb.conf file. Configuration parameters are documented in the smb.conf man page. Some parameters can be used only in the global stanza, some only in share or meta-service stanzas, and some can be used globally or just within a share or meta-service stanza.

A minimal smb.conf contains a very minimal smb.conf.

Example of a minimal smb.conf

```
[global]
workgroup = WKG
netbios name = MYNAME

[share1]
path = /tmp

[share2]
path = /my_shared_folder
comment = Some random files
```

The configuration file for Samba is comparatively easy to interpret as there are ample comments throughout the file for guidance. There are also some very well documented man pages available to further assist with your configuration requirements.

The main Samba configuration file should be backed up before any settings are changed, so we can at least restore a good file when problems occur.

```
[bash]# cp /etc/samba/smb.conf /etc/samba/smb.conf.original [bash]# vi /etc/samba/smb.conf
```

The configuration file actually has one main section. The [global] section and its directives provide all the options and parameters required for the Samba daemon (smbd) and NetBIOS daemon (nmbd) to operate within the network. This (in a nutshell) is how your server will operate and be seen on the network.

All other sections of the smb.conf file which are specified with square brackets “[something]”, is start of a share definition and enclose all of the options and parameters that pertain only to the resource that is being shared. Any directives that are specified within a share will override any directives that are specified in the global section.

The following directives describe the start of the global configuration options and more importantly, provides the options that identify the server on the network, they are the names for your networking environment.



Example:

```
[global]
workgroup = WORKGROUP
```



```
netbios name = GALAXY
server string = Samba Server
```

These directives are networking orientated and classify which networking interfaces to operate on and which subnetworks are allowed to connect to your server. These are important to specify so the server is protected from any possible connections that are attempted from the external network.



Example:

```
interfaces = eth1 lo
hosts allow = 192.168.1. 127.0.0.1
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

You should indicate a log file for the server. Here the “%m.log” definition means that each workstation connecting to the server will have its own log file.



Example:

```
log file = /var/log/samba/%m.log
max log size = 5
```

Some windows clients before Windows 98 and Windows NT (SP3) do not support encrypted passwords, you may require to adjust this if you are using very old Microsoft clients.

Samba will store passwords in encrypted format by default.



Example:

```
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
```

The following options detail how your Samba server will behave on your network. The “security” directive concludes whether the server will function as a Windows Domain Controller or as a simple standalone server for a peer-to-peer system, the “user” option is the default mode. The “wins support” option tells the server to run its own WINS server for name resolution, this is typical of many Microsoft networks. “dns proxy” is not required for a small network and can be disabled.

The remaining settings determine if the server will present itself as a master browser on the network, this will cause the server to participate in network browser elections and attempt to become the master browser.



Example:

```
security = user
local master = Yes
os level = 33
domain master = Yes
preferred master = Yes
wins support = Yes
dns proxy = No
```

Notes

```
passwd backend = smbpasswd
passwd expand explicit = No
```

The mask directives determine the local file permissions for any new files or directories that are formed in any of the shared resources. These global values can be overridden for each share, allowing for finer control of permissions.



Example:

```
create mask = 0644
directory mask = 0755
```

Using all of the above configurations, the Samba server will now be configured to run on the suitable network and can be seen from your Windows based clients, however no network shares or user access has been granted yet.

Samba offer a small test application that reads the configuration file and tests it for any errors, this ensures that the new configuration should be excepted by the server. Any errors should be fixed before restarting the server and loading the faulty configuration.

The following testparm output is from a configuration file that has only the [global] section (as per all of the above settings), no other share sections have yet been defined.



Example:

```
[bash]# testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

The appropriate runlevels that the service will operate at can now be set and checked.



Example:

```
[bash]# chkconfig --level 345 smb on
[bash]# chkconfig --list smb
```

All of the global section options have now been specified for your configuration and the server can now be started. Be sure to check all the log files to ensure the server started cleanly and there are no errors.



Example:

```
[bash]# /etc/init.d/smb restart
[bash]# tail -n 50 /var/log/messages
[bash]# cat /var/log/samba/smbd.log
[bash]# cat /var/log/samba/nmbd.log
```

If any of the server's networking parameters have been attuned, it may take a few minutes before the server can be seen from the Windows client after restarting the smb service.

Both of the smbd and nmbd daemons are happening by the initscripts. Any command line options that need to be specified for the daemons can be listed in the "/etc/sysconfig/samba" file.

```
[bash]# vi /etc/sysconfig/samba
```

14.5 Starting SWAT Service

If you need to share files or printers with Windows machines, you're likely using Samba and know how to administer and configure it by editing configuration files and starting and stopping the daemon. However, there's an easier, graphical way to configure your box: the Samba Web Administration Tool.

SWAT allows you to set up all aspects of your Samba server through an intuitive Web interface in a style similar to Webmin. (In fact, if you use Webmin, you can access SWAT by going to Servers -> Samba Windows File Sharing and then clicking on the hammer icon for SWAT.)

Among its interesting features, SWAT includes a wizard that can help you configure Samba quickly with a basic setup. It also provides context-sensitive help for all parameters (taken directly from the man pages), and it lets you monitor the current state of connections and users. Since it is an integral part of the Samba suite, all available parameters are always up to date.

While SWAT offers convenience, it won't help you learn about Samba. You need to know about Samba before you use SWAT; if you don't know what you're doing, you're likely to do more harm than good.

SWAT allows you to set up all aspects of your Samba server through an intuitive Web interface in a style similar to Webmin.



Did u know? What are the uses of SWAT?

14.5.1 Installation and Configuration

SWAT comes with Samba, so unless you're running a really old version of Samba, you will already have it installed. You can check its availability, though, by using the `swat --help` command. You can also use the following commands to find the program:

```
# whereis swat swat: /usr/sbin/swat /usr/share/man/man8/swat.8.gz # find / -name swat /
etc/webmin/samba/swat /etc/xinetd.d/swat /usr/sbin/swat /usr/share/samba/swat
```

The `/usr/sbin/swat` file is the SWAT executable program itself. (I'm using openSUSE 10.3. In other distributions you might find SWAT somewhere else.) You will have the (root access only) `/etc/webmin/samba/swat` file only if you have Webmin installed; this is where Webmin keeps your Samba user and password. Finally, you must edit `/etc/xinetd.d/swat` so that SWAT will be available through xinetd. Working as root, edit it so it looks like this:



Example:


```
# SWAT is the Samba Web Administration Tool.
```

```
service swat
{
    port          = 901
    groups       = yes
    socket_type   = stream
    protocol     = tcp
    wait         = no
    user         = root
```

Notes

```

server      = /usr/sbin/swat
only_from   = 127.0.0.1
log_on_failure += USERID
disable     = no
    
```



Note In the server line, I put the path I found earlier with whereis or find. It's likely that the last line will read disable=yes because many distributions don't enable SWAT by default. My version of this file was missing the port and groups lines, so I had to add them. When you're done editing the file, run /etc/init.d/xinetd restart, and SWAT will be ready to work.

On older systems, you might have inetd instead of xinetd; in that case, look for /etc/inetd.conf, which should include an entry like this:

```

# swat is the Samba Web Administration Tool
swat stream tcp nowait.400 root /usr/sbin/swat swat
    
```

Use /etc/init.d/inetd restart to activate your edits.

14.5.2 Using SWAT

SWAT works by rewriting the configuration files at /etc/samba/smb.conf and restarting Samba as needed. Unsupported parameters are deleted, parameters that have the default value are ignored, and SWAT changes the order of the parameters, so your carefully handcrafted configuration file may look rather bleak after SWAT is done with it.



Access SWAT by opening a browser and going to <http://127.0.0.1:901>. The home page provides access to the man documentation pages and some extra Samba documentation, including some complete books. You'll have to install the samba-doc additional package to get the latter.



Caution When it rewrites the file, it wipes away all comments, so if you're the kind of sysadmin who likes to include documentation within configuration files, you'll have a reason to hate SWAT.

Here's a list of SWAT's other pages and the functions you can use:

1. **Globals:** Provides access to the global parameters (what else?) in smb.conf. You can work at one of two levels: Basic, which shows only the more important options listed, or Advanced, which shows every available parameter. Click on the corresponding buttons to pick your desired level. You can click on the Help links to get specific help about each parameter. After making any edits, click on Commit Changes to save your choices, or Reset Values to go back to the original smb.conf values.
2. **Shares:** Lets you create, edit, or drop shares. To edit an existing share, pick it from the combo box, then click on Choose Share; clicking on Delete Share will delete it. When you add or edit a share, you can specify several parameters on either Basic or Advanced levels, such as name, path, valid and invalid users, and read-only or writable. Remember to click on Commit Changes (or Reset Values) before changing pages or your work will have been in vain.
3. **Printers:** Similar to Shares, but works with printers instead.
4. **Wizard:** Lets you do a quick server configuration. You can pick either a standalone server, a domain controller, or a domain member. You must also specify how the server will work with Windows Internet Name Service (WINS) and whether you want to expose your home directories. Click on "Rewrite smb.conf" or on Commit to save your values, and you will have a basic Samba server running, which you can then further tweak using the other options.
5. **Status:** Shows you which services are running: usually smbd (the Samba daemon itself) and nmbd (the NetBIOS nameserver support daemon), and possibly Winbind (to allow a Linux box to become a Windows domain member). You also get the listing of all active connections (which you may kill, if you need to), active shares, and open files. You can click on Auto Refresh so the page will refresh on its own every so many seconds (30 by default).
6. **View:** Lets you view the current configuration file. You can click on the View button to see it either in the normal view (minimalist, with only the nondefault attributes) or the full view (with all possible parameters).
7. **Password:** Allows you to create, delete, enable, or disable local Samba users and change passwords for a local or remote server.



Note Whether you're just starting out with Samba or are an experimented sysadmin, SWAT can help you configure your box more easily through its graphical interface.

14.6 Creating User Accounts

Before the Samba server can be accessed across the network, access must be granted to users and any shared resources which are going to be provided by the server. Essentially, for a user to be granted access to the server they need to have a valid UNIX account and a separate Samba password which is stored in the "smb password file", so in fact a user's password for their UNIX account may be different to their Samba account.

Notes

In the following example a UNIX account is being produced for Alice. It details her full name (-c), her home directory (-d) and prevents her from logging into the Linux server by allocating her a false shell (-s). This account type will only allow Alice to access the server via the Samba network. This entry is located in the /etc/passwd file.

```
[bash]# useradd -c "Alice Jones" -d /home/alice -s /sbin/nologin alice
```

Alice now requests to be allocated a password for accessing the Samba server. Remember, this is a separate password to her UNIX account any may be different if necessary.

```
bash]# smbpasswd -a alice
```

```
New SMB password:
```

```
Retype new SMB password:
```

```
Added user alice
```

The above command adds (-a) an entry into the "smb password file" and encrypts the password. Type "smbpasswd alice" to only change passwords.

Alice can now access the server over the network, though there are still no shares defined.

```
[bash]# grep alice /etc/samba/smbpasswd
```

```
alice:4732:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805F797BF2A82807973B89537:[
U          ]:LCT-41CFEFD8:
```

If Alice's account needs to be deleted, the following command can be used. Otherwise her account can be disabled (-d) or enabled (-e) as required.

```
[bash]# smbpasswd -x alice
```

```
Deleted user alice.
```

14.6.1 Mapping Different Usernames

There may be a requirement wherever the samba username being used to access the server does not match the same UNIX account username, or you would like to force a change between the two different account types. This can easily be done by implementing the "username map" directive into the [global] section of the main configuration file.

```
[bash]# vi /etc/samba/smb.conf
```

```
[global] username map = /etc/samba/smbusers
```

The username map characteristic is fairly simple, the file takes a UNIX account name on the left hand side and Samba account names on the right hand side (separated by "="). The username map allows those NT accounts listed on the RHS to be granted the access rights and file permissions of the UNIX account on the LHS when they connect to a resource.

In the following example:

1. The NT usernames "administrator" and "admin" will be mapped to the UNIX "root" account,
2. The NT usernames "guest", "pcguest" and "smbguest" will be mapped to the UNIX "nobody" account,
3. The NT username "alice" will be mapped to the UNIX "alice.jones" account,
4. All four NT Users (glen, fred, terry and sarah) will be mapped to the single UNIX "readonly" account, and
5. The NT username "Lachlan Smith" will be mapped to the UNIX "lachlan" account.

The last example uses quotes about the NT username because there is a space separating the user's first and last names. Failure to use quotes on an NT username containing a space means that Samba will treat the user's name as two separate UNIX accounts; this will cause the connections to fail.

```
[bash]# vi /etc/samba/smbusers
# Unix_name = SMB_name1 SMB_name2 ...
root= administrator admin
nobody = guest pc guest smb guest
alice.jones = alice
readonly = glen fred terry sarah
lachlan = "Lachlan Smith"
```

Further details on mapping usernames can be get in the configuration man page, type "man smb.conf".

14.6.2 Sharing Network Directories

The major purpose of setting up a Samba server is to provide networked resources to your Microsoft workstations and clients, so let's set up some resources for them to connect to and use. Shared resources are specified as sections within the `/etc/samba/smb.conf` file, the sections are identified by using squared brackets around each of the section names, similar to the global section.

```
[bash]# vi /etc/samba/smb.conf
```

The first share that can be configured are the home directories for all the connecting users. The `[homes]` section is a unusual configuration and Samba already knows how to handle all the users different home directories, it really only needs to be specified and Samba will do the rest.



Example:

```
[homes]
comment = Home Directory
read only = No
browseable = No
valid users = %S
```

The `[Shared]` section that we have formed below allows access to all of the files and directories within the `/samba/shared` local directory. The resource can be written to by all guest and public users and the resource can be viewed (browseable) by workstations and clients on the network.

Any new directory created in the share will be specified to the directory permissions of `777`, and any new file will have file permissions of `666`. These mask settings allow any user to save files to the shared directory, and any other user can read, write or delete the files.



Example:

```
[Shared]
comment = Global Share - All Users
path = /samba/shared
```

Notes

```
read only = No
guest ok = Yes
public = Yes
browseable = Yes
create mask = 0666
directory mask = 0777
```

The [SmallGroup] example section is not allowable to have guest or public access, but the resource can be viewed by networked workstations and clients.

Some new directory created in the share will be given the directory permissions of 777, and any new file will have file permissions of 666. The only valid user accounts that can connect to this resource are peter, paul, and mary.



Example:

```
[SmallGroup]
comment = Small Share - Few Users
path = /samba/smallgroup
read only = No
guest ok = No
browseable = Yes
create mask = 0666
directory mask = 0777
valid users = peter, paul, mary
```

The [Financial] example resource can be viewed by all networked workstations and clients but is not publicly accessible. Some new directory created in the share will be given the directory permissions of 770, and any new file will have file permissions of 660.

The only valid users that are allowed to access this resource are UNIX user accounts planned in the UNIX group file (/etc/groups) called "financial". This is specified by the "@financial" parameter.

Any file or directory that is produced on the shared resource will have the (forced) group name of "financial" applied to it, this is similar to typing "chgrp financial *" at the Linux command prompt.



Example:

```
[Financial]
comment = RESTRICTED - Financial Cell
path = /samba/financial
read only = No
guest ok = No
browseable = Yes
create mask = 0660
directory mask = 0770
```



```
valid users = @financial
force group = financial
```

Below is an example [FTP-Server] resource which is mapped to the root of the FTP Server (/var/ftp) running on the local Samba server. The share has been configured so it is publicly accessible to everybody on the network, but the filesystem is read only. All new files and directories will have the file permissions of 755.

The “write list” directive overrides the “read only” directive, which means in this example that the two users (john and fred) can completely manage all the files and resources like a normal share.

The “force group” and “force user” directives states that any new files or directories will be processed as belonging to the UNIX ftp user account. This is similar to typing “chown ftp.ftp **” at the command prompt and also makes the files readily accessible by the FTP server.



Example:

```
[FTP-Server]
comment = READ ONLY - Corp FTP Server
path = /var/ftp
read only = Yes
guest ok = Yes
browseable = Yes
create mask = 0755
directory mask = 0755
write list = john, fred
force group = ftp
force user = ftp
```

The [WEB-Server] example share is mapped to the “document root” (/var/www/html) of the Apache web server running on the local Samba server. The “browseable = No” directive informs the Samba server not to tell any networked workstations and clients that the resource is available, this requires that a connecting client must already know the resource is shared as “WEB-Server”. In effect the resource is available to the valid users but is hidden from view.

The filesystem has also been marked as read only but can be fully administered by the UNIX user account called fred. All files and directories written to the networked share will be forced to belong to the root group and user accounts.

This is a quite powerful share as the effective user (fred) will have root privileges to the filesystem located under the /var/www/html directory. It also allows fred (the local webmaster) to add or update any web pages as required.



Example:

```
[WEB-Server]
comment = HIDDEN - Corp Web Server
path = /var/www/html
read only = Yes
guest ok = No
```

Notes

```
browseable = No
create mask = 0644
directory mask = 0755
write list = fred
force group = root
force user = root
```

Before any configuration changes are implemented, the configuration file should be tested to make sure it is free from any errors and that the new configuration will be accepted by the server. Any errors should be fixed before restarting the server and loading the faulty configuration.



Example:

```
[bash]# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[Shared]"
Processing section "[SmallGroup]"
Processing section "[Financial]"
Processing section "[FTP-Server]"
Processing section "[WEB-Server]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

14.7 Create and Configuring Samba Share on Linux

The following procedure provides an example of how to create and configure a Samba share. For more detailed information about configuring a Samba share, refer to the Samba documentation.

14.7.1 Creating Samba Share

1. On the Linux server, become the root user.
2. In a terminal window, type `groupadd -r groupname` to create a Linux group (this is the group name that users of QuickBooks will belong to), where `groupname` is the name of the group you want to create (for example, `qbusers`).
3. Add the line `groupname: user1, user2, user3` to the `/etc/group` file to list the users that will be part of the group you created in step 2.
4. Type `useradd user` to add each user you specified in step 3 that will be accessing QuickBooks company files stored on the share directory.
5. For each user you specified in step 4, type `smbpasswd -a user` to activate the Samba user account and set a password.
6. Type `chmod -R 775 /directory` to provide the users read/write/execute permissions to the share directory.

7. Type `chgrp -R groupname /directory` to change the group ownership for the share directory.
8. Edit the `smb.conf` file to include the following lines. By default, this file is located in `/etc/samba`.



Example:

```
[share_name]
path = /directory
comment = samba share for company files
valid users = user1 user2 user3
public = no
writable = yes
printable = no
create mask = 0765
```

9. Replace `share_name` with the name you want to use for the share (this is the name that your Windows clients can see).
10. Replace `directory` with the full path of the directory you want to configure as the Samba share (the directory you created on your Linux server to store the QuickBooks company files).
11. Type `service smb restart` to restart the Samba daemon.

14.7.2 Configuring SAMBA Share

You may configure the SAMBA server by editing the `/etc/samba/smb.conf` file to change the default settings or add new settings. More information about each setting is available in the comments of the `/etc/samba/smb.conf` file or by viewing the `/etc/samba/smb.conf` manual page from the prompt with the following command typed at a terminal prompt:

```
man smb.conf
```



Note Prior to editing the configuration file, you should make a copy of the original file and protect it from writing so you will have the original settings as a reference and to re-use as necessary.

Backup the `/etc/samba/smb.conf` file:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.original
```

Now, edit the `/etc/samba/smb.conf` file and make your changes.

Server

In addition to the SAMBA suite of file and printer sharing server applications, Ubuntu also includes other powerful server applications designed to provide additional network server functionality to Windows clients similar to the functionality provided by actual Windows servers. For example, Ubuntu offers centralized management of network resources such as computers and users via

Notes

Directory Services, and facilitates the identification, and authorization of computers and users via Authentication Services.

The following sections will discuss SAMBA and the supporting technologies, such as Lightweight Directory Access Protocol (LDAP) server, and Kerberos authentication server in more detail. You will also learn about some of the available configuration directives available the SAMBA configuration file which facilitate network integration with Windows clients and servers.

Active Directory

Active Directory is a proprietary implementation of Directory Services by Microsoft, and is used to provide a means to share information about network resources and users. In addition to providing a centralized source of such information, Active Directory also acts as a centralized authentication security authority for the network. Active directory combines capabilities traditionally found in separate, specialized directory systems to simplify integration, management, and security of network resources. The SAMBA package may be configured to use Active Directory services from a Windows Domain Controller.

LDAP

The LDAP server application provides Directory Services functionality to Windows computers in a manner very similar to Microsoft Active Directory services. Such services include managing the identities and relationships of computers, users, and groups of computers or users that participate in the network, and providing a consistent means to describe, locate, and manage these resources. The freely available implementation of LDAP available for your Ubuntu system is called OpenLDAP. The server daemons responsible for handling OpenLDAP directory requests and the propagation of directory data from one LDAP server to another on Ubuntu, are slapd and slurpd. OpenLDAP may be used in conjunction with SAMBA to provide File, Print, and Directory services in much the same way a Windows Domain Controller does so long as SAMBA is compiled with LDAP support.

Kerberos

The Kerberos authentication security system is a standardized service for providing authentication to computers and users by means of a centralized server which grants encrypted authorization tickets accepted for authorization by any other computer using Kerberos. Benefits of Kerberos authentication include mutual authentication, delegated authentication, interoperability, and simplified trust management. The primary server daemons for handling the Kerberos authentication and Kerberos database administration on Ubuntu are krb5kdc and kadmin. SAMBA may use Kerberos as a mechanism for authenticating computers and users against a Windows Domain Controller. To do so, the Ubuntu system must have Kerberos installed, and the /etc/samba/smb.conf must be modified to select the the proper realm and security mode. For example, edit the /etc/samba/smb.conf file and add the values:

```
realm = DOMAIN_NAME  
  
security = ADS
```

to the file, and save the file.



Note Be sure to replace the token DOMAIN_NAME in the example above with the actual name of your specific Windows Domain.

You will need to restart the SAMBA daemons to effect these changes. Restart the SAMBA daemons with the following command entered at a terminal prompt:

```
sudo /etc/init.d/samba restart
```

Notes

Computer Accounts

Computer Accounts are used in Directory Services to uniquely identify computer systems participating in a network, and are even treated in the same manner as users in terms of security. Computer accounts may have passwords just as user accounts do, and are subject to authorization to network resources in the same manner as user accounts. For example, if a network user, with a valid account for a particular network attempts to authenticate with a network resource from a computer which does not have a valid computer account, depending upon policies enforced on the network, the user may be denied access to the resource if the computer the user is attempting authentication from is considered to be an unauthorized computer.

A computer account may be added to the SAMBA password file, provided the name of the computer being added exists as a valid user account in the local password database first. The syntax for adding a computer or machine account to the SAMBA password file is to use the `smbpasswd` command from a terminal prompt as follows:

```
sudo smbpasswd -a -m COMPUTER_NAME
```



Note

Be sure to replace the token `COMPUTER_NAME` in the example above with the actual name of the specific computer you wish to add a machine account for.

File Permissions

File Permissions define the explicit rights a computer or user has to a particular directory, file, or set of files. Such permissions may be defined by editing the `/etc/samba/smb.conf` file and specifying the explicit permissions of a defined file share. For example, if you have defined a SAMBA share called `sourcedocs` and wish to give read-only permissions to the group of users known as `planning`, but wanted to allow writing to the share by the group called `authors` and the user named `richard`, then you could edit the `/etc/samba/smb.conf` file, and add the following entries under the `[sourcedocs]` entry:

```
read list = @planning
```

```
write list = @authors, richard
```

Save the `/etc/samba/smb.conf` for the changes to take effect.

Another possible permission is to declare administrative permissions to a particular shared resource. Users having administrative permissions may read, write, or modify any information contained in the resource the user has been given explicit administrative permissions to. For example, if you wanted to give the user `melissa` administrative permissions to the example `sourcedocs` share, you would edit the `/etc/samba/smb.conf` file, and add the following line under the `[sourcedocs]` entry:

```
admin users = melissa
```

Save the `/etc/samba/smb.conf` for the changes to take effect.

Notes

Clients

Ubuntu includes client applications and capabilities for accessing network resources shared with the SMB protocol. For example, a utility called `smbclient` allows for accessing remote shared file-systems, in a manner similar to a File Transfer Protocol (FTP) client. To access a shared folder resource known as `documents` offered by a remote Windows computer named `bill` using `smbclient` for example, one would enter a command similar to the following at the prompt:

```
smbclient //bill/documents -U <username>
```

You will then be prompted for the password for the user name specified after the `-U` switch, and upon successful authentication, will be presented with a prompt where commands may be entered for manipulating and transferring files in a syntax similar to that used by non-graphical FTP clients. For more information on the `smbclient` utility, read the utility's manual page with the command:

```
man smbclient
```

Local mounting of remote network resources using the SMB protocol is also possible using the `mount` command. For example, to mount a shared folder named `project-code` on a Windows server named `development` as the user `dlightman` to your Ubuntu system's `/mnt/pcode` mount-point, you would issue this command at the prompt:

```
mount -t smbfs -o username=dlightman //development/project-code /mnt/pcode
```

You will then be prompted for the user password, and after successfully authenticating, the contents of the shared resource will be available locally via the mount-point specified as the last argument to the `mount` command. To disconnect the shared resource, simply use the `umount` command as you would with any other mounted file system. For example:

```
umount /mnt/pcode
```

User Accounts

User Accounts define persons with some level of authorization to use certain computer and network resources. Typically, in a network environment, a user account is provided to each person allowed to access a computer or network, where policies and permissions then define what explicit rights that user account has access to. To define SAMBA network users for your Ubuntu system, you may use the `smbpasswd` command. For example to add a SAMBA user to your Ubuntu system with the user name `jseinfeld`, you would enter this command at the prompt:

```
smbpasswd -a jseinfeld
```

The `smbpasswd` application will then prompt you to enter a password for the user:

New SMB Password

Enter the password you wish to set for the user, and the `smbpasswd` application will ask you to confirm the password:

Retype new SMB Password

Confirm the password, and `smbpasswd` will add the entry for the user to the SAMBA password file.

Groups

Notes

Groups define a collection of computers or users which have a common level of access to particular network resources and offer a level of granularity in controlling access to such resources. For example, if a group `qa` is defined and contains the users `freda`, `danika`, and `rob` and a second group `support` is defined and consists of users `danika`, `jeremy`, and `vincent` then certain network resources configured to allow access by the `qa` group will subsequently enable access by `freda`, `danika`, and `rob`, but not `jeremy` or `vincent`. Since the user `danika` belongs to both the `qa` and `support` groups, she will be able to access resources configured for access by both groups, whereas all other users will have only access to resources explicitly allowing the group they are part of.

When defining groups in the SAMBA configuration file, `/etc/samba/smb.conf` the recognized syntax is to preface the group name with an `@` symbol. For example, if you wished to define a group named `sysadmin` in a certain section of the `/etc/samba/smb.conf`, you would do so by entering the group name as `@sysadmin`.

Group Policy

Group Policy defines certain SAMBA configuration settings pertaining to the Domain or Workgroup computer accounts belong to, and other global settings for the SAMBA server. For example, if the SAMBA server belongs to a Workgroup of Windows computers called `LEVELONE`, then the `/etc/samba/smb.conf` could be edited, and the following value changed accordingly:

```
workgroup = LEVELONE
```

Save the file and restart the SAMBA daemons to affect the change.

Other important global policy settings include the `server string` which defines the NETBIOS server name reported by your Ubuntu system to other machines on the Windows-based network. This is the name your Ubuntu system will be recognized as by Windows clients and other computers capable of browsing the network with the SMB protocol. Additionally, you may specify the name and location of the SAMBA server's log file by using the `log file` directive in the `/etc/samba/smb.conf` file.

Some of the additional directives governing global group policy include specification of the global nature of all shared resources. For example, placing certain directives under the `[global]` heading of the `/etc/samba/smb.conf` file will affect all shared resources unless an overriding directive is placed under a particular shared resource heading. You specify all shares are browseable by all clients on the network by placing a `browseable` directive, which takes a Boolean argument, under the `[global]` heading in the `/etc/samba/smb.conf`. That is, if you edit the file and add the line:

```
browseable = true
```

under the `[global]` section of `/etc/samba/smb.conf`, then all shares provided by your Ubuntu system via SAMBA will be browseable by all authorized clients, unless a specific share contains a `browseable = false` directive, which will override the global directive.

Other examples which work in a similar manner, are the `public` and `writable` directives. The `public` directive accepts a Boolean value and decides whether a particular shared resource is visible by all clients, authorized or not. The `writable` directive also takes a Boolean value and defines whether a particular shared resource is writable by any and all network clients.

Notes



Caselet

Samba Team member John Terpstra has a Samba3 article on InformIT. The case study offers an “example of simple Samba network server architecture.” Small to mid-sized offices looking to migrate to Samba might be interested in this piece:

The office of Abmas Accounting Inc. is a 40-year-old family-run business. There are nine permanent computer users. The network clients were upgraded two years ago. All computers run Windows 2000 Professional. This year the server will be upgraded from an old Windows NT4 server (actually running Windows NT4 Workstation, which worked fine as there were fewer than 10 users) that has run in workgroup (Stand-Alone) mode, to a new Linux server running Samba.

The office does not want a Domain Server. Mr. Alan Meany wants to keep the Windows 2000 Professional clients running as workgroup machines so that any staff member can take a machine home and keep working. It has worked well so far and your task is to replace the old server.

14.8 Summary

Samba is a suite of utilities that allows your Linux box to share files and other resources, such as printers, with Windows boxes. This chapter describes how you can make your Linux box into a Windows Primary Domain Controller (PDC) or a server for a Windows Workgroup. Samba essentially consists of two or three daemons. A daemon is a UNIX application that runs in the background and provides services. The smb.conf file uses the same syntax as the various old .ini files in Windows 3.1: Each file consists of various sections, which are started by putting the section name between brackets ([]) on a new line. Each contains zero or more key/value pairs separated by an equality sign (=). Before the Samba server can be accessed across the network, access must be granted to users and any shared resources which are going to be provided by the server. The main purpose of setting up a Samba server is to provide networked resources to your Microsoft workstations and clients, so lets set up some resources for them to connect to and use. Shared resources are specified as sections within the /etc/samba/smb.conf file, the sections are identified by using squared brackets around each of the section names, similar to the global section.

14.9 Keywords

Configuration Parameters: Configuration parameters are documented in the smb.conf man page.

PDC: Primary Domain Controller

Security Directive: The security directive determines whether the server will function as a Windows Domain Controller or as a simple standalone server

14.10 Self Assessment

Notes

Fill in the blanks:

1. Samba is a suite of utilities that allows your Linux box to and other resources, such as printers, with Windows boxes.
2. Samba's configuration is stored in the file.
3. The main purpose of setting up a Samba server is to provide resources.
4. Before the Samba server can be accessed across the network, access must be granted to users and any
5. The option is a quick way of stopping and then starting Samba.
6. The daemon will check the smb.conf file for the presence of the idmap uid and idmap gid parameters.
7. The NT usernames "administrator" and "admin" will be to the UNIX "root" account.
8. includes client applications and capabilities for accessing network resources shared with the SMB protocol.
9. The section [global] is special, since it contains settings that apply to the whole
10. Samba is a suite of utilities that let your Linux box to share files and other resources, such as, with Windows boxes.
11. Samba offers a small test application that reads the file and tests it for any errors.
12. define a collection of computers or users which have a common level of access to particular network resources.
13. works by rewriting the configuration files at /etc/samba/smb.conf.
14. defines certain SAMBA configuration settings pertaining to the Domain.
15. is a proprietary implementation of Directory Services by Microsoft.

14.11 Review Questions

1. What are the opportunities and threats in the Samba server? Explain in detail.
2. What is the processor to install Samba server? What are the other packages which are needed to install?
3. How to download and install packages in samba server?
4. What are the differences between Samba server and NFS server?
5. "The Samba server is made up of some daemons." Comment
6. How can be the Samba server can be configured?
7. What is the processor to share the network directory?
8. Explain briefly the starting and stopping of SAMBA server.
9. "SWAT works by rewriting the configuration files at /etc/samba/smb.conf and restarting." Comment
10. What is the processor to map the username?

LOVELY PROFESSIONAL UNIVERSITY

Jalandhar-Delhi G.T. Road (NH-1)

Phagwara, Punjab (India)-144411

For Enquiry: +91-1824-300360

Fax.: +91-1824-506111

Email: odl@lpu.co.in

Notes

Answers: Self Assessment

- | | | |
|---------------------|-------------------|----------------------|
| 1. share files | 2. smb.conf | 3. networked |
| 4. shared resources | 5. restart | 6. winbindd |
| 7. mapped | 8. Ubuntu | 9. Samba server |
| 10. Printers | 11. Configuration | 12. Groups |
| 13. SWAT | 14. Group Policy | 15. Active Directory |

14.12 Further Readings



Books

Bharat Bhusan , Understanding Linux , Khanna Publishing , Nai Sarak , New Delhi.

Brian Ward, How Linux Works, No Starch Press.

Christopher Negus, Linux Bible, Wiley.

Dee-Ann LeBlan and Richard K. Blum, Linux for Dummies.

Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love and Arnold Robbins, Linux in a Nutshell, O'Reilly Media.

Wale Soyinka, Linux Administration: A Beginner's Guide, McGraw-Hill Osborne Media.



Online links

http://news.samba.org/users/nine_user/

<http://www.linuxstreet.net/articles/Samba/>