LOVELY PROFESSIONAL UNIVERSITY

OPERATING SYSTEM TOOLS

Edited By
Avinash Bhagat

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

SYLLABUS

Operating System Tools

Objectives: To enable the student to understand technicalities of Linux including installation, troubleshooting, shell programming, file system, managing servers and internet communication.

Sr. No.	Topics
1.	**Installing Linux:** Preparing for the installation, The installation, Root account configuration.
2.	**Red Hat Linux 9 Basics:** Moving around the desktop, Using applications, The file system, hierarchy, Navigating in the file system, Managing Hardware, Configuring the desktop, Managing processes, Managing Users.
3.	**Connecting to the Internet:** Connecting to the Internet, Managing Multiple ISPs and connections, Software and configuration.
4.	**Installing Software:** RPM, its benefits, The RPM command line tool.
5.	**Everyday Applications:** Office applications, Internet applications, Personal information management.
6.	**Everyday Applications:** Multimedia applications, System applications.
7.	**The Shell:** The Shell as a command line interface, types of shell, Built-in programs and external programs.
8.	**The Shell:** Common Shell commands , Special keys and shortcuts, Command line syntax.
9.	**The File System:** The File System, anatomy of a file, File Search Utilities, locate command, find command, GNOME, Navigating file system .
10.	**Servers:** Introduction to DNS, FTP, Apache, DHCP servers.

CONTENT

Unit 1:	Installing Linux *Avinash Bhagat, Lovely Professional University*	1
Unit 2:	Red Hat Linux 9 Basics *Manish Kumar, Lovely Professional University*	19
Unit 3:	Connecting to the Internet *Avinash Bhagat, Lovely Professional University*	51
Unit 4:	Installing Software *Avinash Bhagat, Lovely Professional University*	69
Unit 5:	Everyday Applications-I *Avinash Bhagat, Lovely Professional University*	84
Unit 6:	Everyday Applications-II *Avinash Bhagat, Lovely Professional University*	105
Unit 7:	The Shell *Dr. Manmohan Sharma, Lovely Professional University*	118
Unit 8;	Shell Commands *Manmohan Sharma, Lovely Professional University*	129
Unit 9:	The File System *Manmohan Sharma, Lovely Professional University*	141
Unit 10:	Servers *Manmohan Sharma, Lovely Professional University*	166

Notes

Unit 1: Installing Linux

CONTENTS

Objectives

Introduction

1.1 Preparing for the Installation

1.1.1 Hardware

1.1.2 Hardware Compatibility Lists

1.1.3 Server Design

1.1.4 Dual-Booting Issues

1.2 The Installation

1.2.1 Installing Red Hat Linux

1.2.2 Creating a Boot Disk

1.2.3 Starting the Installation

1.2.4 Welcome of Red Hat Linux

1.3 Summary

1.4 Keywords

1.5 Review Questions

1.6 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain the process of preparing for installation
- Describe the process of installation
- Discuss Root account configuration

Introduction

Linux has developed into an operating system for business, education, and personal productivity. Linux is a UNIX operating system clone which runs on a variety of platforms, especially personal computers with Intel 80386 or better processors. 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 Preparing for the 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.

Notes

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.

1.1.1 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.

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 = 0 x 300), interrupt (IRQ 10), but not the hardware address (00 00 a6 27

Notes

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.

1.1.2 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.

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

1.1.3 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

Notes

(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.

1.1.4 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) file systems. 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.

Task Differentiate between FAT32 and NTFS file systems.

Self Assessment

Fill in the blanks:

1. Red Hat's major product line is ....., which is mostly based on Red Hat's free software distribution, Fedora.
2. ..... offers fields beyond Vendor, Device, and Category.
3. The Linux ..... Compatibility HOWTO provides information on all device types and all major manufacturers.
4. A server installation removes all existing ..... on all installed hard drives.
5. A basic Linux ..... will load and run the installation script.
6. By default, Windows doesn't recognize any of the native Linux .....
7. If you're using an ..... Windows installation, the files on the Windows partition will be read-only.

Notes

1.2 The Installation

Different types of installation methods are available. You can select any one from them.

- ***CD-ROM:*** If you have a CD-ROM drive and the Red Hat Linux CD-ROMs, you can use this method. You will need a boot diskette or a bootable CD-ROM. A PCMCIA driver diskette may also be used.
- ***Hard Drive:*** If you have copied the Red Hat Linux ISO images to a local hard drive, you can use this method. You will need a boot diskette. A PCMCIA driver diskette may also be used.
- ***NFS Image:*** If you are installing from an NFS server using ISO images or a mirror image of Red Hat Linux, you can use this method. You will need a network driver diskette. A PCMCIA driver diskette may also be used. Please note that NFS installations may also be performed in GUI mode.
- ***FTP:*** If you are installing directly from an FTP server, use this method. You will need a network driver diskette. A PCMCIA driver diskette may also be used.
- ***HTTP:*** If you are installing directly from an HTTP (Web) server, use this method. You will need a network driver diskette. A PCMCIA driver diskette may also be used.

1.2.1 Installing Red Hat 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.

1.2.2 Creating a Boot Disk

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

Notes

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

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.

1.2.3 Starting the Installation

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, when the machine starts.

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 it is recommended to 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.

Notes

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

Selecting Your Language

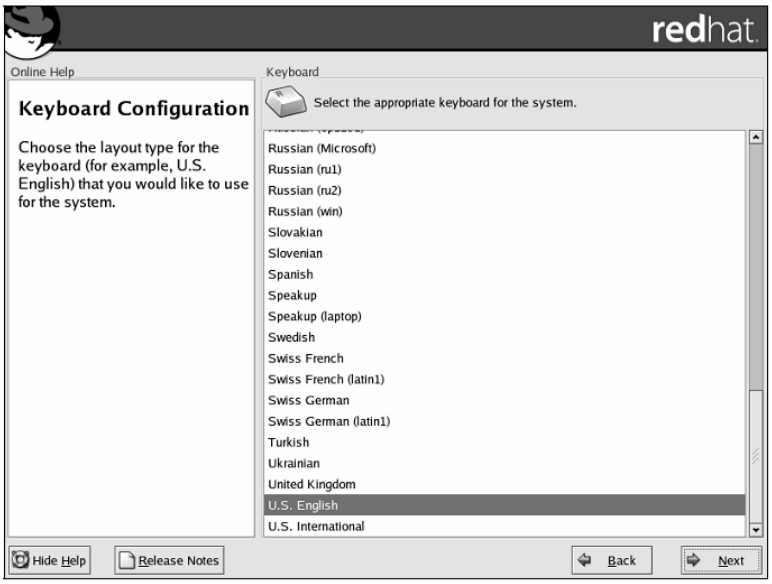
The installation screens are available in more than 30 languages. Select your native tongue from the Language Selection screen shown in Figure 1.1, and click Next.

Figure 1.1: Choosing an Installation Language

Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Select the language of your keyboard from the screen shown in Figure 1.2.

Figure 1.2: Choosing a Keyboard Layout



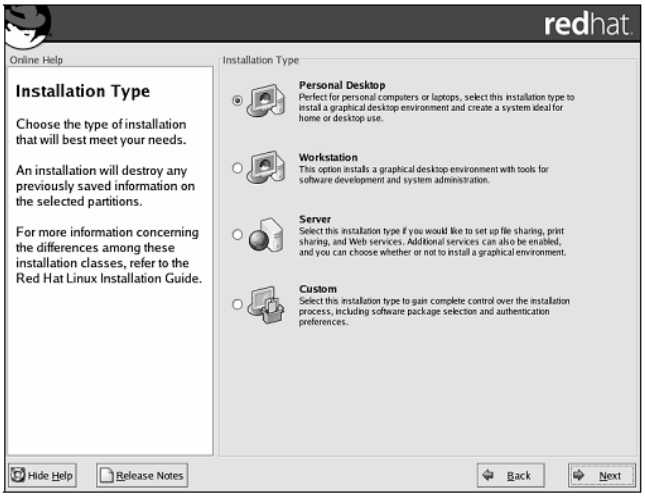
Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Notes

Installation Types

The Red Hat 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 1.3, before clicking Next.

Figure 1.3: Choosing an Installation Type

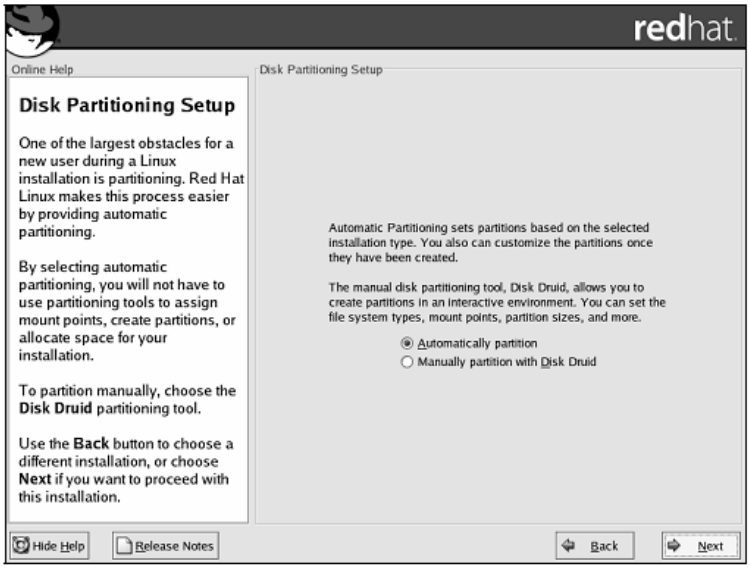


Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Disk Partitioning

This installer offers two partitioning methods – automatic and manual – as shown in Figure 1.4.

Figure 1.4: Selecting a Partitioning Method



Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Notes

Automatic partitioning creates three partitions:

1. The/boot partition is the home of the kernel: the program at the very heart of Linux. Red Hat recommends a/boot partition of no less than 100 MB, 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;

Example: It's quite common to put / var (where data, including such things as MySQL databases and Websites, live) on a separate partition.

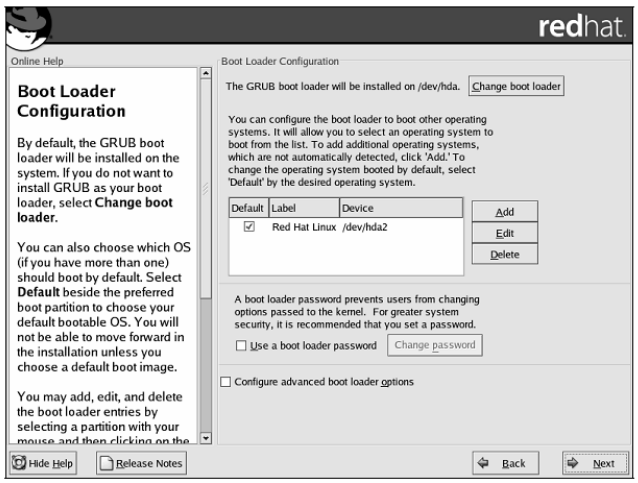
Automatic partitioning makes things simpler, and spreading your data across different partitions doesn't achieve very much. Some administrators strongly recommend it, but the rescue CD will help you avoid most problems that might have been aided by splitting the data across different partitions in the past. Therefore, the default partitioning setup is usually sufficient.

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 1.5 shows, it's pretty easy to set up.

Notes You should set a boot loader password to prevent unauthorized users from gaining access to the kernel's startup parameters.

Figure 1.5: Configuring GRUB



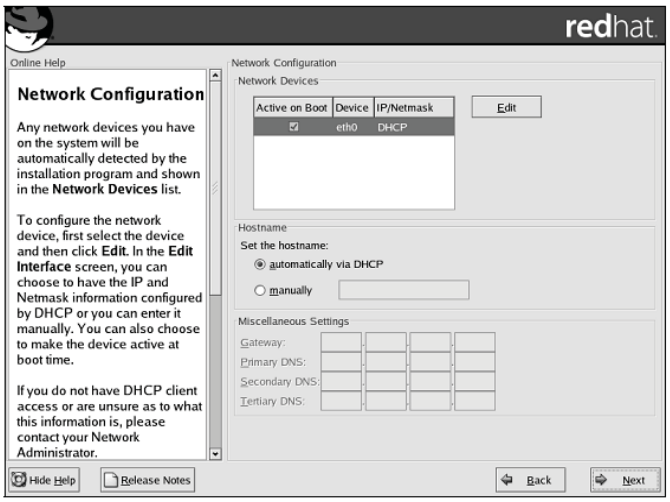
Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Notes

Networking

After you've set up all of your partitions, you'll be offered the networking options shown in Figure 1.6, "Configuring 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 1.6. 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.

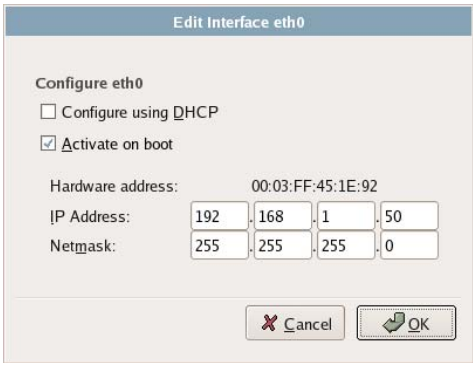
Figure 1.6: Configuring Networking Options



Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Clicking the Edit button in the Network Configuration screen will display the Edit Interface window shown in Figure 1.7. Here, you can make custom configuration adjustments such as giving the server a static IP address.

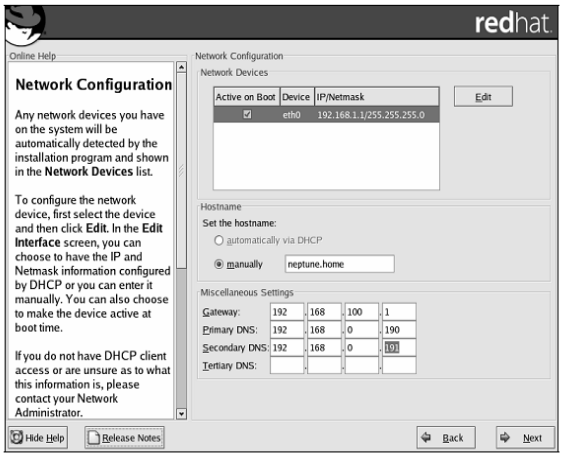
Figure 1.7: Manually Configuring the Ethernet Interface



When the network device settings have been configured from the previous screen, you're free to configure the hostname, gateway and DNS settings. Figure 1.8 shows a network device configured primarily for internal use.

Notes

Figure 1.8: A Manually Configured Network Interface

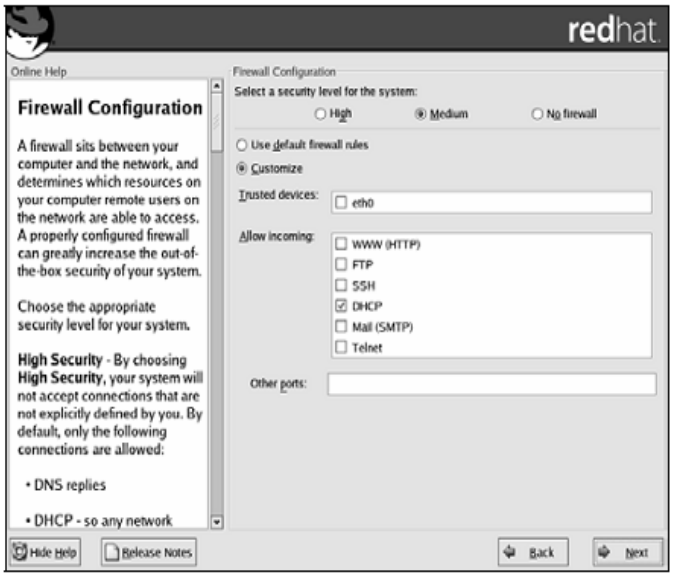


Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Network Security

Various major distributions of Linux – strive to make configuring your network security as easy as possible. 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 1.9.

Figure 1.9: Setting Server Security Options



Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Notes

Telnet and FTP Security

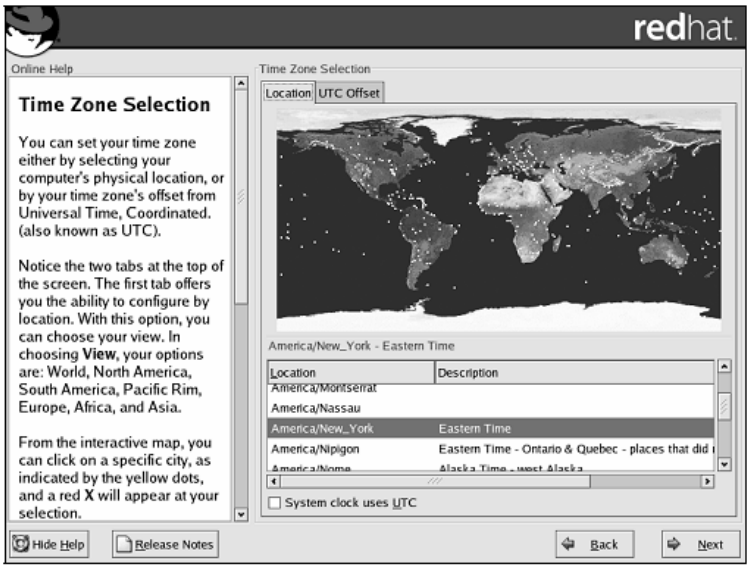
Though they're shown as options in the security configuration screens, both telnet and FTP are widely recognized as insecure protocols. SSH is a much more secure option than telnet for accessing remote machines, as SFTP is a more secure option than FTP for transferring files. If an FTP capability is 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.

Task Make distinction between SSH and Telnet.

Setting the Time Zone

There are 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 1.10.

Figure 1.10: Setting the Time Zone



Root Account Configuration

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 1.11. It is recommended that you record your root password somewhere and keep it safe.

Did u know? If you forget the password, it becomes very difficult to gain access to your machine should things go wrong.

Notes

Figure 1.11: Setting the Root Password

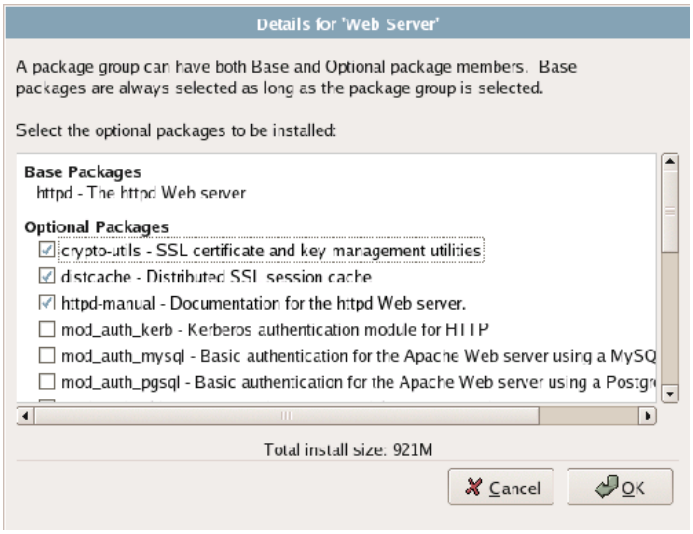
Source: http://www.net-security.org/dl/reviews/0764543784.pdf

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 distribution, with a few nice extras thrown in. Select each of the package groups you want to install by clicking the appropriate check boxes.

Each package group contains a number of packages; you can see a list of these (similar to the one shown in Figure 1.12) 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 1.12: Refining the Package Selection



Source: http://www.net-security.org/dl/reviews/0764543784.pdf

Notes

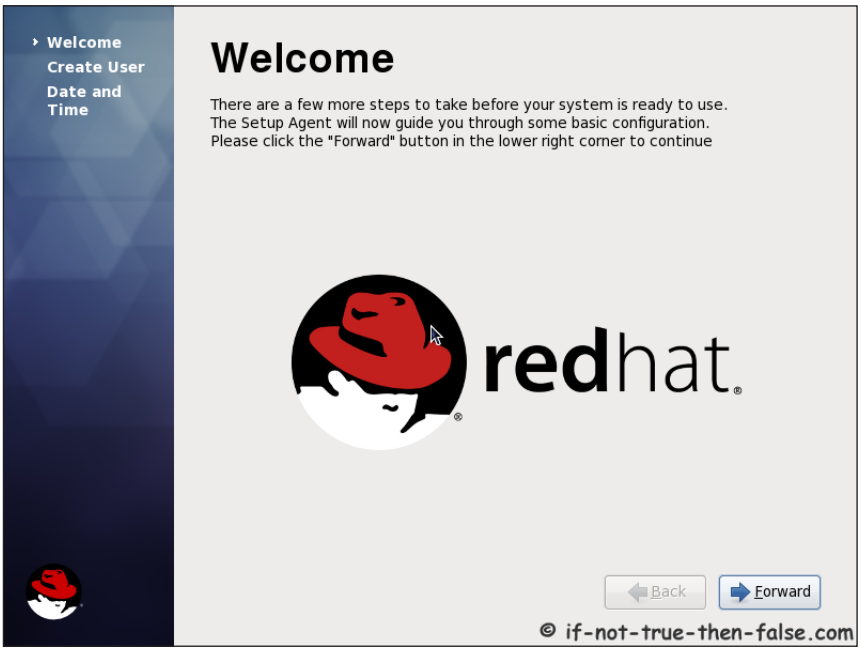
Through a long process of refinement, the Red Hat distributions have come to provide a full range of packages that meet nearly any common computing need. While it's a good goal to keep a server installation to a minimum, you may find that there are some packages you just can't do without.

Caution 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 red hat team to cover the needs of most people.

1.2.4 Welcome of Red Hat Linux

With the main installation completed, a few housekeeping items are all that remain to be done. Your 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 1.13, followed by the licence agreement. Once you've indicated that you agree to the license, you'll enter the configuration screens.

Figure 1.13: The Setup Agent's Welcome Screen



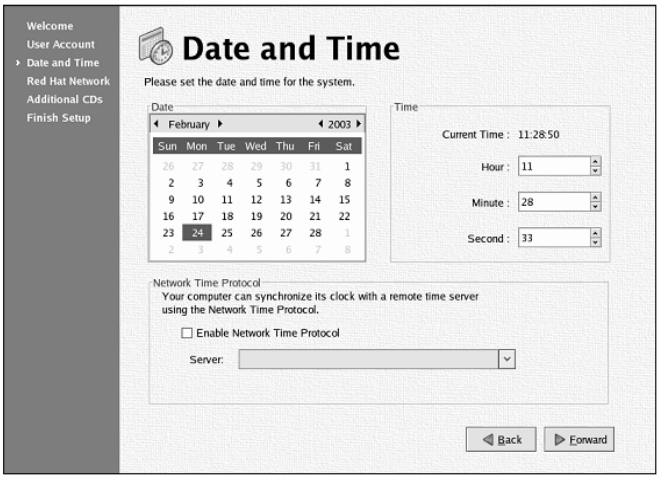
Source: http://www.if-not-true-then-false.com/2010/red-hat-6-installation-guide-rhel-6-install-screenshots/

Setting Date and Time

Now will check and if required, we set the date and time for the computer:

Notes

Figure 1.14: The Network Time Protocol Tab



Source: http://www.net-security.org/dl/reviews/0764543784.pdf

This page has a fairly intuitive calendar interface, and we can set the time using the entry fields toward the right of the screen.

Over a period of time, the clock in the computer tends to drift away from the actual time of the computer. To keep the clock in sync, we can use the Network Time Protocol (NTP) option in the same screen. This allows us to specify an NTP server, and the machine will keep in sync with it.

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 1.15. If you decide to allow network logins, you can also select that option from this screen.

Figure 1.15: Setting up a System User

Source: http://www.net-security.org/dl/reviews/0764543784.pdf

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.

Notes

Caution Performing an action as root without careful forethought can have catastrophic consequences for your system.

Did u know? 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 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.

Notes There are cases in which the text mode installation is a quicker and more efficient means to the same end.

Self Assessment

Fill in the blanks:

8. If you have copied the Red Hat Linux ISO images to a local hard drive, you can use '.....' method.
9. In order to ..... Linux, we must begin by booting the Linux kernel.
10. For your destination diskette, you can use either an ..... or a ..... (for DOS) diskette.
11. The ..... partition is the home of the kernel: the program at the very heart of Linux.
12. Linux partitions don't use the ..... designations.
13. ..... is a program that will let you select from a list of installed operating systems.
14. Once you've indicated that you agree to the license, you'll enter the ..... screens.
15. The ..... account has permissions to create, modify, and destroy any file on the system.

1.3 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.

Notes

- 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.

1.4 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.

CD-ROM: CD-ROM (Compact Disc, read-only-memory) is an adaptation of the CD that is designed to store computer data in the form of text and graphics, as well as hi-fi stereo sound.

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.

FTP: File Transfer Protocol (FTP) is a protocol through which internet users can upload files from their computers to a website or download files from a website to their PCs.

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.

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

1.5 Review Questions

1. Explain the steps that are to be performed before installation.
2. How do you perform a server installation? Discuss.
3. Illustrate the process of installing Red Hat Linux.
4. Discuss the steps used in choosing Installation types.
5. Describe the steps for creating a Linux boot disk under DOS.
6. Explain the different types of Linux installation methods.
7. Describe the Disk Partitioning process during Linux installation.
8. Describe how to set up the Root User.
9. What is the installing from CD-ROM?
10. What is the hard drive partition installation?

Notes

Answers: Self Assessment

1. Red Hat Enterprise Linux (RHEL)
2. Extended Search
3. Hardware
4. partitions
5. Kernel
6. filesystems
7. NTFS-based
8. Hard Drive
9. install
10. /boot
11. drive letter
12. drive letter
13. GRUB
14. Configuration
15. root

1.6 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.control-escape.com/linux/lx-install.html

http://www.pcworld.com/article/155517/ubuntu_install_guide.html

http://www.ee.surrey.ac.uk/Teaching/Unix/

http://ebook.cna.ilkom.unsri.ac.id/linux/Installing_RHEL_5.pdf

Notes

Unit 2: Red Hat Linux 9 Basics

CONTENTS

Objectives
Introduction
2.1 Moving Around the Desktop
2.1.1 The Menu System
2.1.2 The Panel
2.1.3 The Desktop Space
2.2 Using Applications
2.2.1 Run Small Applications on the Panel
2.2.2 Using Launchers to Run Applications
2.3 The File System
2.3.1 The File System Hierarchy
2.3.2 Navigating in the File System
2.4 Managing the Hardware
2.4.1 Detecting the Available Hardware
2.5 Configuring the Desktop
2.5.1 The GNOME Configuration Tool
2.5.2 The GNOME Control Center
2.6 Managing Processes
2.6.1 Understanding Processes
2.6.2 Viewing Running Processes
2.6.3 Taking Control of Running Processes
2.6.4 Understanding Run Levels
2.7 Managing Users
2.7.1 Concept of User Management
2.7.2 Understanding Permissions
2.7.3 Concept of Managing Users
2.8 Summary
2.9 Keywords
2.10 Review Questions
2.11 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain the process of preparing for installation
- Describe the process of installation
- Discuss Root account configuration

Notes

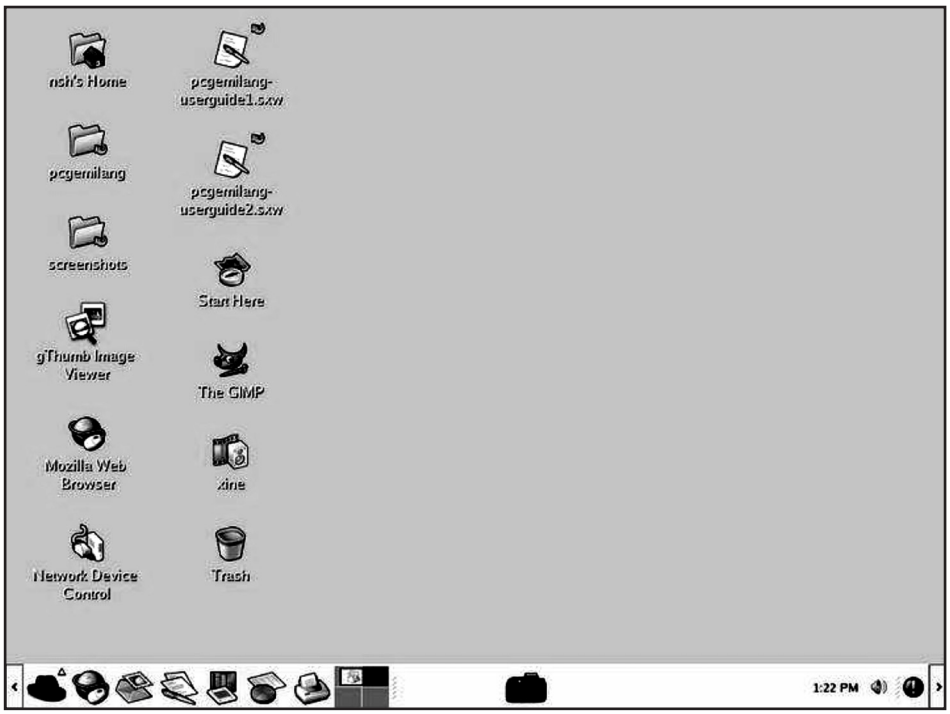
Introduction

If you are not much familiar with Linux, there is a need of practical guide that provides the information regarding the installation and set-up of Linux, and also shows the process of using Linux for specific functions, for example a Web server or a software development platform. The version of Linux is improved by Red Hat. The Red Hat Linux 9 is released recently which includes various new system components, including the XFree86 4.3.0, GCC 3.2.2 compiler, Linux 2.4.20 kernel, and the glibc 2.3.2 system libraries. The X Window System is considered as XFree86 version 4.3.0 with support for many more new as well as powerful graphics cards as compared to the previous versions. The desktop experience of Linux is improved by Red Hat. It provides many more graphical tools which are used to configure and manage the system. For productivity applications, Red Hat includes the Mozilla Web browser, the OpenOffice.org office suite, and Ximian Evolution personal information manager. Also, Red Hat has unified the appearance and feel of GNOME as well as KDE desktops. Thus, irrespective of the selected desktop, the user feels at home. Also, there are many improvements. Red Hat Linux now makes use of the CUPS (Common UNIX Printing System) as the default printing system. It also includes the Native POSIX Thread Library, which provides improvements in performance with Pentium Pro processors or better.

2.1 Moving Around the Desktop

The Figure 2.1 shows a typical view of the graphical Linux desktop.

Figure 2.1: A Typical Linux Desktop



Source: http://linux.about.com/od/linux101/a/desktop01d.htm

There are three main components on the desktop:

- the Menu System
- the Panel
- the Desktop itself

Notes

2.1.1 The Menu System

We can accsess main menu by clicking on the Main Menu button situated at the extreme lower left hand corner of the desktop. This may be presented by one of several icons depending on the desktop environment used.

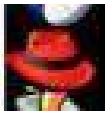
Generic GNOME Main Menu Button is shown below:

Figure 2.2: Generic GNOME Main Menu Button

Source: http://linux.about.com/od/linux101/a/desktop01d.htm

Red Hat Main Menu Button is shown in Figure 2.3.

Figure 2.3: Red Hat Main Menu Button



Source: http://linux.about.com/od/linux101/a/desktop01d.htm

This will bring up the Menu System as shown in the Figure 2.4.

Figure 2.4: The Menu System

Source: http://linux.about.com/od/linux101/a/desktop01d.htm

We can start various applications (installed on your Linux system) from the menu system. The Menu System consists of a Main Menu panel and sub-menu panels. Every entry in the Menu

Notes

System which has an arrow on its right signifies that it is an entry point to a sub-menu, and there can be sub-menus within every sub-menu. In this manner, the Menu System applications can be organised as well as categorised for easy reference and access.

To use the sub-menu associated with a menu entry, move the mouse and rest it on the menu entry in question and a sub-menu panel will appear. Figure 2.5 shows the Main Menu and Sub-menu.

Figure 2.5: Main Menu and Sub-menu

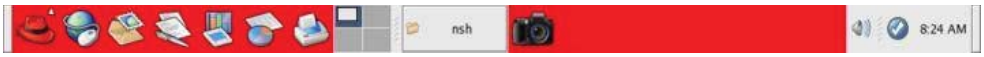
Source: http://linux.about.com/od/linux101/a/desktop01d.htm

Clicking once on a menu entry will cause an application associated with it to be launched, i.e. executed.

2.1.2 The Panel

The long bar across the bottom of the desktop screen is known as the Panel. The Panel consists of the Main Menu icon, the application launcher icons, a notification area and applets.

Figure 2.6: The Panel



Source: http://linux.about.com/od/linux101/a/desktop01d.htm

There are several application launcher icons which are installed by default on the Panel. Clicking on one of these will run an application. Generally accessed applications can be added to the Panel and those that are less frequently used can be taken off.

Did u know? The notification area holds alert icons so that the user can be alerted to critical messages.

Notes

The small applications that run on the Panel are called applets. These usually perform useful and informative tasks like setting the sound level of the soundcard, monitoring whether the system software needs an update, etc. The following applets run by default.

- ***The Workspace Switcher:*** We can consider the graphical desktop as a workspace drawing an analogy with the working area on a real physical tabletop. Programs are run, documents displayed and files opened on the workspace. To cut down on workspace clutter and to allow the user to organise his workspace more efficiently, the graphical desktop environment permits the usage of numerous workspaces. Every workspace can be considered as a virtual desktop.

Figure 2.7: The Workspace Switcher

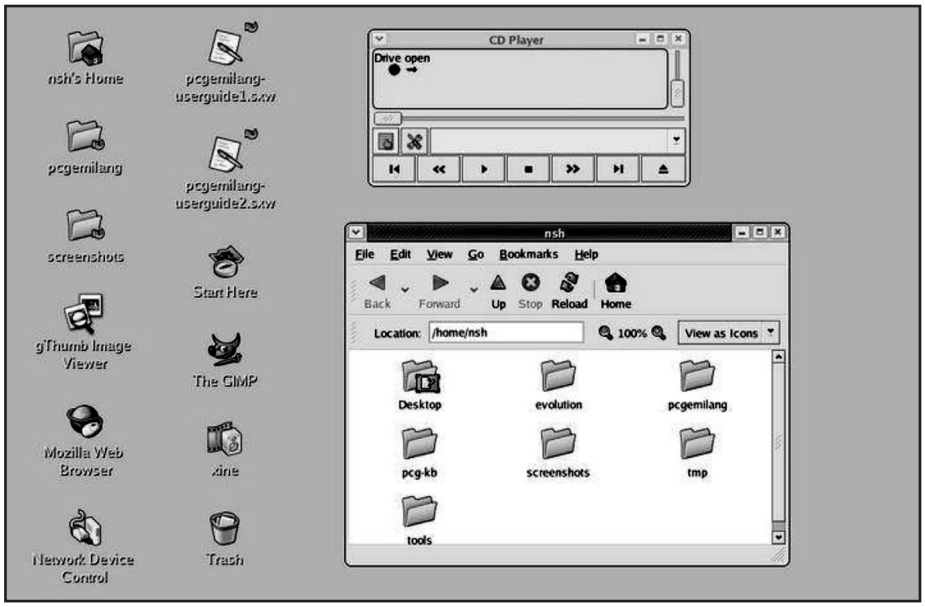
By default the user has four desktop workspace areas to work on. The workspace switcher represents every workspace as a small square and shows the applications running in each of them. To access a workspace click on the square with the mouse.

- ***The Taskbar:*** This applet is situated next to the workspace switcher and displays the titles of all the running applications in a virtual desktop (a workspace).

2.1.3 The Desktop Space

The Desktop space refers to the remaining screen. It consists of icons which are graphical representations of short-cuts to application launchers, file folders, files and peripheral devices like floppy disks, CD-ROM drives and printers. Double-clicking on an icon representing an application will launch or execute the application. Generally used applications and/or files/ folders are usually located on the desktop space.

Figure 2.8: The Desktop Space



Source: http://linux.about.com/od/linux101/a/desktop01d.htm

Notes

Task Make distinction between workshop switcher and taskbar.

Self Assessment

Fill in the blanks:

1. Each entry in the Menu System which has an arrow on its right means that it is an entry point to a .....
2. The long bar across the bottom of the desktop screen is called the .....

2.2 Using Applications

The Desktop provides a useful metaphor for a modern office work environment. Thus, find on the Desktop that the applications that we want to run can easily be started or launched and the required information and data can easily be located.

Applications that we can run from the Desktop are to be found either from the Main Menu (and sub-menus therein), or as icons on the Panel and the Desktop itself. To run an application from the Main Menu, open up the menu (or sub-menu) and click on the application listed in the menu bar.

To run an application from the Panel, click once on the icon representing the application.

To run an application from the Desktop itself, double-click on the icon representing the application.

At times, it is more handy to have the application as an icon on the Panel or Desktop where we can launch it more simply by just clicking on it, rather than in the Main Menu, particularly if it is hidden deep inside several sub-menus. To attain this, open the Main Menu and choose the application item listed in it by clicking with the rightmouse button. Click on the selection "Add this launcher to panel" and a copy of the application icon will be located on the Panel.

To make a copy of this on the Desktop, we can drag the icon from the Panel over to the Desktop.

To delete an application icon from the Panel right-click on it and choose the "Remove from Panel" option.

To delete an application icon from the Desktop, right-click on it and choose the "Move to Trash" option.

2.2.1 Run Small Applications on the Panel

The Panel houses many useful utilities called applets. Applets are small applications that run on the Panel.

By default, the taskbar and the workspace switcher applets are placed and run on the Panel. Other useful applets that may be placed on the Panel include: Clock, sticky notes, and volume control.

Placing an Applet on the Panel

For placing an applet on the Panel, we move the mouse over to an empty space on the Panel and right-click it. Choose the item "Add to Panel" and from the sub-menus choose the applet to place on the Panel.

Notes

Example: For example, to place the "sticky notes" applet, select:

Add to Panel → Accessories → Sticky Notes

To avoid accidental removal of an applet, we can lock it on the Panel by right-clicking on its icon and choosing "Lock".

Removing an Applet from the Panel

In order to remove an applet, right-click on the applet icon and choose "Remove from Panel". If the applet is locked, you will have to unlock it first by right-clicking on the icon and selecting "Unlock".

Configuring the Panel

In order to change the properties of the Panel, right-click on the Panel and choose "Properties". From the general properties menu, we can change the orientation, size and (un)hide the Panel. From the background properties menu, we can change the colour of the Panel in addition to its visual appearance.

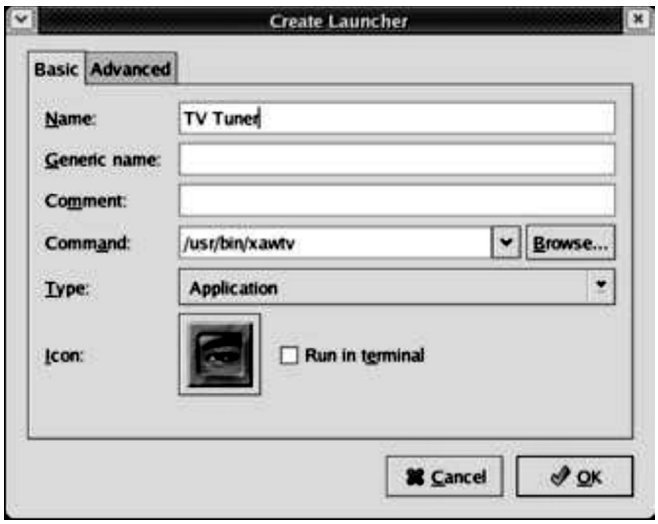
2.2.2 Using Launchers to Run Applications

Using Launchers provide a quick way to the user to access specific resources on the system.

Example: If a user needs to access a specific file in one of the folders often, he can create a launcher to run an application to open the file and this launcher can be placed it on the Desktop. In this way the resource (the file) can be accessed very quickly by just double-clicking on it.

To create a launcher on the Desktop, right-click on an empty area on the Desktop and select the item "Create Launcher". Enter the Name and the Command to run and if you want you can select an icon for it by clicking on the icon button.

Figure 2.9: Creating a Launcher



Notes

Self Assessment

Fill in the blanks:

3. ..... are small applications that run on the Panel.
4. ..... allow the user a quick way to access specific resources on the system.

2.3 The File System

A file system is defined as 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.

Most files are just files, called regular files; they contain normal data.

Example: Text files, executable files or programs, input for or output from a program and so on.

While it is reasonably safe to suppose that everything you encounter on a Linux system is a file, there are some exceptions.

- *Directories:* files that are lists of other files.
- *Special files:* the mechanism used for input and output. Most special files are in /dev.
- *Links:* a system to make a file or directory visible in multiple parts of the system's file tree.
- *(Domain) sockets:* a special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.
- *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 2.1: 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.

2.3.1 The File System Hierarchy

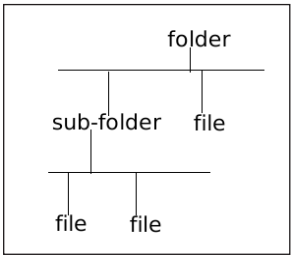
Similar to a physical folder, an electronic folder can enclose various files. A folder may also have sub-folders or sub-directories.

We can create, copy, move, and delete files and folders. The folders themselves are organised in a hierarchical manner starting at the root of the file system. Every user is provided a home directory and upon logging in, the user is placed in his home directory.

Everything begins from the root directory, symbolized by '/', and then expands into sub-directories. Where DOS/Windows had several partitions and then directories under those partitions, Linux places every partition under the root directory by 'mounting' it under specific directories.

The Hierarchical File System on Linux Desktop is shown as below:

Figure 2.10: The Hierarchical File System



In case of Windows, the various partitions are detected at boot and allocated a drive letter. In case of 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.

Example: 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.

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.

Notes

Linux chooses to be case sensitive. This signifies that the case, whether in capitals or not, of the characters becomes very significant. 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/

Some Top-level Directories are Discussed Below:

/sbin – This directory includes all the binaries that are important to the working of the system.

These comprise system administration in addition to 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 includes other binaries that are useful 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 comparison to /sbin, the bin directory includes various useful commands. Both the system administrator and non-privileged users make use of these commands. 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 - The system.map file and the Linux kernel are contained in this directory. LILO places the boot sector backups in this directory.

/dev - This directory highlights one significant feature of the Linux filesystem, that is, 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. /etc – All the configuration files for your system are included in this directory. 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 significantly, 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 considered as a multi-user environment. Thus every user is also allocated 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 includes the user specific settings for programs such as IRC, X etc.

/lib - All the shared libraries needed by system programsare included in this directory. Windows equivalent to a shared library would be a DLL file.

/lost+found – There should always be a proper shutdown in Linux. At times, 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. The files recovered are not likely to be complete or make much sense but there always is a chance that something worthwhile is recovered.

Notes

/mnt – This is a generic mount point under which you mount your filesystems or devices. The process by which a filesystem is made available to the system is known as mounting. 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 – All the software and add-on packages that are not part of the default installation are contained in this directory. 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.

/proc – This is a special directory on your system.

/root – This 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 – The files that are required temporarily are included in this directory. 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 directory 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 includes spooling data such as mail and also the output from the printer daemon. Also, 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.

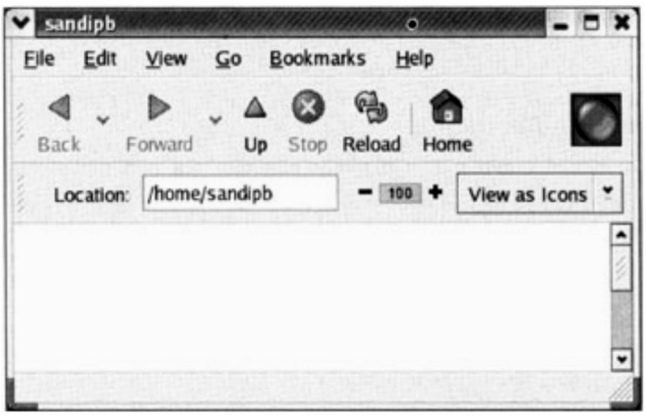
2.3.2 Navigating in the File System

We can navigate the file system by means of either a GUI file explorer (for example, Nautilus) or the terminal.

Navigating Using the GUI

To begin the GUI file explorer, it is required to execute the Nautilus file manager. In order to start Nautilus in a quick way, we can choose it from the menu. That is, by selecting Main Menu → Home Folder. This will provide the Nautilus file manager, which looks like the Windows Explorer of Microsoft. When the Nautilus file manager is opened in this manner, then the directory displayed in the location bar is your home directory.

Figure 2.11: Home Directory



Notes

Now, we can type any location into the location bar. The contents will be displayed at that location. On the other hand, we can select the button marked Up in order to make Nautilus show the contents of the parent directory instead. Thus, we can navigate the whole way up to the root directory (/). All the top-level directories of the computer can be found in the root directory. In order to navigate inside the directories, simply double−click on the directories.

Generally, the file system navigation by means of the file manager is very intuitive. If you are familiar with an operating system based on GUI, then you will see yourself right at home with Nautilus.

Navigating Using the Terminal Emulator

When we open a terminal emulator window, it places us at the shell prompt, and in our home directory. The shell is defined as the program which interprets every command that we type at the terminal.

The directory in which we are located at any specific time is known as the current directory. When we open up the terminal first time, then the current directory is set to be the home directory. We can find out the directory at any time by executing the pwd command.

Example: The command pwd:$ pwd /home/sandipb signifies that the pwd program is being utilized to confirm that the current directory just after opening the terminal application is the home directory of the user.

Self Assessment

Fill in the blanks:

5. ..... makes a file or directory visible in multiple parts of the system's file tree.

6. ..... directory contains all the binaries that are essential to the working of the system.

2.4 Managing the Hardware

The installation process of Red Hat Linux automatically attempts to identify the existing hardware and configure it so as to make use of it in the desktop. At times, this hardware configuration is required to be fine-tuned, or the auto-detection of hardware at some stage in the installation process does not function as intended. In these type of cases it is essential to know the process of adjusting the desktop's hardware configuration manually.

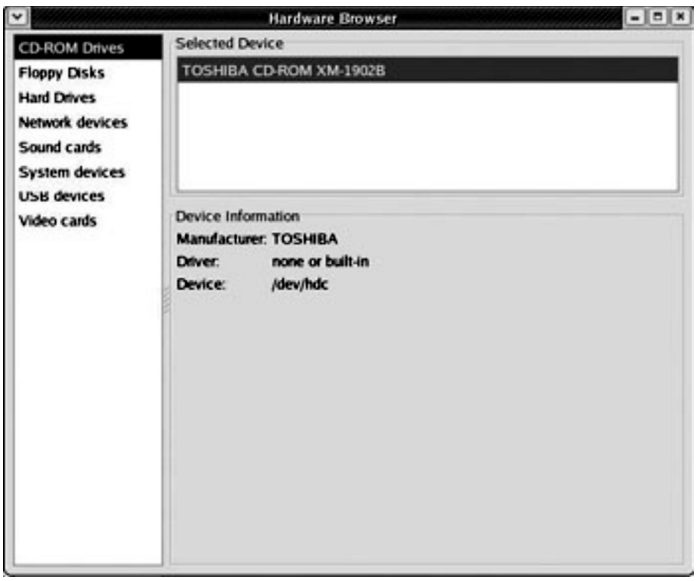
2.4.1 Detecting the Available Hardware

The present major hardware configuration can be determined by means of the hardware browser. In order to launch the hardware browser, choose Main Menu → System Tools → Hardware Browser. Alternatively, type hwbrowser at a command line. The function of hardware browser is to look for the hardware on the system and show the results. This is shown as below:

We can collect more information regarding the hardware by working from the terminal itself. In this section, we will also discuss how to gather information regarding the system by means of the CLI. The main areas that we will consider for investigation are as follows:

Notes

Figure 2.12: Finding the Available Hardware



Source: http://flylib.com/books/en/1.65.1.26/1/

Area	Description
CPU	The type of CPU and the amount of processing load.
Memory	The amount of memory installed on the system and its usage pattern.
Hard disk, CD-ROM, and floppy drives	The disk capacities and their usage status.
Network cards	The network devices which are operational and their usage statistics.
Mouse	The kind of mouse being used by the system.
Keyboard	The kind of keyboard being used by the system.
Display	The display card detected by the system and its current usage status.
Various PCI and USB devices	The various devices detected by the system.
Sound	The sound card detected and used by the system.

Operating system maintains some special files in a dynamic manner. A lot of information can be found by using these special files. These files are located in the /proc directory. We call these files special as operating system generates them "on demand". Also, these files are not actually available on the file system except when requested (for instance, by the ls /proc command).

The cat command can be used to view the contents of these files. On the other hand, we can make use of a text editor like gedit.

CPU-related Information

We can find out the CPU information discovered by the operating system by displaying the contents of the file /proc/cpuinfo. This is displayed as below:

Notes

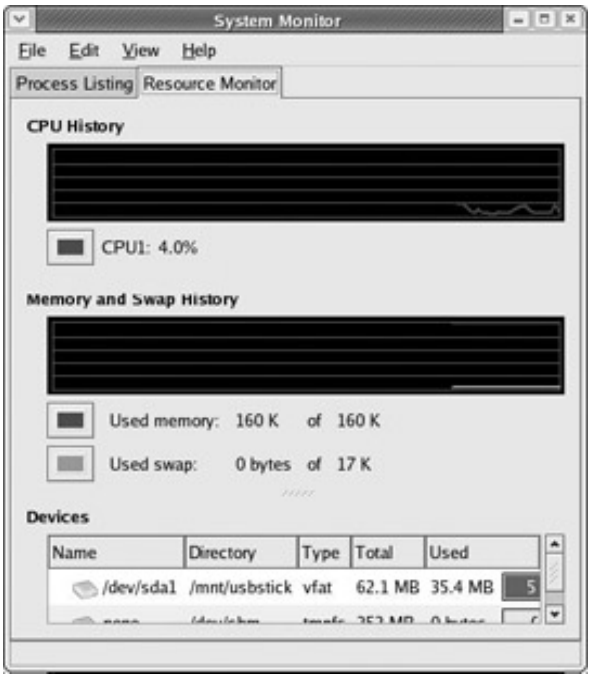
Figure 2.13: The CPU Information

Source: http://flylib.com/books/en/1.65.1.26/1/

The value of bogomips specified at the bottom of the display is an approximate calculation of the processor's speed. We can use this figure to get an approximate comparison of the relative speeds of two different computers running different processors.

The usage of CPU can be found in various ways. There is a graphical tool known as the System Monitor. We can launch system monitor by choosing Main Menu → System Tools → System Monitor. In this application, the System Monitor tab displays the usage of CPU over time in a graphical format which is shown as below:

Figure 2.14: System Monitor Tab



Source: http://flylib.com/books/en/1.65.1.27/1/

Besides the CPU usage information, the System Monitor tab also gives information on memory usage and mounted file systems.

Notes

Notes Actually, Swap memory is not a memory in the conventional sense. It is a disk space which is used by the operating system in order to store information for the time being.

The Use of Load Average to Measure CPU Usage

The CPU usage, in case of Linux as well as other UNIX-based operating systems, is frequently assessed in terms of load average. We can display Load average by making use of three numerical values. The load average is defined as the average number of processes which are waiting to be executed system over a specified time interval. The load average can be examined in different manners.

Example: We can examine the contents of the file /proc/loadavg.

Here, the first three figures (0.44, 0.29, 0.17) depict the system's load average taken over the last one, five, and fifteen minutes.

The output of this program also displays the total time for which the computer has been running.

The top program can be executed. This program displays a wealth of information. Also, the display updates itself in every few seconds.

Note that the first part depicts the load average of the system, and the usage of memory in the system. The remaining display demonstrates the processes running on the system, together with a wealth of information on every process.

This output allows you to change many properties of the display and even to destroy processes.

In order to exit the top display, it is required to press q on the keyboard.

Memory-related Information

To determine information regarding the total amount of RAM in the system and the amount of swap memory, we can look at the System Monitor dialog. Another quick option for determining the status of RAM and swap is to display the contents of the file /proc/meminfo. This is shown as below:

Figure 2.15: Example of Command /proc/meminfo

```
cat /proc/meminfo
MemTotal:        3973736 kB
MemFree:          431064 kB
Buffers:           46604 kB
Cached:           494648 kB
SwapCached:        11360 kB
Active:          2322760 kB
Inactive:         933028 kB
Active(anon):    2057952 kB
Inactive(anon):   679956 kB
Active(file):     264808 kB
Inactive(file):   253072 kB
Unevictable:          16 kB
Mlocked:              16 kB
SwapTotal:       4096568 kB
SwapFree:        3961748 kB
Dirty:               236 kB
Writeback:             0 kB
AnonPages:       2704520 kB
Mapped:           182240 kB
Shmem:             23372 kB
Slab:              93848 kB
SReclaimable:      52044 kB
SUnreclaim:        41804 kB
KernelStack:        5064 kB
PageTables:        64928 kB
```

Source: http://superuser.com/questions/521551/cat-proc-meminfo-what-do-all-those-numbers-mean

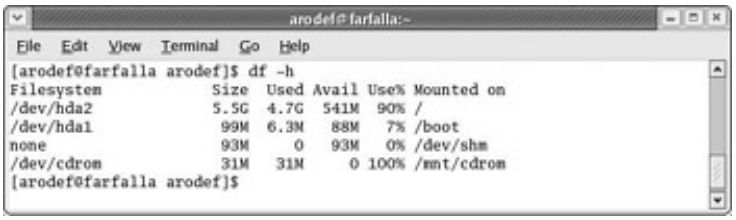
We can also examine the usage of memory via the System Monitor or the top program.

Notes

Information Regarding the File System Devices

We can use System monitor's graphical interface to locate information on the mounted file system in the computer. On the other hand, there is also a fast CLI-based method to find similar information, and this requires the df program. We have shown below the usage of the df program:

Figure 2.16: df Program Usage



Source: http://flylib.com/books/en/1.65.1.26/1/

In the figure shown above, the -h option shows the output in a more readable format.

If using a hard disk with multiple partitions as well as multiple operating systems, it is useful to be able to check the hard disk's partition structure and the different partition types obtainable on it. We can show this by using the fdisk program, with the –1 option.

The superuser mode is showed by the change in the prompt. That is, the last character of the prompt changes from a $ sign to a # sign.

The −l option is utilized to query the hard disk structure, and is quite safe. On the other hand, other fdisk options manipulates your hard disk partitions, and erroneous usage of these options can demolish the data of hard disk.

Caution It is essential to make use of the fdisk program only with care.

Network Cards

As a Linux user you should know about some network properties like network card name, speed's, driver details etc. Here we will discuss the same with help of examples.

Example 1: Find all your network cards (Ethernet, Gigabit Ethernet, Wireless) names attached to a given system.

Command

ifconfig | cut -c1-8 | sort -u

Output

eth0

lo

virbr0

wlan0

Example 2 : find how many network connections(network cables are connected to your machine) are active i.e. link is up.

Command

ip link show

Notes

Output:

1. lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN mode DEFAULT

   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2. eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT qlen 1000

   link/ether 10:1f:74:58:e1:04 brd ff:ff:ff:ff:ff:ff

3. wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DORMANT qlen 1000

   link/ether d0:df:9a:e6:1d:a6 brd ff:ff:ff:ff:ff:ff

4. virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT

   link/ether 52:54:00:be:79:50 brd ff:ff:ff:ff:ff:ff

5. virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN mode DEFAULT qlen 500

   link/ether 52:54:00:be:79:50 brd ff:ff:ff:ff:ff:ff

   Or

   As a root user

   mii-tool

   Output:

   eth0: no link

   Or for wireless network card

   ethtool interfacename

Example

root@linuxnix.com:/home/surendra# ethtool wlan0

Settings for wlan0:

Link detected: yes

Example 3: How can we find out network card speeds so that we can see throughput of the devices attached to a machine.

ethtool eth0 | grep speed

Output:

ethtool eth0 | grep Speed

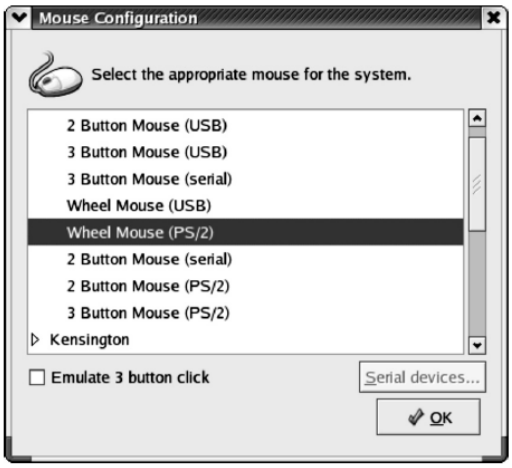
Speed: 10Mb/s

Information Regarding the Mouse and the Keyboard

The tool of Red Hat Linux used for the mouse configuration is the program redhat−config−onfig. This can be run from the command line. Alternatively, we can run this by choosing Main Menu → System Settings → Mouse. You will have found this program already, as it was run during the installation process.

Notes

Figure 2.17: Red Hat Linux's Tool for Configuring the Mouse



Source: http://flylib.com/books/en/1.65.1.26/1/

The dialog box, shown in Figure 2.17, displays the sysytem's current mouse configuration.

The GUI tool "redhat−config−onfig−keyboard" program is configuring the keyboard layout When we execute this program, it shows the dialog box shown below.

Figure 2.18: Red Hat Linux's Tool for Configuring the Keyboard

Source: http://flylib.com/books/en/1.65.1.26/1/

Information Regarding the Display

The lspci command is used to find out the information regarding the display card on the system. This discovers the actual video hardware on the system. In the lspci output, the video hardware was disclosed to be a S3 ProSavage card.

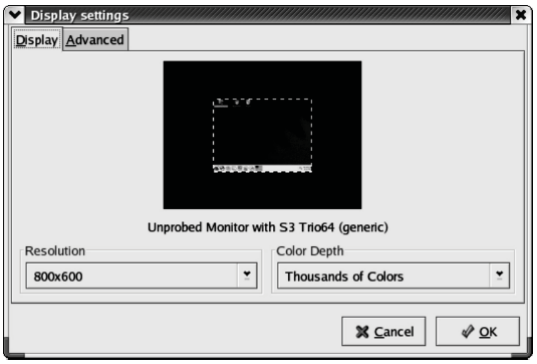
The X-Server software is accountable for the fundamental graphical display of the desktop. The X-Server software loads suitable display drivers as per the video hardware.

Did u know? If any roughness is visible in the display, it is possible that your video card has been detected wrongly, and that a generic display driver lsuch as Vesa is loaded.

Notes

We can use the Red Hat configuration tool redhat−config−xfree86 in order to check or modify the video card as well as monitor settings of the display. This can be done by typing this command at the command line. Alternatively, choose Main Menu → System Settings → Display. The Figure 2.19 shows the dialog box which appears on the execution of this tool.

Figure 2.19: Display Setting



PCIs and USBs

Peripheral Component Interconnect (PCI), as its name implies is a standard that describes how to connect the peripheral components of a system together in a structured and controlled way. The standard describes the way that the system components are electrically connected and the way that they should behave.

The PCI initialisation code in Linux is broken into three logical parts:

PCI Device Driver

This pseudo-device driver searches the PCI system starting at Bus 0 and locates all PCI devices and bridges in the system. It builds a linked list of data structures describing the topology of the system. Additionally, it numbers all of the bridges that it finds.

PCI BIOS

This software layer provides the services described in bib-pci-bios-specification. Even though Alpha AXP does not have BIOS services, there is equivalent code in the Linux kernel providing the same functions,

PCI Fixup

System specific fixup code tidies up the system specific loose ends of PCI initialization.

List all PCI Devices

lspci is a utility for displaying information about PCI buses in the system and devices connected to them. By default, it shows a brief list of devices. Use the options described below to request either a more verbose output or output intended for parsing by other programs.

lsusb – List USB Devices

To make use of all the features of this program, you need to have a Linux kernel which supports the /proc/bus/usb interface (e.g., Linux kernel 2.3.15 or newer).

Notes

Command

$ lsusb

Output:

Bus 004 Device 002: ID 0930:6532 Toshiba Corp.

Bus 004 Device 001: ID 0000:0000

Bus 003 Device 001: ID 0000:0000

Bus 002 Device 001: ID 0000:0000

Bus 001 Device 001: ID 0000:0000

To get verbose output type the command:

$ lsusb -v

Information Regarding the Sound Card

To find out the information regarding the sound card, we can use the sound card configuration tool, that is, redhat−config−soundcard. We can run this typing this command at the command line. Alternatively, choose Main Menu → System Settings → Soundcard Detection. This program shows a simple dialog box displaying that the sound card is detected on the system and the kernel sound driver is loaded:

Figure 2.20: The Sound Card

Source: http://flylib.com/books/en/1.65.1.26/1/

Self Assessment

Fill in the blanks:

7. The function of ..... is to search for the hardware on the system and show the results.

8. The ..... mode is showed by the change in the prompt.

2.5 Configuring the Desktop

The default desktop environment that is used until now is the GNOME desktop environment.

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, disabilitites, or native language. GNOME makes it easy for people to use their computers.

Notes

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.

2.5.1 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.

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.

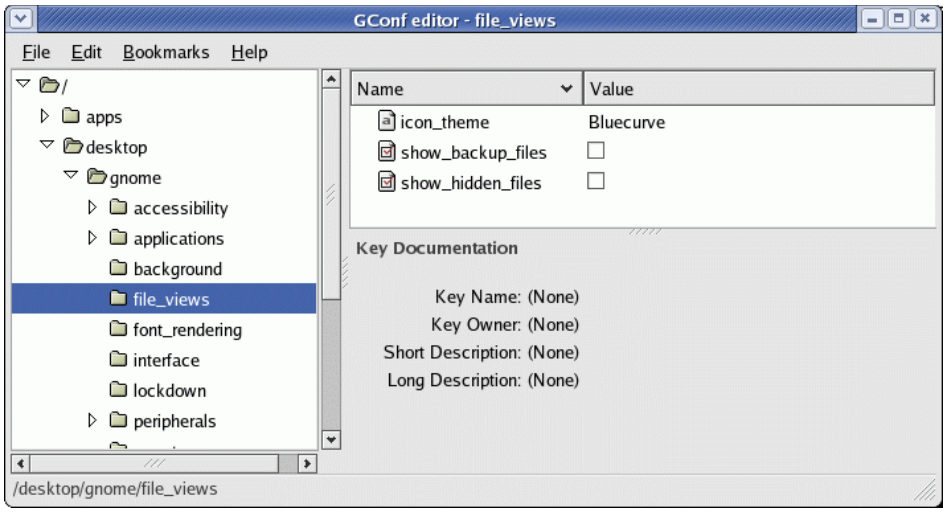
GConf can look up settings from different settings at once, typically from different locations on the file system. By having appropriate system sources configured, GConf enables systems administrators to provide both default and mandatory settings for all users. Tools such as GNOME's Configuration Editor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

Figure 2.21: GConf Editor



Notes

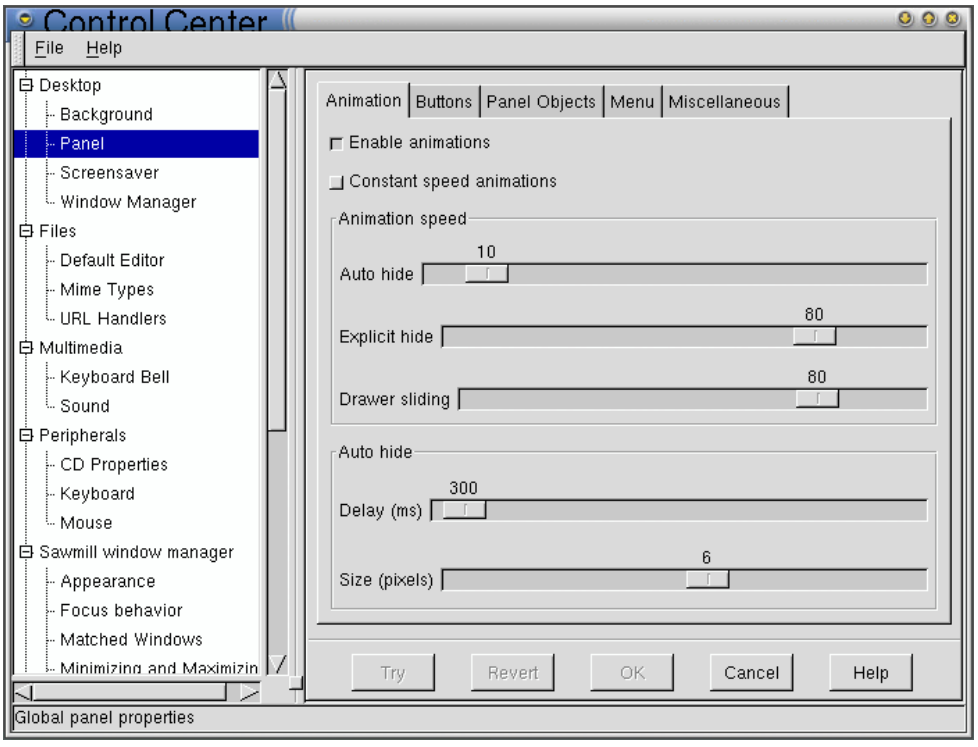
GConf is used by GNOME to expose settings to multiple applications. GConf is the GNOME way of persisting application settings and should be used by GNOME application programmers. GConf includes notification service alerts to applications to changes in configuration data, and is used by GNOME itself. A command line tool (gconftool, FC3 /usr/bin/gconftool-2) and a simple GUI application (gconf-editor) are supplied to facilitate administration.

2.5.2 The GNOME Control Center

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 2.22. 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.

Figure 2.22: The GNOME Control Center



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.

Using control center, you can:

- Select background properties

Notes

- Configure a screen saver
- Select a desktop theme
- Select a window manager
- Configure the default text editor
- Specify MIME types that control the handling of multimedia files
- Configure the keyboard bell and sounds
- Configure keyboard and mouse properties
- Specify applications that GNOME automatically launches when it starts
- 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. The Try button lets you experiment without permanently altering the GNOME configuration.

Caution The OK button permanently updates the GNOME configuration whereas the Cancel button discards your changes.

Desktop

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

- ***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.
- ***Global Panel Preferences:*** This capplet is provided by the Gnome Panel and is documented in the Panel Manual.
- ***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.
- ***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).
- ***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.

Notes

Document Handlers

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. This section configures the default programs used to run specific files based on their content type or URL.

Default Editor: Default editor sets the text editor to be utilised by default when we open an editable file or select the Edit menu command on a file within the file manager. A drop-down list of available editors is displayed. Click the Run in Terminal Window button if your chosen editor must be run in a terminal.

Mime Types: Mime types allows us to set or edit mime types. Click the Add button to add a new type. A dialog will open asking for the category/type listing for the mime type and the extensions to associate with it. Optionally, you can supply up to two regular expressions to identify the mime type. To edit a mime type, select it in the listing and click the Edit button. You will see a dialog box in which you can choose an icon to be used for the file type, add or remove file extensions, and supply commands that will open, view, and edit this type of file. To delete a type, select it from the list and click the Delete button.

URL Handlers: It allows us to adjust the settings for special URL launchers utilized by the GNOME help system. The defaults for protocols like HTTP, FTP, and Mail are already set and likely handled by your default web browser (for instance, Netscape). The special URLs are *ghelp*, *info*, and *man*, corresponding to GNOME help files, command *info* files, and *man* files. The defaults use the help browser for these types of files. It is best to leave these settings as they are.

Task Analyze the functions provided by GNOME Control Center.

Self Assessment

Fill in the blanks:

9. ..... is the system for storing and retrieving configuration settings in GNOME.
10. The GNOME Control Center allows you to configure various parts of your GNOME system with different tools called "....."..
11. The ..... section controls the most visible global settings for your Gnome environment.

2.6 Managing Processes

A process is defined as an independent program entity which executes and makes use of computer resources like CPU time and memory. Every application that is executed is started as an independent process. An application may begin its own child processes. Certainly many applications such as web servers and database servers consist of numerous processes to fulfil the requirements of different clients simultaneously.

2.6.1 Understanding Processes

Generally, a process is initiated by a parent process. The process that is launched recently can successively launch child processes if required. When working at the desktop, all new programs are executed as child processes of the program that is obtaining as well as interpreting the commands.

Notes

Example: Suppose that you are at the terminal prompt, it means that you are communicating with the shell interpreter program.

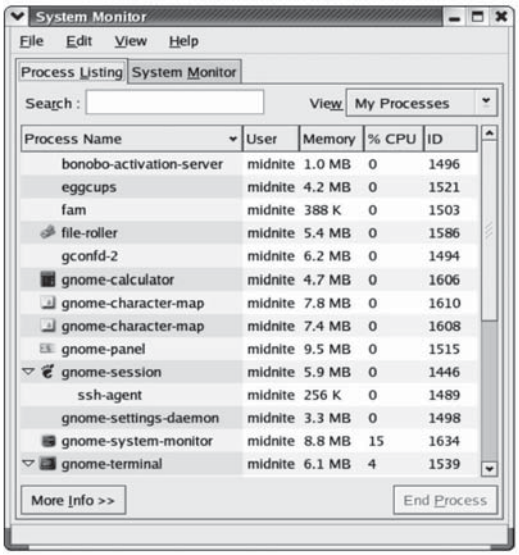
All executing programs are considered as processes that have been launched by some other parent processes. Thus, we can represent all the running processes in the form of a hierarchical tree.

Notes When launching a child process, the parent process creates a copy of it by forking, and then executes the child program in the memory space of the new copy.

2.6.2 Viewing Running Processes

The System Monitor program provides permission to see the processes that are running in the machine. Figure 2.23 shows the list of processes.

Figure 2.23: List of Processes



The processes sorted by any column on display can be viewed by just clicking at the respective column header. We have sorted the processes in accordance with their memory usage.

We can utilze the ps program at the terminal in order to view the processes running in the machine in a non-dynamic manner. We can make use of the ps program without any arguments to view every process running within your present logging session. In order to view every process running in the machine, it is required to use the command ps ax:

The pstree program provides permission to view every process in a tree format. Thus, it provides a good idea of the hierarchical nature of processes. The following displays just a small part of the output of this program, starting with the init process (Figure 2.24):

The terminal also provides the top program. It provides permission to view the process list in an interactive way. It displays the process list in a similar manner to the GUI system monitor, and permits us to arrange the results by clicking on different keystrokes. We can click on the help key in order to obtain more help on the keys that are used for sorting the display.

Notes

Figure 2.24: Example of pstree Program

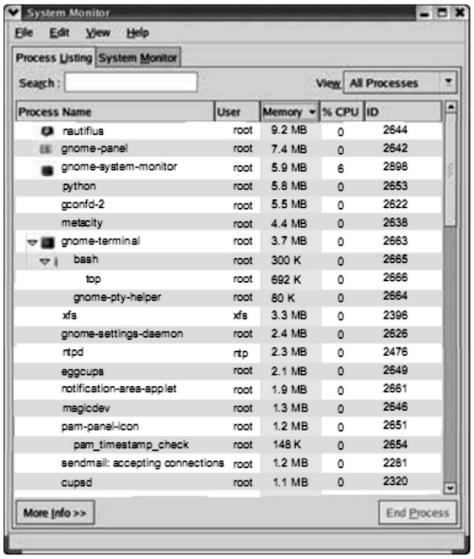
```
dthomas@localhost:~
File  Edit  View  Terminal  Tabs  Help
[dthomas@localhost dthomas]$ pstree
init─┬─acpid
     ├─aio/0
     ├─anacron───run-parts─┬─awk
     │                     └─slocate.cron───updatedb
     ├─atd
     ├─bonobo-activati
     ├─clock-applet
     ├─crond
     ├─cupsd
     ├─dbus-daemon-1
     ├─eggcups
     ├─events/0
     ├─gconfd-2
     ├─gdm-binary───gdm-binary─┬─X
     │                         └─gnome-session───ssh-agent
     ├─gimp───script-fu
     ├─gnome-keyring-d
     ├─gnome-panel
     ├─gnome-settings-
     ├─gnome-system-mo
     ├─gnome-terminal─┬─bash───su───bash
     │                ├─bash───pstree
```

Source: http://img.tebyan.net/library/english/13777_0248_0.jpg

2.6.3 Taking Control of Running Processes

In order to keep our CPU or memory resources in check, we sometimes need to use some control on a process. All the existing memory of the system might be taken by a process which will make the system almost unfeasible. In this type of situation, it might be suitable to terminate the offending process in order to recuperate system's control. System Monitor can be used to kill runaway processes. We can choose the target process in the list. Also, we can choose the End Process button at the bottom right of the dialog box. When we kill a process, it can lead to loss of unsaved data. Therefore, a confirmation dialog is there which allows you to verify your verdict to kill the process.

Figure 2.25: Confirmation Dialog



Notes

A process can be killed only if we are having permission for it, that is, if we attain permission to take control of the processes of another user or if it is running under your user account.

This is considered as a security aspect which makes sure that the processes of a user are secured from being ended by other users. The root, that is, super user is the only exception. As we can see in Figure 2.25, the user known as sandipb has full access and control over every process that can be seen in the process list.

The top program needs a signal number to send to the process.

A special message that is sent to the process by the existing process is known as a kill signal. On accepting the default option of 15, then a signal is sent to the target process which asks it to shut down on its own. This signal is known as SIGTERM.

Thus, the target process shuts down cleanly after saving all unsaved information. If it is observed that the program is neglecting this signal, the process can be repeated by sending the signal number of 9 rather. This will terminate the process and thus it will not get shut down on its own.

2.6.4 Understanding Run Levels

On starting the computer, the operating system loads and starts off the first process. This first process is known as the init process. Then the init process starts off any necessary sub-processes. This is done before the computer is used by the user. The operating system needs some processes to start off after the booting process. These processes are specified by means of **run levels**. We can define a run level as a state of the machine, which finds out the processes to be run.

We have discussed below various run levels which are numbered from 0 to 6:

- ***Run level 0:*** It indicates the stopped state of a machine. If we set the new run level to be 0 when changing run levels, then, it successfully stops the machine.
- ***Run level 1:*** It indicates the single user mode. At run level 1, the machine gets into a super user Mode. Thus external users cannot make use of the machine and every networking function gets disabled at this level. Another name for this level is the system maintenance mode as it is usually used to recuperate from severe system problems.
- ***Run level 2:*** At this level, numerous users are allowed to log in to the machine via virtual terminals and other login devices. However, it still does not make any networking function active.
- ***Run level 3:*** All the networking processes are allowed to be started at this level. Thus, all resources of the system can be used in an efficient manner.
- ***Run level 4:*** This level is considered as unused. We can use it to define your own custom run level.
- ***Run level 5:*** At this level, the X-server process is allowed to be started and the associated desktop can be loaded so that users can use the system with a GUI.
- ***Run level 6:*** It indicates the machine's rebooting state. We can make use of this level to restart the machine.
- The current run level of the computer can be determined by making use of the program run level. This provides two numbers, which symbolize the previous as well as current run levels.
- We can change the present run level by utilizing the program telinit. This program is required to be accomplished as root, and takes a single argument, that is, the run level to boot to.

Example: We can execute the command telinit 6 to reboot the system.

Notes

Self Assessment

Fill in the blanks:

12. A ..... is defined as an independent program entity which executes and makes use of computer resources like CPU time and memory.
13. A special message that is sent to the process by the existing process is known as a ....

2.7 Managing Users

Usually, a desktop is made to be utilized by people. Users that are familiar with the operating systems such as Windows 9x, Windows Me, etc. will consider Red Hat Linux a more severe organization of permissions that depict what every user can do and vice-versa. This is to note that the processes and data of one user cannot be used or altered by other users. This can be done only if they are allowed to do so.

2.7.1 Concept of User Management

As we know, the root of the computer can use all the resources of the computer. Thus, many security restrictions are there in the system in order to make sure that this user privilege is not compromised. A system administrator should also use system as a less privileged user. He should use super user privileges only when required.

Just one user can work physically at the desktop and other users can log in via network services such as telnet and ssh.

Linux provides the simple method of organizing users. Each user can be the part of one or more groups. Thus, as apart from protecting a resource by enforcing permissions to individual users, we can also put users into a group and enforce permissions to the group.

2.7.2 Understanding Permissions

As we know privileges are needed to control other user's processes; the same goes for files and directories. Each file as well as directory is connected with a single user and also with a group. There are three classes of users which are provided specific permissions to use a file:

- The owner
- The group members
- Everybody else

All the three classes of users are allowed a mixture of three types of access permissions:

Table 2.2: Three Types of Access Permissions

Permission	Meaning when applied to files	Meaning when applied to directories
Read	It allows the user to read the contents of the file	It allows the user to list the contents of the directory
Write	It allows the user to change the contents of the file	It allows the user to create or remove files within the directory
Execute	It allows the user to execute the file	It allows the user to access files as well as subdirectories

Notes

We can find the permissions of a given file by making use of the ls program.

Example: Here, we will use the command ls −l in order to display a "long" version of the file information:

$ ls −l /usr/bin/top

−r−xr−xr−x 1 root root 56406 Sep 1215:50 /usr/bin/top

We can represent the read, write, and executable permissions by making use of the characters r, w, and x respectively. The first part (for example, the expression −r−xr−xr−x, as shown in the above example) signifies the file permissions. The first character signifies the file type. This sequence comprises of three sets of three characters, that is, one set each to depict permissions of the owner, the group, and the others.

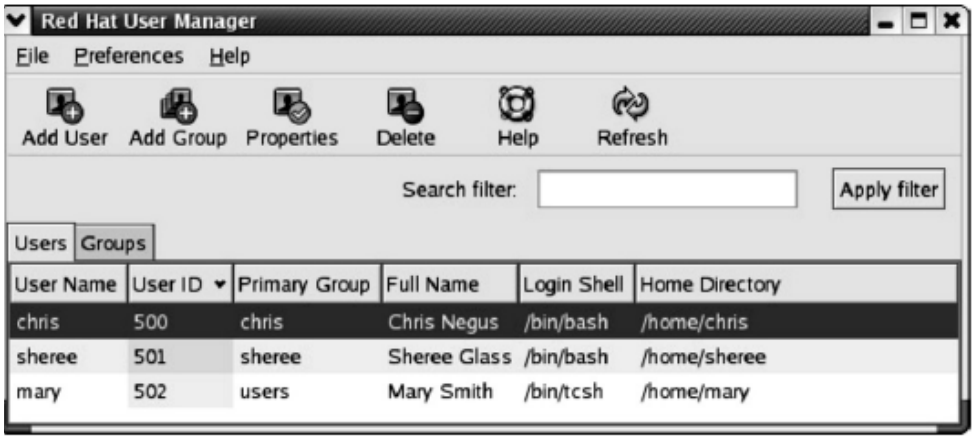
Example: The sequence r−x signifies that only the read and execution permissions are set. No write permission is available. Thus, this file permission specifies that every possible user can just read and execute the file content but not modify the contents of the file.

In order to modify the permissions of files and directories, we can make use of the CLI tools such as chmod, chown, and chgrp.

2.7.3 Concept of Managing Users

By means of the Red Hat user configuration tool, that is, "redhat−config−users", we can accomplish the task of adding, removing, and modifying user accounts on the system in a GUI. This tool can be launched by selecting Main Menu → System Settings → Users and Groups. In Figure 2.26, we have shown the primary interface of this tool.

Figure 2.26: Managing Users



It is required to provide the root password in order to use this utility. We can add, edit, or modify users and groups by making use of this interface.

We can perform the same operations in the CLI also. The different tools used are given below:

Notes

Table 2.3: Different Tools

Program	Purpose
useradd	Adding a user
userdel	Removing a user
chfn, chsh	Modifying user details

Self Assessment

Fill in the blanks:

14. A ..... should use system as a less privileged user.
15. The first character signifies the .....

2.8 Summary

- Red Hat Linux now makes use of the CUPS (Common UNIX Printing System) as the default printing system. It also includes the Native POSIX Thread Library, which provides improvements in performance with Pentium Pro processors or better.
- Applications that we can run from the Desktop are to be found either from the Main Menu (and sub-menus therein), or as icons on the Panel and the Desktop itself.
- 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.
- We can navigate the file system by means of either a GUI file explorer (for example, Nautilus) or the terminal.
- The installation process of Red Hat Linux automatically attempts to identify the existing hardware and configure it so as to use in the desktop.
- GNOME is a powerful but simple desktop environment with a strong focus on usability, accessibility, and internationalization.
- The GNOME Control Center allows you to configure various parts of your GNOME system with different tools called "capplets".
- A process is defined as an independent program entity which executes and makes use of computer resources like CPU time and memory.

2.9 Keywords

File system: A file system is an organization of data and metadata on a storage device.

GNOME Control Center: The GNOME Control Center allows you to configure various parts of your GNOME system with different tools called "capplets".

GNOME: GNOME is a powerful but simple desktop environment with a strong focus on usability, accessibility, and internationalization.

Panel: This sub-option configures the behavior of the panel in the desktop. You can opt to disable the animation of panels in the desktop or to change their animation speed.

Notes

Process: A process is defined as an independent program entity which executes and makes use of computer resources like CPU time and memory.

Run level: A run level is a state of the machine, which determines the processes to be run.

Sessions: This sub-option allows you to configure the behavior of the GNOME desktop for maintaining states between successive log-in sessions.

2.10 Review Questions

1. Discuss the Red Hat Linux desktop menus in details.
2. Explain the working of workspace switcher and how to use it.
3. Discuss the process of running program in Red Hat Linux.
4. How to use the panel in Linux? Discuss.
5. What are the advantages and disadvantages of CLI?
6. Explain the using process of the file system hierarchy.
7. What do you understand by navigating in the file system?
8. How to change the current directory in Linux?
9. Find the information of any existing hardware device in Red hat.
10. Explain the process of configuring the desktop.

Answers: Self Assessment

1. sub-menu.
2. Panel
3. Applets
4. Launchers
5. Links
6. /sbin
7. hardware browser
8. Superuser
9. GConf
10. Caplets
11. Desktop
12. Process
13. kill signal
14. system administrator
15. file type

2.11 Further Readings

Books

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

Notes

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

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

Online links

http://www.redhat.com/mirrors/LDP/LDP/intro-linux/intro-linux.pdf

http://www.linuxtopia.org/online_books/linux_beginner_books/redhat_9_getting_started_guide/ch-basics.html

http://redhat.activeventure.com/9/customizationguide/ch-intro.html

http://searchitchannel.techtarget.com/feature/Managing-Linux-hardware-and-the-kernel-Introduction

Notes

Unit 3: Connecting to the Internet

CONTENTS

Objectives

Introduction

3.1 Connecting to the Internet

3.1.1 Networking Overview

3.1.2 Configuring an Internet Connection

3.2 Managing Multiple ISPs and Connections

3.2.1 Outbound Traffic Using Multiple Connections to the Internet

3.2.2 Inbound Traffic Using Multiple Connections to the Internet

3.3 Software and Configuration

3.3.1 The Mozilla Web Browser

3.3.2 gFTP Client

3.4 Summary

3.5 Keywords

3.6 Review Questions

3.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Describe the concept of using Red Hat Linux to connect to the Internet
- Discuss managing multiple ISPs and connections
- Explain the concept of software and configuration

Introduction

In this unit, we will discuss how to use Red Hat Linux to connect to the Internet via a telephone dialup, ISDN (Integrated Services Digital Network), DSL (Digital Subscriber Line), Ethernet, or wireless connection. We will give an overview of networking and discuss how to use the Network Administration Tool. Network Administration Tool is an X-based program that makes it easy to connect to the Internet via an Internet Service Provider (ISP). Also, we will discuss several popular network client applications available under Linux, including the Mozilla web browser, the email client, and a graphical FTP client.

3.1 Connecting to the Internet

In this section, we will discuss how to use Red Hat Linux to connect to the Internet.

3.1.1 Networking Overview

Nowadays, most computers handle network traffic much like the post office handles mail.

Notes

Example: Consider the steps included in sending as well as receiving a letter. Your postal carrier must know where to drop off and where to pick up mail. So your home must have some kind of recognizable interface; we call this a mailbox. And whereas your postal carrier may know your neighborhood quite well, delivery in other areas will require other carriers. Mail is passed to these other carriers through a gateway; we call this the post office. Even though you can think of the whole postal system as one big network, it's easier to recognize if you consider it as a hierarchy of subnetworks (or subnets): the postal system is divided into states, states are divided into counties and cities with a range of Zip Codes, Zip Codes contain a number of streets, and each street contains a unique set of addresses.

Computer networking mirrors this model. Let us trace an email message from you to a workfellow. After composing the message, click Send. The message is passed to a network interface by computer. This interface may be a modem by which you dial up an Internet Service Provider (ISP), or it may be via an Ethernet connection on a LAN. Either way, on the other side of the interface is a gateway machine. The gateway knows how to look at the address of the recipient of the email message and interpret that message in terms of networks and subnets. By means of this information, the gateway passes the message to other gateways until the message reaches the gateway for the destination machine. That gateway in turn delivers the message through a recognizable interface (like a modem or Ethernet link) to the recipient's inbox.

On reviewing this, it can be easily seen which parts of networking are required to configure on your Linux system. It is required to know the address of your machine. Just as the town name *Sebastopol* and the Zip Code *95472* are two different names for the same location, you may have both a name, called a *hostname*, and a number, called an *IP number* or *IP address*, that serve as the address for your machine. IP represents Internet Protocol.

To translate among these two notations, it is necessary to know the address of a DNS (Domain Name Server). This is a machine that matches IP addresses with hostnames. It is also necessary to know the address of a gateway machine through which network traffic will be routed. Finally, you will need to be able to bring up a network interface on your system, and you will need to assign a route from that interface to the gateway.

While all of this can seem difficult, it actually is not any more difficult than the postal system, and it functions in much the same way. Luckily, Linux comes with tools to assist you in automating network configuration.

3.1.2 Configuring an Internet Connection

The configuration of your system is simplified by the Red Hat Linux Network Administration Tool in order to access the Internet through a telephone dialup, DSL, ISDN, Ethernet, or wireless connection. The Network Administration Tool needs you to follow a three-step process:

1. Set up the hardware device associated with the connection.
2. Specify DNS settings and hostnames.
3. Activate the device, if necessary.

Now we will explain how to perform these steps.

Notes All these methods of connecting to the Internet are supported by the Network Administration tool. However, some hardware devices are not compatible with Red Hat Linux. And, some Internet service providers insist that their customers use only Windows. In either case, you can experience difficulties in connecting to the Internet.

Setting up Hardware Devices

Notes

Earlier, most computer users were connected to the Internet through a plain old telephone service (POTS) dial-up modem. However, nowadays, there are various methods of connecting to the Internet.

Example: Many home computer users have high-speed connections using ISDN or DSL.

Did u know? The users of corporate computer often connect to the Internet through their local area network, by means of an Ethernet adapter.

In order to set up a hardware device by means of the Network Administration Tool, login as root and select System Settings → Network from the GNOME or KDE menu. The Network Administration Tool appears, as shown in Figure 3.1:

Figure 3.1: The Network Administration Tool

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

The Network Administration Tool consists of four tabs:

- *Devices:* This tab *is* used to associate a physical device with a network connection
- *Hardware:* This tab *is u*sed to set up a physical device
- *Hosts:* This tab is used to specify names of hosts not known to a DNS server
- *DNS:* This tab is used to specify DNS servers and related options

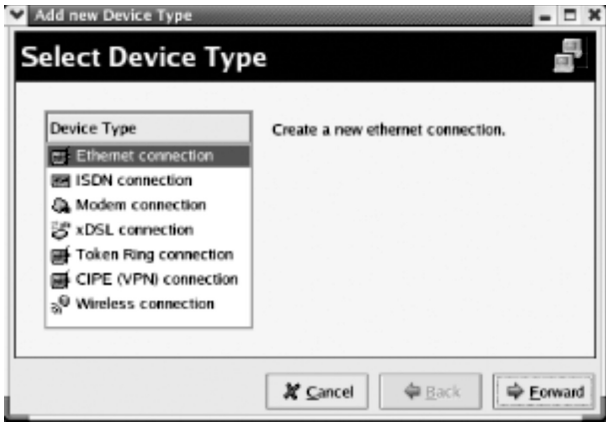
To start setting up an Internet connection, click the Add button of the Device tab. It will give a dialog box which invites you to select the type of your device, as shown in Figure 3.2. Choose the appropriate device type and click Forward. Then, follow the steps in the following subsection suitable to the type of your device.

Notes Red Hat Linux supports two schemes utilised by DSL providers: PPPoE (Point-to-Point Protocol over Ethernet) and DHCP (Dynamic Host Control Protocol).

Notes

Caution If your DSL provider uses DHCP, you should configure your DSL connection as though it were an Ethernet connection. Otherwise, you should specify xDSL as the device type.

Figure 3.2: The Add New Device Type Dialog Box

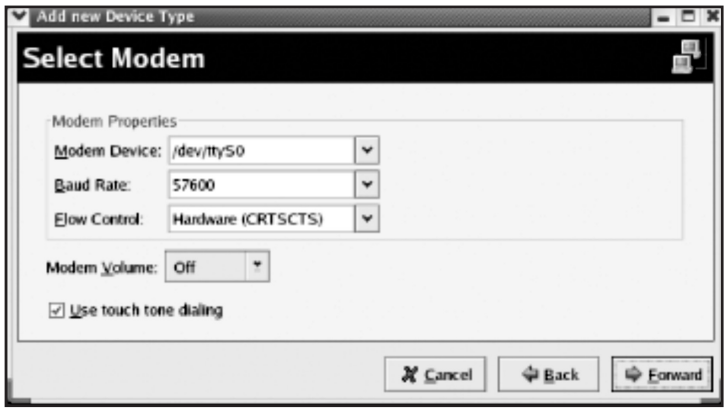


Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

Setting up a Dialup Modem

When a Modem is specified as the device type, your modem is probed by the Network Administration Tool. This process may take some seconds. Once the Network Administration Tool locates the modem, it displays the Select Modem dialog box, shown in Figure 3.3, which lets you specify modem characteristics. Generally, the defaults are acceptable. However, check the documentation for your modem to be sure. If your phone line does not support touch tone dialing, de-select the Use touch tone dialing checkbox. Then, click Forward to continue.

Figure 3.3: The Select Modem Dialog Box

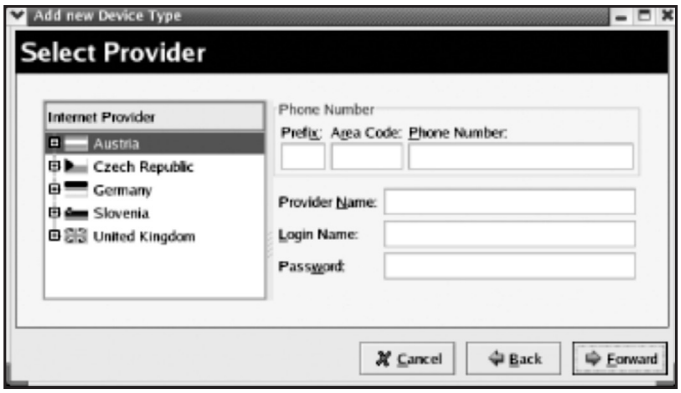


Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

As shown in Figure 3.4, the Select Provider dialog box appears. If your country and provider are listed, choose them. Or else, Specify the phone number, name, login name, and password associated with your account. Then click Forward to continue.

Notes

Figure 3.4: The Select Provider Dialog Box



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

This will show the main Tool screen again. However, this time the screen includes a line identifying your modem as a ppp (Point-to-Point Protocol) device, as shown in Figure 3.5. Click Apply to save your changes.

Figure 3.5: The Network Administration Tool

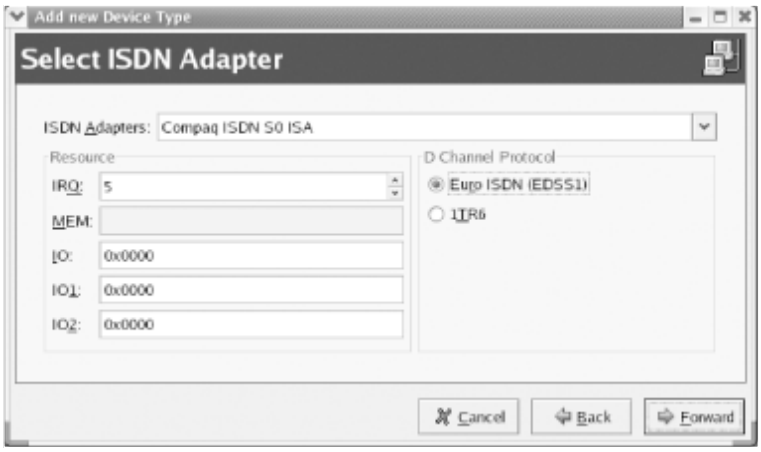
Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

Setting up an ISDN Modem

When an ISDN modem is specified as the device type, the Network Administration Tool gives a list of supported ISDN modems, as shown in Figure 3.6. Specify the device characteristics and select the D Channel Protocol utilised by your ISDN provider. Click Forward to continue.

Notes

Figure 3.6: The Select ISDN Adapter Dialog Box



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

As shown in Figure 3.4, the Select Provider Dialog Box appears. If your country and provider are listed, choose them. Else, specify the phone number, name, login name, and password associated with your account. Then, click Forward to continue. The main Tool screen reappears. However, this time the screen includes a line identifying your ISDN modem as an ippp (ISDN Point-to-Point protocol) device. Click Apply to save your changes.

Setting up an xDSL Modem

Numerous varieties of DSL are in use, including SDSL, IDSL, and ADSL. The xDSL device type supports every variety. On specifying DSL modem, the Network Administration Tool presents the Configure DSL connection dialog box, as shown in Figure 3.7. Specify the Ethernet device associated with your DSL link and the login name and password associated with your DSL account. Click Forward to continue. The main Tool screen reappears. However, this time the screen includes a line identifying your DSL modem as a Point-to-Point Protocol (PPP) device. Click Apply to save your changes.

Figure 3.7: The Configure DSL Connection Dialog Box

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

Notes

Task Identify several varieties of DSL and their uses.

Setting up an Ethernet Adapter

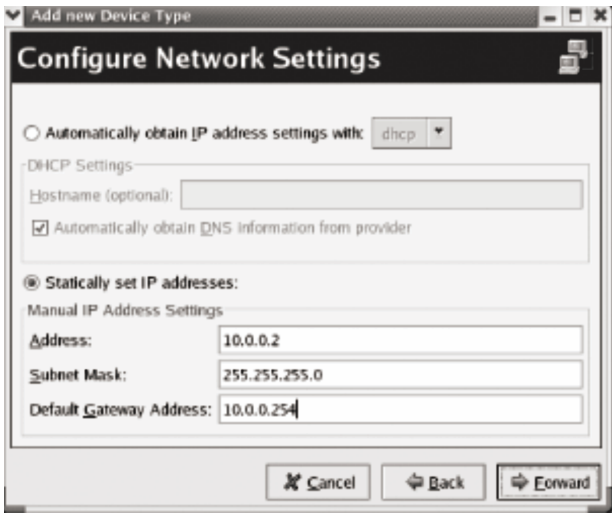
On selecting Ethernet adapter as the device type, the Network Administration Tool probes your system for supported Ethernet adapters and displays a list of the adapters it finds, as shown in Figure 3.8. Select the adapter you want to configure and click Forward.

Figure 3.8: The Select Ethernet Device Dialog Box

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

As shown in Figure 3.9, the Configure Network Settings dialog box appears. If your ISP provides a DHCP server that supplies your system with its network configuration, enable the Automatically obtain IP address settings with radio button and select DHCP from the drop-down list. Also, enable the Automatically obtain DNS information from provider checkbox. If your provider uses BOOTP, which is now unusual, select BOOTP from the drop-down list.

Figure 3.9: The Configure Network Settings Dialog Box



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

Notes

If your provider needs you to specify the network configuration of your system manually, enable the Statically set IP address radio button. Then, specify the IP address, subnet mask, and default gateway address as directed by your network administrator. Click Forward to continue. The main Tool screen reappears. However, this time the screen includes a line identifying your Ethernet adapter as an eth device. To save your changes, click Apply.

Setting up a Wireless Adapter

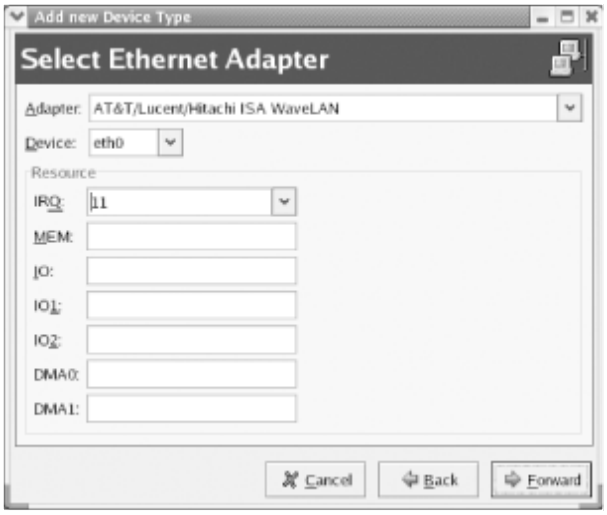
On selecting Wireless adapter as the device type, the Network Administration Tool probes your system for supported wireless adapters and displays a list of the adapters it finds, as shown in Figure 3.10. Choose the appropriate adapter and click Forward.

Figure 3.10: The Select Wireless Device Dialog Box

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

If the adapter is not specifically identified, select Other Wireless Card and click Forward. As shown in Figure 3.11, the Select Ethernet Adapter dialog box appears Select the suitable adapter and specify its characteristics. Click Forward to continue.

Figure 3.11: The Select Ethernet Adapter Dialog Box

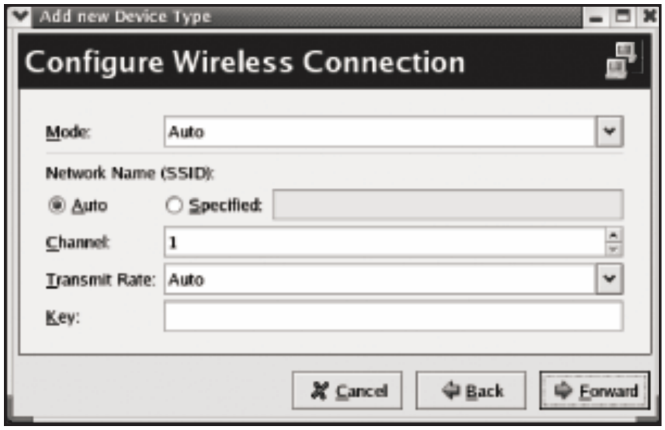


Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

As shown in Figure 3.12, the Configure Wireless Connection dialog box appears. Specify the mode (Managed or Ad Hoc) in which your wireless access point operates or specify Auto to configure the adapter to use whatever mode the access point uses.

Notes

Figure 3.12: The Configure Wireless Connection Dialog Box



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

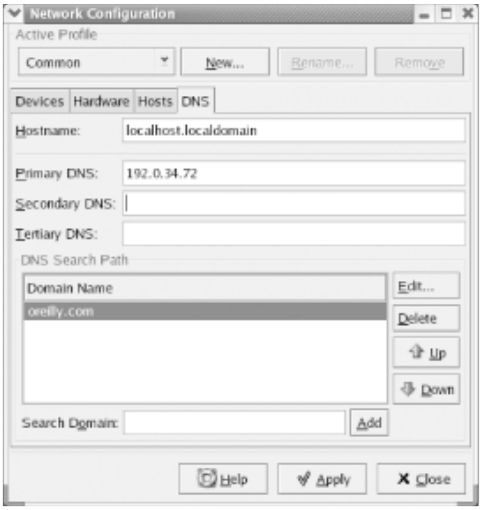
Caution Access points using Ad Hoc mode are susceptible to different types of attack. Usually, it is best to configure your access points to use Managed mode and specify Managed rather than Auto as the adapter mode.

Specify the Service Set Identifier (SSID) of your network or specify Auto if you want to connect to any available network. Then, specify the channel, transmit rate, and key (if any) associated with your network. Click Forward to continue. The main Tool screen reappears. However, this time the screen includes a line identifying your Ethernet adapter as an eth device. Click Apply to save your changes.

Specifying DNS Settings and Hostnames

The DNS tab allows you to configure DNS. However, if your ISP offers DNS information via DHCP, you are not required to do so.

Figure 3.13: The DNS Tab of the Network Configuration Tool



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.2+Configuring+an+Internet+Connection/

Notes

Generally, Internet hosts are known by both an IP address as well as hostname. The function of DNS is to translate hostnames to IP addresses and IP addresses to hostnames. Translating a hostname to an IP address is known as *hostname lookup* or *address resolution*. Translating an IP address to a hostname is known as *reverse lookup*. DNS is important, but not essential. For instance, without DNS, you'd have to type IP addresses instead of hostnames when browsing the Web. Doing so would be inconvenient, but workable. However, you wouldn't be able to simply click hyperlinks specified using hostnames. As an alternative, you need to somehow discover and type the proper IP address.

By means of the tab, the hostname of your system, and the IP addresses of primary, secondary, and tertiary DNS servers can be specified. Also, one or more domain names that are automatically added to hostnames when performing hostname lookups can also specified.

Example: If you access hosts in the *sample.com* domain on a frequent basis, you can include *sample.com* on the search path. Then, the host *www.sample.com* can be referred as simply *www*.

Task Differentiate between hostname lookup and reverse lookup.

Activating the Device

Some network devices, like Ethernet adapters, automatically gets activated by the Network Administration Tool. Other network devices, like dial-up modems, must be manually activated and deactivated. To activate a device, select its name in the Devices tab of the Network Administration Tool and click Activate. If the device is a dial-up modem, it will try to connect to your ISP.

After establishing a connection, you should be able to access the Internet. Make an attempt to ping an Internet host by issuing a command like ping www.linux.com in a terminal window.

Terminate the Ping Command by Typing Ctrl-C

If the command doesn't work, maybe your connection isn't working. Or possibly your ISP's DHCP server failed to properly offer DNS information. Try pinging the IP address of a host you know to be available.

Example: Issue a command such as:

$ ping -n 66.187.232.56

If pinging the IP address works, just use the DNS tab to revise your DNS configuration and you're set. Otherwise, you may have some difficulty getting the connection to work.

Make use of the **ifconfig** and **route** commands to view your network configuration. If you can discover the problem, again you're set. If not, you may be able to get help from your ISP or from participants in an Internet newsgroup, for example, *linux.redhat* or *linux.redhat.misc*.

In order to terminate a connection, deactivate the associated device by clicking Deactivate on the Devices tab.

Notes

Self Assessment

Fill in the blanks:

1. ..... is a machine that matches IP addresses with hostnames.
2. The tab "....." is used to associate a physical device with a network connection.
3. When you specify an ISDN modem as the device type, the ..... Tool presents a list of supported ISDN modems.
4. Specify the ..... of your network or specify Auto if you want to connect to any available network.
5. The DNS tab of the ..... Tool lets you configure DNS.
6. Translating a hostname to an IP address is called .....
7. Translating an IP address to a hostname is called .....
8. Using the tab, you can specify the hostname of your system, and the ..... of primary, secondary, and tertiary DNS servers.

3.2 Managing Multiple ISPs and Connections

There are various places where multiple Internet connections can be managed by a linux based router/masquerading device. Here, we will discuss some of the more common setups concerning multiple Internet connections and how to manage them with iptables, ipchains, and iproute2. One of the first distinctions you can make when planning how to use multiple Internet connections is what inbound services you expect to host and how you want to split traffic over the multiple links.

Now we will discuss the issues involved with two separate uplinks to two different providers. Let us assume the following:

- You are not using BGP, and you do not have your own AS. If you are using BGP and have your own AS, you have a different set of problems than the problems described here.
- You have two netblocks from two different ISPs.
- You are funneling your internal network through this routing device, which is performing masquerading/NAT to the Internet.

3.2.1 Outbound Traffic Using Multiple Connections to the Internet

There are two major uses for multiple Internet links connected to the same internal network. Selecting an outbound link based on the type of outbound service is the one common use. The other is to split traffic arbitrarily across multiple ISPs for reasons like failover and to accommodate greater aggregate bandwidth than would be available on a single uplink.

Here, we will discuss how to classify traffic for different ISPs, how to handle the packet filtering for this sort of classification scheme, and how to create routing tables appropriate for the task at hand.

The simplest way to split Internet access into two separate groups is by source IP of the outbound packet. This can be done most simply with ip rule and a second routing table.

Example: Assume that masq-gw in the example network gets a second, low cost network connection through a DSL vendor.

Notes

The DSL IP on masq-gw will be 67.17.28.12 with a gateway of 67.17.28.14. Let us suppose that this is for outbound connectivity only, and that the IP is active on eth4 of the masq-gw machine.

Before starting, let us outline the process:

- Copy the main routing table to another routing table and set the alternate default route.
- Use iptables/ipchains to mark traffic with fwmark.
- Add a rule to the routing policy database.
- Test!

3.2.2 Inbound Traffic Using Multiple Connections to the Internet

There are many different methods to handle hosting servers to multiple ISPs. If you are in requirement of this sort of advanced networking, you probably already know where to research. If not, it is recommended to start your research in load balancing, global load balancing, failover, and layer 4-7 switching. These are networking tools which can facilitate the management of a highly available service.

Publishing the same service on two dissimilar ISPs is can be a tough challenge. While this is possible using some of the advanced networking characteristics under linux, one should recognize the greater issues involved with publishing a service on two public IPs, especially if the idea is to provide service to the general Internet even though one of the ISPs go down. If you are aware of the various difficult issues involved in managing inbound connections to a network, and still want to publish a service on two dissimilar ISPs, you'll find the recipe below.

Before we examine the recipe, let's look at a complex scenario to see what the crucial points are. One other item to remember is that routing decisions are stateless.

We will assume that the client IP is a fixed IP (64.70.12.210) and we'll discuss how this client IP would reach each of the services published on masq-gw's two public networks. The IPs used for the services will be 67.17.28.10 and 205.254.211.17. Now, whether you are using NAT with iproute2 or with iptables, you'll run across the problem here outlined. Here is the flow of the packet through masq-gw to the server and back to the client.

Inbound NAT to the Same Server Via Two Public IPs in Two Different Networks

1. inbound packet from 64.70.12.210 to 67.17.28.10 arrives on eth4
2. packet is accepted, rewritten, and routed; from 64.70.12.210 to 192.168.100.17; if iptables DNAT, packet is rewritten in PREROUTING chain of nat table, then routed; if iproute2, packet is routed and rewritten simultaneously
3. rewritten packet is transmitted out eth0
4. isolde receives packet, accepts, responds
5. inbound packet from 192.168.100.17 to 64.70.12.210
6. routing decision is made; default route (via 205.254.211.254) is selected; if iproute2 is used, packet is also rewritten from 67.17.28.10 to 64.70.12.210
7. if iptables DNAT is used, connection tracking will take care of rewriting this packet from 67.17.28.10 to 64.70.12.210
8. packet is transmitted out eth1

This is the problem! The packet may have the right source address, however it is leaving through the wrong interface. Many ISPs filter traffic entering their network and will block traffic from your network with source IPs outside your allocated range. To an ISP this looks like faked traffic.

Notes

The solution is amazingly simple and elegant. Choose one IP on the internal server which will be reachable through one provider and one IP which will be reachable through the other provider. By using two IP addresses on the internal machine, we can use ip rule on masq-gw to choose a routing table with a different default route based upon the source IP of the response packets to clients.

Self Assessment

State whether the following statements are true or false.

9. The simplest way to split Internet access into two separate groups is by source IP of the outbound packet.
10. Publishing the same service on two similar ISPs can be formidable challenge.

3.3 Software and Configuration

Now, we will discuss the tools provided by Linux.

3.3.1 The Mozilla Web Browser

After establishing a connection to the Internet, you can surf the Web by means of Mozilla, the default Red Hat Linux web browser. To launch Mozilla, select Internet → Web Browser from the GNOME or KDE menu. Figure 3.14 shows Mozilla. Mozilla corresponds to its closed source ancestor, Netscape Navigator. Thus, if you have used Navigator, you will feel at home in Mozilla.

Figure 3.14: The Mozilla Web Browser

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.3+The+Mozilla+Web+Browser/

When launching Mozilla for the first time, a dialog box may be seen explaining that a Mozilla profile is being created from existing Netscape 4 files. Click Convert Profile to allow Mozilla to create the profile.

Notes

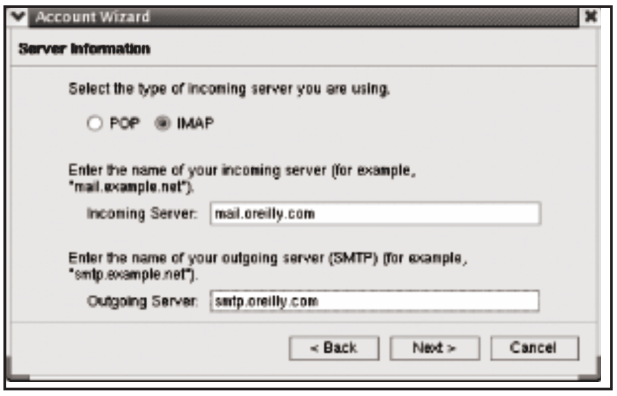
Mozilla includes email and news clients that are easily configured. It also includes a web page composer and address book. To configure Mozilla email, select Window → Mail & Newsgroups from the Mozilla menu. An Account wizard appears, asking whether you want to configure an email or newsgroup account. Choose Email Account and click Next. The Identity dialog box appears, as shown in Figure 3.15. Specify your name and email address and click Next.

Figure 3.15: The Identity Dialog Box

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.3+The+Mozilla+Web+Browser/

As shown in Figure 3.16, the Server Information dialog box appears. Specify which protocol your mail server uses, Post Office Protocol (POP) or Interim Mail Access Protocol (IMAP). POP servers require you to download email. IMAP servers let you read email that resides on the server. Most up-to-date ISPs support IMAP, the newer protocol. Also specify the hostnames of your incoming and outgoing mail services. A single host may fill both roles. If you don't know this information, you can obtain it from your ISP. Click Next to continue.

Figure 3.16: The Server Information Dialog Box



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.3+The+Mozilla+Web+Browser/

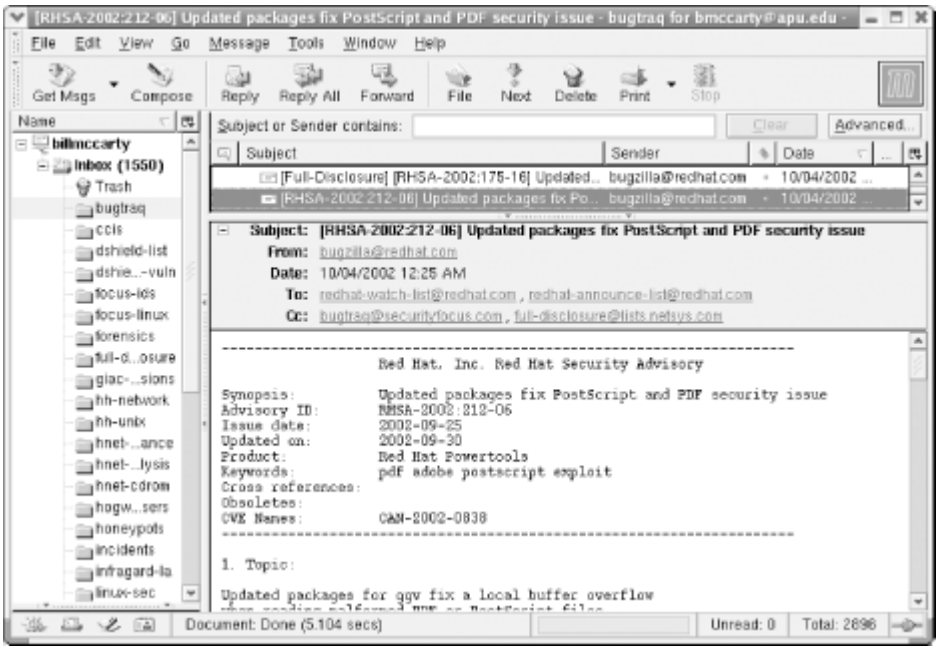
The User Name dialog box appears. Specify the username by which you login to your ISP's mail server. Click Next to continue. The Account Name dialog box appears. Specify a name by which to refer to your new email account so that you can distinguish it from other accounts you may create. Click Next to continue. The Congratulations! dialog box appears. Verify that the information it presents is accurate. If necessary, click Back to return to an earlier dialog box and correct the information. When you're satisfied that the information is correct, click Next to continue.

Notes

In order to login to your email account, Mozilla asks your password. Specify the password. Enable the checkbox "Use Password Manager" to remember this password. Click OK to login.

As shown in Figure 3.17, Mozilla reads and displays the contents of your Inbox. Window's left pane displays your mail folders. The top pane displays messages within the current folder. And, the bottom pane displays the contents of the currently selected message.

Figure 3.17: The Mozilla Email Screen



Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.3+The+Mozilla+Web+Browser/

Mozilla's user interface is straightforward.

Example: To send a message, click the Compose button in the toolbar.

The best technique to learn your way around Mozilla is to explore its menu and toolbar, clicking and observing the results.

3.3.2 gFTP Client

Web browser can be used to download files from an FTP server. However, you need an FTP client to upload files. The gFTP client, included with Red Hat Linux, is an excellent choice. This is because its user interface resembles that of popular Windows FTP clients, such as WS-FTP. Figure 3.18 shows the gFTP client, which can be launched by choosing Extras → Internet → gFTP from the GNOME or KDE menu.

In order to connect to a remote system, specify the hostname, username, and password in the textboxes appearing on the toolbar. On getting the permission of anonymous logins by server, the username and password can be omitted. To connect, click the Connect icon resembling a pair of computers at the left of gFTP's toolbar. To upload a file, click on the name of the file in the local list box at the left of the window and then click on the right-pointing arrow. To download a file, click on the name of the file in the list box at the right of the window and then click on the left-pointing arrow. When you've transferred all your files, select Remote → Disconnect or click again on the Connect icon.

Notes

Figure 3.18: The gFTP FTP Client

Source: http://etutorials.org/Linux+systems/red+hat/Chapter+10.+Connecting+to+the+Internet/10.4+gFTP+FTP+Client/

Did u know? You can access an FTP server in command-line mode, if you prefer.

Self Assessment

Fill in the blanks:

11. Mozilla resembles its closed source ancestor, ...
12. Click Convert Profile to allow Mozilla to create the .....
13. ..... servers require you to download email.
14. ..... servers let you read email that resides on the server.
15. You can use your web browser to download files from an FTP server, but to upload files you need an .....

3.4 Summary

- Domain Name Server (DNS) is a machine that matches IP addresses with hostnames.
- The Red Hat Linux Network Administration Tool simplifies configuration of your system to access the Internet via a telephone dialup, ISDN, DSL, Ethernet, or wireless connection.
- Red Hat Linux supports two schemes used by DSL providers: PPPoE (Point-to-Point Protocol over Ethernet) and DHCP (Dynamic Host Control Protocol).
- Several varieties of DSL are in use, including IDSL, ADSL, and SDSL. The xDSL device type supports each variety.
- Internet hosts are generally known by both an IP address and hostname. DNS translates hostnames to IP addresses and IP addresses to hostnames.

Notes

- Translating a hostname to an IP address is called hostname lookup or address resolution. Translating an IP address to a hostname is called reverse lookup.
- Mozilla includes email and news clients that are easily configured. It also includes a web page composer and address book.
- You can use your web browser to download files from an FTP server, but to upload files you need an FTP client. The gFTP client, included with Red Hat Linux, is an excellent choice, because its user interface resembles that of popular Windows FTP clients, such as WS-FTP.

3.5 Keywords

DNS: Domain Name Server (DNS) is a machine that matches IP addresses with hostnames.

Hardware: It is used to set up a physical device.

Hostname lookup: Translating a hostname to an IP address is called hostname lookup or address resolution.

Hosts: Hosts are used to specify names of hosts not known to a DNS server.

IP address: IP address serve as the address for your machine.

ISP: An ISP (Internet service provider) is a company that provides individuals and other companies access to the Internet and other related services such as Web site building and virtual hosting.

PPP: PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server.

Reverse lookup: Translating an IP address to a hostname is called reverse lookup.

3.6 Review Questions

1. Describe how to use Red Hat Linux to connect to the Internet.
2. Elucidate the use of Domain Name Server.
3. Describe the steps included in configuring an internet connection.
4. Discuss the tabs included in Network Administration Tool.
5. Make distinction between PPPoE (Point-to-Point Protocol over Ethernet) and DHCP (Dynamic Host Control Protocol).
6. Illustrate how to specify DNS Settings and Hostnames.
7. Describe the steps included in activating the device.
8. Explain the concept of managing multiple ISPs and connections.
9. Mozilla resembles its closed source ancestor, Netscape Navigator. Comment.
10. Discuss the process of launching gFTP client.

Answers: Self Assessment

1. DNS
2. Devices
3. Network Administration
4. SSID (Service Set Identifier)
5. Network Configuration
6. hostname lookup
7. reverse lookup
8. IP addresses

Notes

9. True
10. False
11. Netscape Navigator.
12. Profile
13. POP
14. IMAP
15. FTP client

3.7 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.northernjourney.com/opensource/newbies/newb008.html

http://www.isdnllc.com/tech/linux/

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Step_by_Step_Guide/ch-connect.html

http://www.linux.ie/newusers/beginners-linux-guide/ppp.php

Unit 4: Installing Software

Notes

CONTENTS

Objectives
Introduction
4.1 The Red Hat Package Manager (RPM)
4.2 Benefits of RPM
4.2 The RPM Command Line Tools
4.3.1 Installing a New Package
4.3.2 Querying a Package
4.3.3 Uninstalling a Package
4.3.4 GnoRPM
4.3.5 Compiling Software
4.3.6 Getting and Unpacking the Package
4.3.7 Looking for Documentation
4.3.8 Configuring the Package
4.3.9 Compiling Your Package
4.3.10 Installing the Package
4.4 Summary
4.5 Keywords
4.6 Review Questions
4.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain the concept of RPM
- Discuss benefits of RPM
- Describe RPM command line tool

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 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

Notes

library API, permitting advanced developers to manage such transactions from programming languages such as C or Python.

4.1 The Red Hat Package Manager (RPM)

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 customisations — something that you will not accomplish with regular .tar .gz files.

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.

Notes 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. The RPM package contains a complete version of the program, which overwrites existing versions or installs as a new package.

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.

Self Assessment

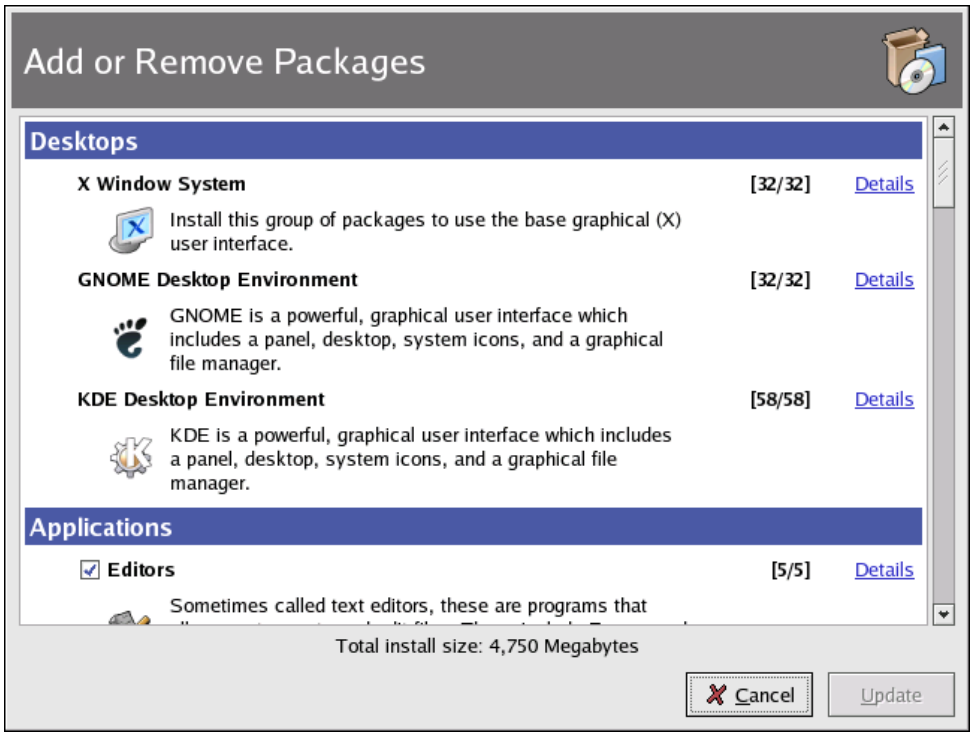
Fill in the blanks:

1. An ..... package consists of an archive of files and meta-data used to install and erase the archive files.

Notes

2. For the ....., RPM makes system updates easy.
3. The ..... includes helper scripts, file attributes, and descriptive information about the package.

Figure 4.1: RPM



4.2 Benefits of RPM

The benefits of RPM are discussed below:

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.

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.

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.

Notes

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.

Did u know? The goal of keeping sources pristine may only seem important for developers, but it results in higher quality software for end users, too.

Self Assessment

Fill in the blanks:

4. Any configuration files that you modified are preserved during .....
5. A crucial design goal was to allow the use of "....." software sources, as distributed by the original authors of the software.

4.3 The RPM Command Line Tool

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.

4.3.1 Installing a New Package

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.

Example: In the example given below, 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.

Notes

Example: # rpm -I penguin-3.26.i386.rpm

warning: /etc/someconfig saved as /etc/someconfig.rpmsave

It's up to you to look at both files and determine what modifications, if any, need to be made.

4.3.2 Querying a 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

Notes

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

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

4.3.3 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 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.

Caution 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.

Notes

4.3.4 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

- Install RPM packages
- Uninstall RPM packages
- Upgrade RPM packages
- Find new RPM packages
- Query RPM packages
- 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.

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.

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.

Did u know? Because of Gnome-RPM's integration with GNOME, you can also perform installation, query and verification on packages from within the GNOME File Manager.

Notes

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 &

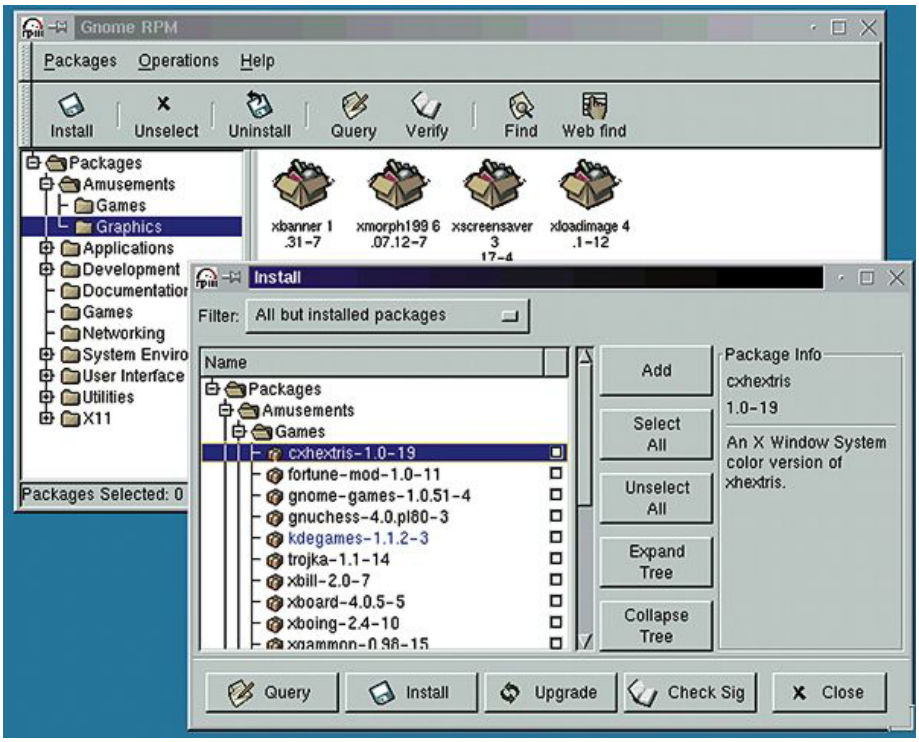
That will bring up the main Gnome-RPM window.

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.

Figure 4.2: Gnorpm



4.3.5 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. 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.

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. Some packages might need some additional libraries or might suffer from dependency issues, so it makes sense to know what's needed.

Configuring

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.

Notes

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. 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.

4.3.6 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.

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.

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 ungzip it before you untar it.

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 ungzip 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.

Notes

Task Compare contrast Zipped compressions and Gzipped compressions.

4.3.7 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.

4.3.8 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.

Caution Make sure that the configure command did indeed complete without any errors.

4.3.9 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.

Notes

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

Notes 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. 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.

4.3.10 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

Notes

deb packages are designed for Debian Linux. The apt system is the simplest way to retrieve and install packages for Debian Linux.

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.

Task Make distinction between .rpm and deb.

Self Assessment

Fill in the blanks:

6. The ..... mode is used to install RPM packages onto your system.
7. The -q switch is used to ..... packages.
8. The RPM ..... command removes a package from your system.
9. ..... is a graphical front end to the RPM package management system.
10. All the source files, associated libraries and documentation are distributed as compressed archives called .....
11. The ..... command uses the Makefile to create installable binaries.
12. ..... option allows you to set the base directory where the package gets installed.
13. ..... is used by Debian Linux and distributions based on Debian.
14. ..... is a graphic front end to dpkg, and can be called by running the command dselect from the command line.
15. To unbzip a file, we run the command .....

Notes

4.4 Summary

- RPM, the Red Hat Package Manager, is a powerful package manager that you can use to install, update and remove packages.
- RPM maintains a database of installed packages and their files, so you can invoke powerful queries and verifications on your system.
- The install mode, as its name suggests, is used to install RPM packages onto your system.
- RPM accounts for every system or application file on your system.
- Gnome-RPM (which is also referred to as gnorpm) is a graphical front end to the RPM package management system.
- 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.
- Most developers provide README or INSTALL files in the program archive. These are text files that include instructions on compiling and installing the program.
- When a package is installed, the installer checks whether the other files the program will need are present. Each installer handles missing files differently.

4.5 Keywords

apt: apt is a system for installing packages.

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.

Install mode: The install mode, as its name suggests, is used to install RPM packages onto your system.

Makefile: It is a script which will compile the program automatically.

README or INSTALL files: These are text files that include instructions on compiling and installing the program.

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.

4.6 Review Questions

1. Explain the concept of RPM Package Manager.
2. "The Red Hat Package Manager (RPM) is an open packaging system". Comment.
3. Explain the benefits of Red Hat Package Manager (RPM).
4. What is GnoRPM? Describe the working of GnoRPM.
5. Explain how to start Gnome-RPM from an Xterm window.
6. Describe the several parts to the Gnome-RPM interface.

Notes

7. Illustrate how to use the query mode of RPM.
8. Briefly explain Linux developer's utility "make".
9. Discuss the process of Upgrading RPM packages using GnoRPM.
10. Illustrate the use of README or INSTALL files in the program archive.

Answers: Self Assessment

1. RPM
2. end user
3. meta-data
4. Reinstallation
5. pristine
6. Install
7. query
8. -e
9. Gnome-RPM
10. Tarballs
11. make
12. -prefix
13. .deb
14. Dselect
15. bunzip2 (file name)

4.7 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.control-escape.com/linux/lx-swinstall.html

http://www.codecoffee.com/tipsforlinux/articles/27.html

http://www.tldp.org/HOWTO/Software-Building-HOWTO.html

http://polishlinux.org/first-steps/installing-software/

Notes

Unit 5: Everyday Applications-I

CONTENTS

Objectives

Introduction

5.1 Office Applications

5.1.1 Open Office

5.1.2. KDE KOffice

5.1.3 The Gnome Office

5.1.4 Writing with OpenOffice.org Writer

5.1.5 Preparing Spreadsheets with OpenOffice.org Calc

5.1.6 Presenting Information with OpenOffice.org Impress

5.1.7 Using Calculators

5.2 Internet Applications

5.2.1 Mozilla Web Browser

5.2.2 Mail Client

5.2.3 Instant Messaging with Gaim

5.3 Personal Information Management

5.3.1 Evolution

5.3.2 Starting Evolution for the First Time

5.3.3 Using Evolution: An Overview

5.4 Summary

5.5 Keywords

5.6 Review Questions

Objectives

After studying this unit, you will be able to:

- Discuss Office Applications
- Discuss Internet Applications
- Explain the concept of personal information management

Introduction

In this unit, we will discuss various applications of linux which are used in everyday life. We will discuss office applications which include Open Office, KDE KOffice, Gnome Office, etc. We will also discuss various Internet applications. The Internet has become an important part of daily life for many people. Linux has all of the important Internet applications that you could need. The Linux system is a great platform for offering networking services. We will give an overview of most common Internet applications. Also we will discuss the concept of personal information management.

Notes

5.1 Office Applications

Some of the office basics include Word processor, spreadsheet, presentation software, calendar, calculator, etc. Most of the Linux distributions come with the OpenOffice.org (often abbreviated as OO.o or Ooo) suite of office applications and tools. You can try all of them one by one and see which one takes your fancy. Each application is fairly intuitive to use. Although some nuances of the user interface may be new to you, you'll become comfortable with the interface after using it a few times. Now we will briefly discuss the following applications, in addition to some other commercially available office applications for Linux:

5.1.1 Open Office

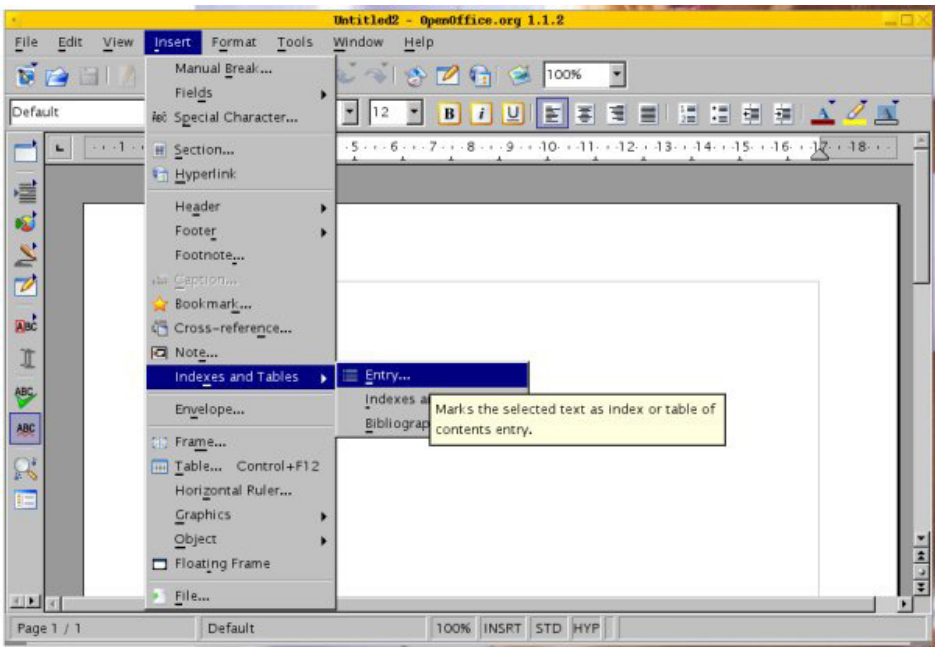
The OpenOffice.org project (www.openoffice.org) has developed an office suite known as OpenOffice.org. OpenOffice.org is similar to major office suites such as Microsoft Office. Its main components are the Writer word processor, the Calc spreadsheet, and the Impress presentation program.

You can easily start OpenOffice.org — either the overall suite or an individual application — from most GUI desktops by clicking a panel icon or by making a choice from the main menu.

Example: In SUSE, you can click a desktop icon to open the initial window of the OpenOffice.org suite.

We can create new Open Office documents or open existing documents from the main window of the OpenOffice.org.

Figure 5.1: OpenOffice.org 1.1.2 Ximian KDE Edition



Source: http://linuxreviews.org/software/office_suits/

5.1.2. KDE KOffice

KDE Office is a very complete Office suite that is, beginning with 1.3.x, getting close to being an advanced and usable Office program. It just isn't there yet.

Notes

KWord

KWord has a basic Word Processor that allows us to write letters and other simple texts. However, some more advanced features are still missing, and it has some irritating bugs. If you are using the KDE desktop environment then you can make a Window Screenshot with Alt+Print and a Desktop Screenshot with Alt+Print, and immediately insert it into Kword (or other KDE programs) by pasting (ctrl-v or shift-insert).

KSpread

KSpread is considered as a decent spreadsheet, and will probably turn out to be a very nice thing someday. Today, anyone used to Microsoft Excel or other modern spreadsheets will simply find that too many significant features are missing.

Task Compare and contrast Kword and Kspread.

5.1.3 The Gnome Office

GNOME Office consists of various applications including AbiWord, Gnumeric, Evince, and Evolution. Table 5.1 shows a current listing for common GNOME Office applications. All implement the support for embedding components, ensuring drag-and-drop capability throughout the GNOME interface. You can install these applications as a group with the meta-package at Ubuntu Software Center, Office | Gnome Office. This meta-package will also install the Evolution mailer, Inkscape graphics editor, and the Gimp image editor.

Table 5.1: GNOME Office Applications

Application	Description
AbiWord	Cross-platform word processor
Gnumeric	Spreadsheet
Evince	Document Viewer
Evolution	Integrated e-mail, calendar, and personal organizer
Dia	Diagram and flow chart editor
GnuCash	Personal finance manager
Glom	Database front end for PostgreSQL database
glabels	Label Designer
Planner	Project planner

AbiWord is considered as an open source word processor whose objective is to be a complete cross-platform solution, running on Unix, Mac and Windows, and Linux.

Gnumeric is considered as a professional-level GNOME spreadsheet which is designed to substitute commercial spreadsheets. Gnumeric supports standard GUI spreadsheet features, including auto filling and cell formatting, and an extensive number of formats. Gnumeric also supports plug-ins, making it possible to extend and customize its abilities easily.

Dia is a drawing program which is designed to create diagrams, like database, flow chart, circuit object, and network diagrams. You can create elements along with lines and arcs with different

types of endpoints such as arrows or diamonds. Data can be saved in XML format, making it transportable to other applications.

GnuCash is a personal finance application which manages accounts, stocks, and expenses.

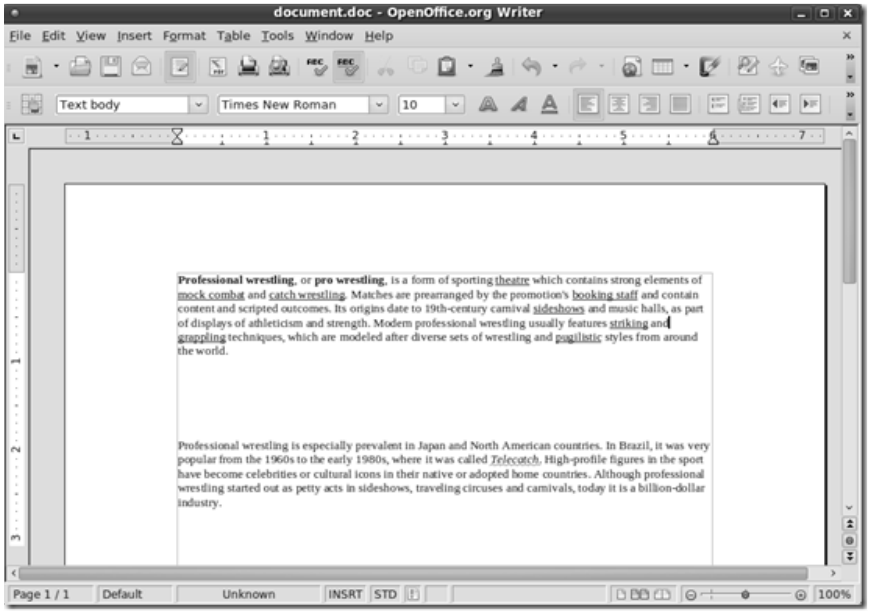
5.1.4 Writing with OpenOffice.org Writer

OpenOffice.org Writer is a word-processing program which is a part of an open-source office application suite, OpenOffice.org project. To start OpenOffice.org Writer, choose Main Menu>Office>OpenOffice.org Writer from GNOME or KDE (you can also start OpenOffice.org Writer by clicking the panel button with icon showing papers next to a pen). OpenOffice.org Writer prompts you to register and displays a blank document in its main window. Using Writer is simple—it's similar to other word processors such as Microsoft Word.

Example: You can type text into the blank document, format text, and save text when done.

We can also open the documents prepared with MS Word on a Windows machine. Figure 5.2 shows a MS Word document being opened in OpenOffice.org Writer.

Figure 5.2: Preparing Documents in OpenOffice.org Writer



Source: http://linuxapplications.blogspot.in/2012/08/introducing-linux-applications.html

On saving a document, Writer saves it by default in OpenOffice.org 1.0 Text Document format. The extension of its file is .sxw.

If you want to share OpenOffice.org Writer documents with MS Word, save the documents in various formats including Microsoft Word 97/2000/XP, Microsoft Word 95, Microsoft Word 6.0, and Rich Text Format (.rtf). Microsoft Word can open .rtf files.

5.1.5 Preparing Spreadsheets with OpenOffice.org Calc

One of the spreadsheet program in the OpenOffice.org application suite is Calc. To start Calc, select Spreadsheet from the Office group in the main menu. Alternatively, select File → New → Spreadsheet from any OpenOffice.org window. Calc displays its main window, which looks similar to Windows-based spreadsheets, like MS Excel.

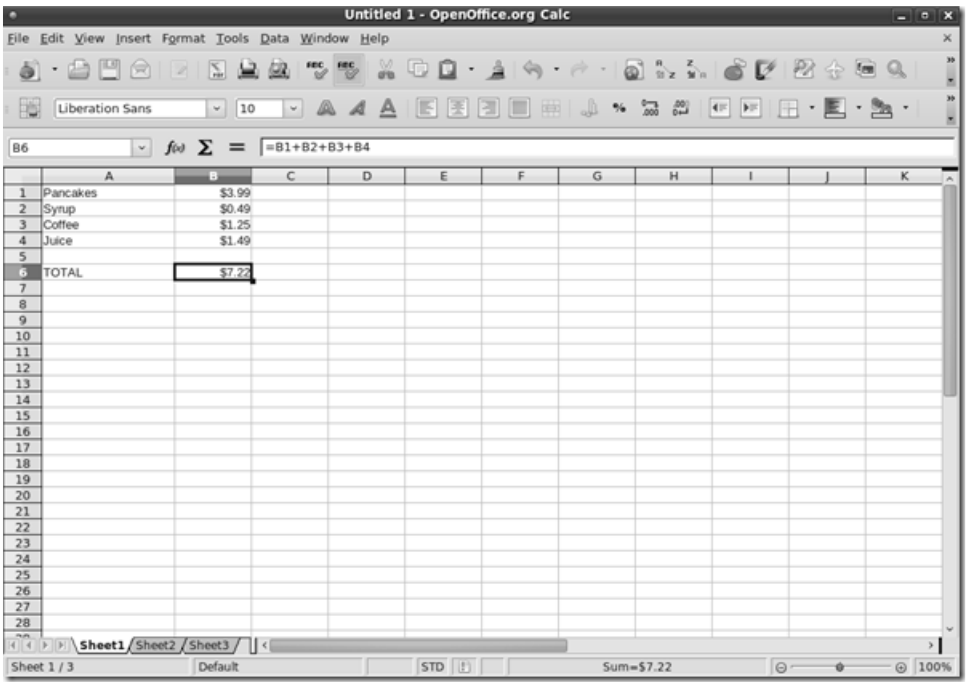
Notes

We use Calc in the same way we use Microsoft Excel. We can type entries in cells, use formulas, and format the cells.

For example, you can specify the type of value and the number of digits after the decimal point.

Figure 5.3 shows a typical spreadsheet in Calc.

Figure 5.3: Preparing Spreadsheets Using OpenOffice.org Calc



Source: http://linuxapplications.blogspot.in/2012/08/introducing-linux-applications.html

When preparing the spreadsheet, formulas are used in the same format that you would in Microsoft Excel.

Example: Use the formula SUM(D2:D6) to add the entries from cell D2 to D6. To set cell D2 as the product of the entries A2 and C2, type =A2*C2 in cell D2.

If you need to find out more regarding the functions available in OpenOffice.org Calc, select Help➪OpenOffice.org Help from the menu. Alternatively, press F1 in newer versions. This opens the OpenOffice.org Help window, from which you can browse the functions by category and click a function to read more about it.

In order to save the spreadsheet, select File→Save As. This will provide a dialog box, from which the file format, the directory location, and the name of the file can be specified. When you save a document, Calc saves it in OpenDocument spreadsheet format in a file with the .ods extension by default.

OpenOffice.org Calc can save the file in several other formats, including Microsoft Excel 5.0/95/97/2000/XP, as well as text file with comma-separated values (CSV).

In order to exchange files with MS Excel, save the spreadsheet in Microsoft Excel format. Then a file can be transferred to a Windows system and we can open it in MS Excel.

Notes

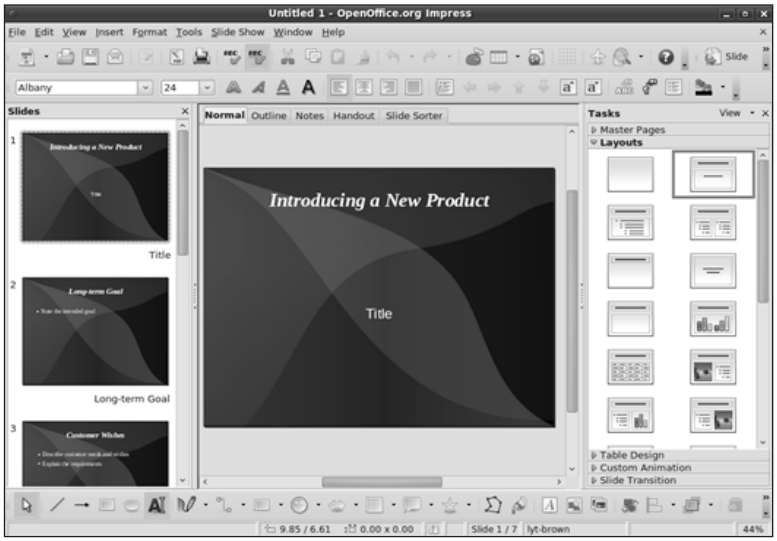
5.1.6 Presenting Information with OpenOffice.org Impress

OpenOffice.org Impress is considered as part of the OpenOffice.org office application suite. Briefing packages (i.e., slide presentations) can be prepared using Impress. It is similar to MS PowerPoint. To run Impress, choose Main Menu→Office→OpenOffice.org Impress from the GNOME or KDE desktop.

To start working, choose the required document type and template. A style for the presentation package is given by the template. Also, you can choose to open an existing document.

Figure 5.4 shows a typical slide presentation in Impress.

Figure 5.4: Preparing Briefing Packages in OpenOffice.org Impress



Source: http://linuxapplications.blogspot.in/2012/08/introducing-linux-applications.html

The Impress window shows the first slide. The exact appearance relies on the document type and template you choose. You can begin adding text and other graphic objects such as images, text, and lines to the slide.

Choose Insert Slide from the floating menu to insert a new table. A gallery of slide layouts appears in a dialog box. Click the style of slide you want in the dialog box. You can then add text and graphics to that new slide.

In order to save a presentation, choose File→Save from the menu. For new documents, you have to provide a filename and select the directory in which to save the file.

To share the slides with somebody who uses MS PowerPoint, save the presentation in MS PowerPoint 97/2000/XP format.

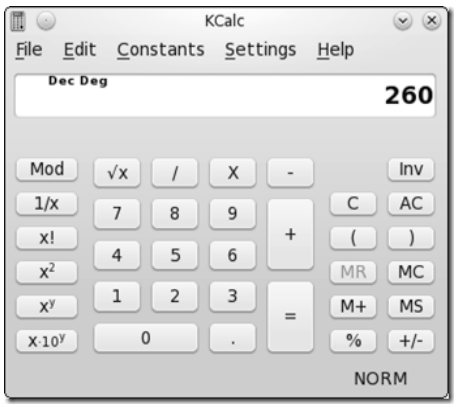
5.1.7 Using Calculators

There is a choice of the KDE calculator or the GNOME calculator. Both are considered as scientific calculators, and you can do the typical scientific calculations like square root and inverse, as well as trigonometric functions, such as sine, cosine, and tangent.

To run the GNOME calculator, chooseMain MenuAccessories→Calculator in the GNOME panel. Figure 5.5 shows the GNOME calculator.

Notes

Figure 5.5: The GNOME Calculator



Source: http://linuxapplications.blogspot.in/2012/08/introducing-linux-applications.html

The KDE calculator has more features than the GNOME calculator.

Example: It can perform calculations in hexadecimal, decimal, octal, and binary format.

You can begin the KDE calculator from the KDE desktop by choosing Main Menu→ Accessories→ Scientific Calculator. The KDE calculator is intuitive to use.

Self Assessment

Fill in the blanks:

1. ..... has a basic Word Processor that will allow you to write letters and other simple texts.
2. ..... is an open source word processor that aims to be a complete cross-platform solution, running on Mac, Unix, and Windows, as well as Linux.
3. ..... is a professional-level GNOME spreadsheet meant to replace commercial spreadsheets.
4. To run the GNOME calculator, select Main Menu>..... >Calculator in the GNOME panel.
5. ..... is a drawing program designed to create diagrams.

5.2 Internet Applications

Linux has several tools related to chatting, web browsing, or even web development.

5.2.1 Mozilla Web Browser

Today, the most popular web browser on the Internet is Mozilla Firefox. Firefox is even extremely customisable. We can use themes, plugins, and personas to create an enjoyable workspace. Thousands of plugins are available to choose from, each offering their own unique features to Firefox. Mozilla Firefox has made many improvements over the years and it is constantly evolving. Many new features have been added in the newer versions. Currently you can find Mozilla Firefox installed by default on many Linux distributions.

This web browser consists of the basic bookmark functions, customisable toolbars, typical menu setup, and numerous tabs for browsing sessions.

Notes

Figure 5.6: Mozilla Web Browser

Source: http://linux.about.com/od/linux101/a/desktop08a.htm

Firefox Installation

If you are not having Firefox on your system, use the following command to install it.

sudo apt-get install firefox

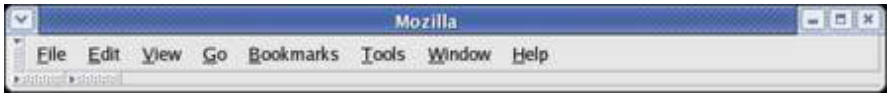
Mozilla Firefox Web Browser Interface

It is necessary for users to learn everything about the Mozilla Firefox interface and precisely what it has to offer.

Did u know? The Firefox interface can be customized completely to suite your needs.

- ***The Menu Bar:*** The menu bar can be found at the top of the Mozilla Firefox web browser window. Various menus are available where you find all of the available actions and options for the Firefox web browser.

Figure 5.7: Menu Bar



Source: http://linux.about.com/od/linux101/a/desktop08a.htm

- ***Navigation Bar:*** This toolbar is located below the main menu bar. It contains navigation buttons, and a navigation address bar. A quick search bar is also available for some of the best search engines.

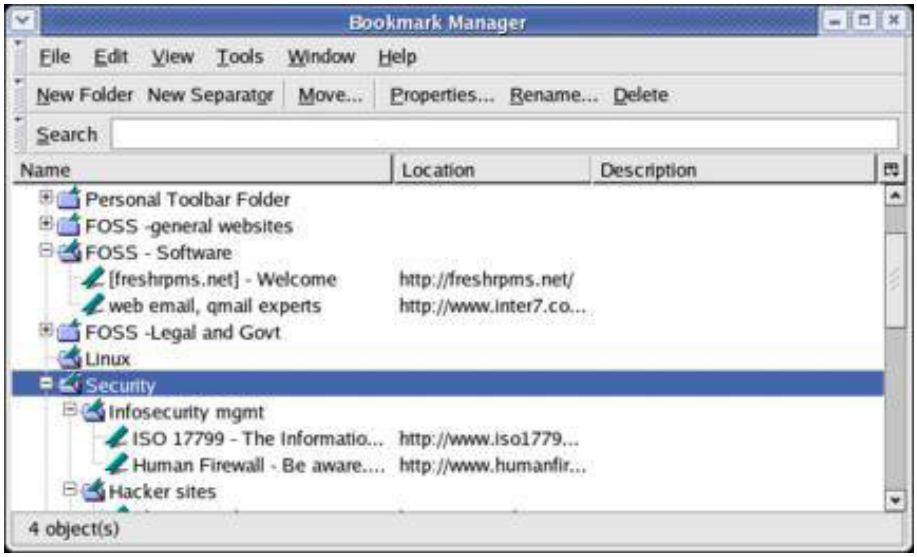
Figure 5.8: Navigation Bar

Source: http://linux.about.com/od/linux101/a/desktop08a.htm

Notes

- ***Bookmarks Toolbar:*** Bookmarks Toolbar is considered an optional toolbar which can be used to display bookmarks and bookmark folders.

Figure 5.9: Bookmarks Toolbar



Source: http://linux.about.com/od/linux101/a/desktop08a.htm

- ***Plugins Toolbar:*** Some plugins also provide additional toolbars which can also be toggled on or off from the view menu.
- ***Page Display Area:*** All your web pages are displayed in page display area. Every web page will receive its own tab that you can switch through using the tabs at the top of the display area.
- ***Status Bar:*** It is located at the bottom of the Mozilla Firefox web browser window. The status bar will simply display useful information while you are browsing.

Task Differentiate between navigation bar and bookmarks bar.

Mozilla Firefox Web Browser Add-ons

Mozilla Firefox consists of various add-ons to choose from. You can find add-ons that will help you integrate your social networks and news feeds directly into Firefox. Use add-ons to extend functions of your bookmarks and tabs. You can even find add-ons that will let your download videos straight from YouTube and many other sites.

The add-ons can be found inside tools menu. After opening your add-ons window you can easily install, remove or search for add-ons. Even view user reviews and program descriptions to help you choose your add-ons.

Caution After installing an add-on you must restart Firefox to enable it.

Notes

Mozilla Firefox Web Browser Bookmarks

Mozilla Firefox bookmark feature is used to save your favorite websites so that you can easily view them later. Firefox even offers a convenient interface for managing your saved bookmarks, or adding new ones. So be sure to take full advantage of this great feature, start bookmarking all of your favorite websites today.

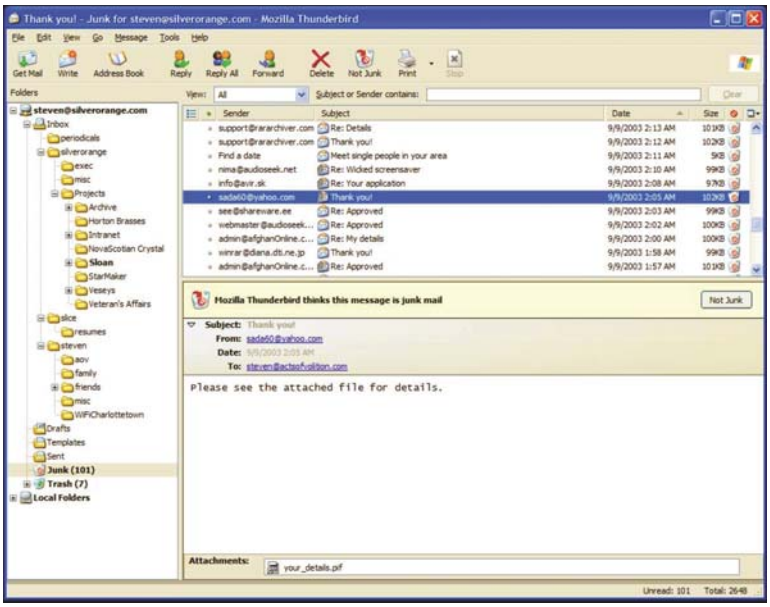
To go to the Mozilla Firefox bookmark library, select the show all bookmarks option from inside the bookmarks menu.

5.2.2 Mail Client

It is a Mozilla program which offers a simple interface in order to use e-mail in Red Hat Linux. You can access the Mozilla Mail window from your Mozilla browser window by choosing Tasks → Mail & Newsgroups.

Example: Figure 5.10 shows an example of the Mozilla Mail window.

Figure 5.10: Manage Your e-mail from the Mozilla Mail Window



Source: http://flylib.com/books/en/4.100.1.83/1/

As with Evolution e-mail, you are required to give some information regarding your mail account before you can connect to your mail server and use Mozilla Mail.

Recently, a new Junk Mail feature was added to Mozilla Mail. With this feature, any message which is believed to be junk mail is automatically tagged.

Notes By means of the Junk Toolbar, you train the Junk Mail feature by telling it when a message is or isn't junk mail. After you have identified which messages are junk mail, you can automatically move incoming junk mail to the Junk folder.

Notes

5.2.3 Instant Messaging with Gaim

By means of the Junk Toolbar, you train the Junk Mail feature by telling it when a message is or isn't junk mail. After you have identified which messages are junk mail, you can automatically move incoming junk mail to the Junk folder.

Gaim is an Instant Messaging client program that runs in Linux.

To start Gaim from a Red Hat desktop menu, select Internet Instant Messenger.

Example: Figure 5.11 shows an example of the Gaim window:

Figure 5.11: Access Your AOL Instant Messaging Using Gaim

The main Gaim window shows the following options:

- Accounts — Add screen names, passwords, and options associated with your AOL Instant Messaging accounts.
- Sign on — Sign on to AOL Instant Messaging by using your selected screen name.
- Preferences — Set preferences associated with AOL Instant Messaging and Chat windows. You can also load plug-ins that can perform special functions, such as spell checking and auto-reconnection.

Notes Gaim supports various messaging protocols such as AOL, Yahoo!, the Microsoft Network, ICQ, and IRC.

Self Assessment

Fill in the blanks:

6. The ..... bar contains navigation buttons, and a navigation address bar.

Notes

7. The ..... bar will simply display useful information while you are browsing.
8. You can use the Mozilla Firefox ..... feature to save all of your favorite websites for easy viewing at a later date.
9. The Mozilla ..... program provides a simple interface for using e-mail in Red Hat Linux.
10. Gaim is an ..... client program that runs in Linux.

5.3 Personal Information Management

Typically, Personal Information Managers (PIMs) are considered as email clients deluxe, however they generally also include at least an address book, task manager, calendar, and some type of system for taking notes. Earlier, all of these features would be managed by separate applications. Evolution is to Linux what Outlook is to Windows: A full-featured Personal and Workgroup Information Manager. With this outstanding solution you can manage your calendar, contacts, email, and tasks. The process of managing this kind of information is called personal information management (PIM).

5.3.1 Evolution

The tasks of storing, organizing, and retrieving your personal information becomes easy by the program known as evolution. This allows you to work and communicate more efficiently with others. It is a highly evolved groupware program, an integral part of the Internet-connected desktop. Evolution can help you work in a group by handling e-mail, address, and other contact information, and one or more calendars. It can do that on one or several computers, connected directly or over a network, for one person or for large groups.

Most of the common daily tasks can be accomplished with evolution quickly.

Example: It takes only one or two clicks to enter appointment or contact information sent to you by e-mail, or to send e-mail to a contact or appointment. People who get lots of e-mail will appreciate advanced features like search folders, which let you save searches as though they were ordinary e-mail folders.

5.3.2 Starting Evolution for the First Time

Start the Evolution client.

GNOME:	Click Applications > Office > Evolution.
KDE:	Click the K menu > Office > More Programs > Evolution.
Command Line:	Enter evolution.

Using the First-Run Assistant

When we run Evolution first time, it creates a directory evolution in our home directory. All local data is stored in home directory. Then, it opens a First-Run Assistant to help you set up e-mail accounts and import data from other applications.

Using the first-run assistant takes two to five minutes. Afterward, if you want to change this account, or if you want to create a new one, click Edit → Preferences, then click Mail Accounts. Choose the account you want to change, then click Edit. Or, add a new account by clicking Add.

Notes

Caution The First-Run Assistant helps you provide the information Evolution needs to get started.

The steps for using the first-run assistant are discussed below:

1. ***Defining Your Identity:*** The Identity window is the first step in the assistant.

Figure 5.12: Identity Window

Source: http://www.novell.com/documentation/evolution/evolution/?page=/documentation/evolution/evolution data /usage-mail-getnsend-send.html

In identity window, we enter some basic personal information. You can define multiple identities later by clicking Edit → Preferences, then clicking Mail Accounts.

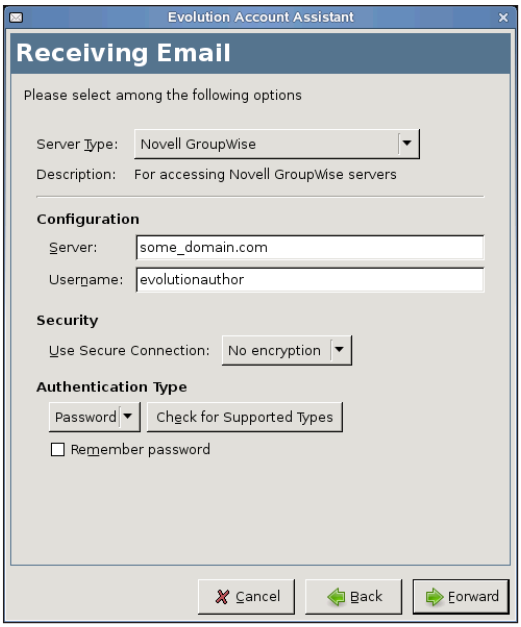
When the First-Run Assistant starts, the Welcome page is displayed. Click Forward to proceed to the Identity window.

- Type your full name in the Full Name field.
- Type your e-mail address in the E-Mail Address field.
- (Optional) Select if this account is your default account.
- (Optional) Type a reply to address in the Reply-To field.
- Use this field if you want replies to messages sent to a different address.
- (Optional) Type your organization name in the Organization field.
- This is the company where you work, or the organization you represent when you send e-mail.
- Click Forward.

2. ***Receiving Mail:*** This option allows you to determine where you get your e-mail.

Notes

Figure 5.13: Receiving Mail

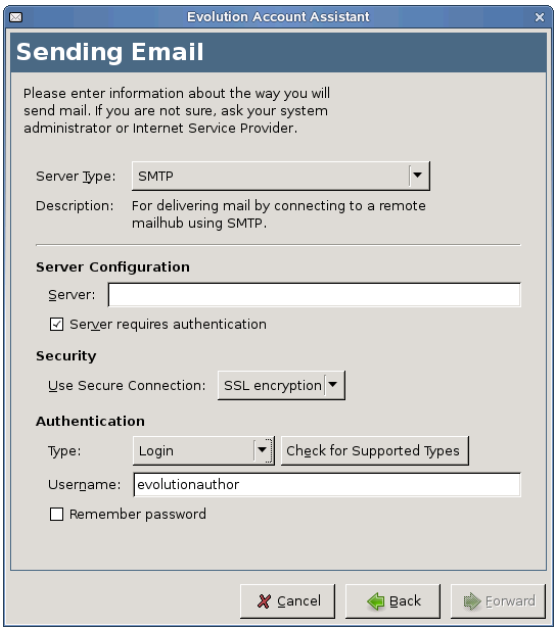


It is required to specify the type of server you want to receive mail with. If you are not sure about the type of server to select, ask your system administrator or ISP.

3. ***Receiving Mail Options:*** After selecting a mail delivery mechanism, you can set some preferences for its behavior.

4. ***Sending Mail:*** After entering the information regarding how you plan to get mail, Evolution needs to know about how you want to send it.

Figure 5.14: Sending email



Notes

- Select a server type from the Server Type list.

The following server types are available:

- ***Sendmail:*** It uses the Sendmail program to send mail from your system. Sendmail is more flexible, but is not as easy to configure, so you should select this option only if you know how to set up a Sendmail service.
- ***SMTP:*** It sends mail by means of an outbound mail server. This is the most common choice for sending mail. If you choose SMTP, there are additional configuration options.
- Type the Server address in the Server field.
- If you are unsure what your Server address is, contact your system administrator.
- Select if your server requires authentication.
- If you selected that your server requires authentication, you need to provide the following information:
  - Select your authentication type in the Authentication list.
  - Or to have Evolution check for supported types, click Check for Supported Types. Some servers do not state the authentication mechanisms they support, so clicking this button is not an assurance that available mechanisms actually work.
  - Type your username in the Username field.

    Select if you want Evolution to remember your password.

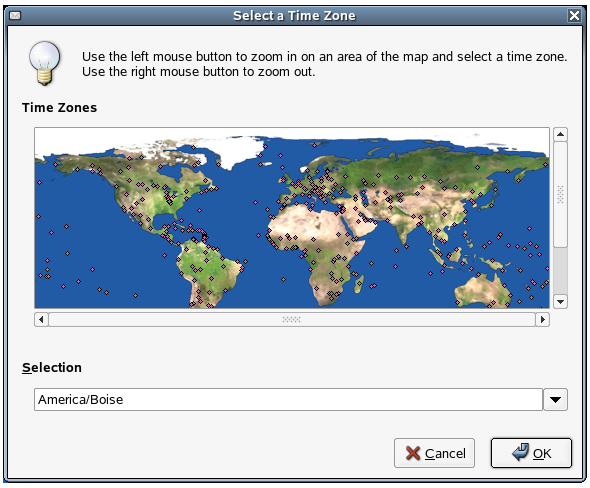
    Select if you use a secure connection (SSL).
  - Click Forward.

5. ***Account Management:*** Now you have completed the e-mail configuration process. Now you are required to provide a name to your account. The name can be any name you prefer. Type your account name on the Name field, then click Forward.
6. ***Time Zone:*** Select your time zone on the map.

   or

   Select from the time zone drop-down list.

Figure 5.15: Time Zone



Notes

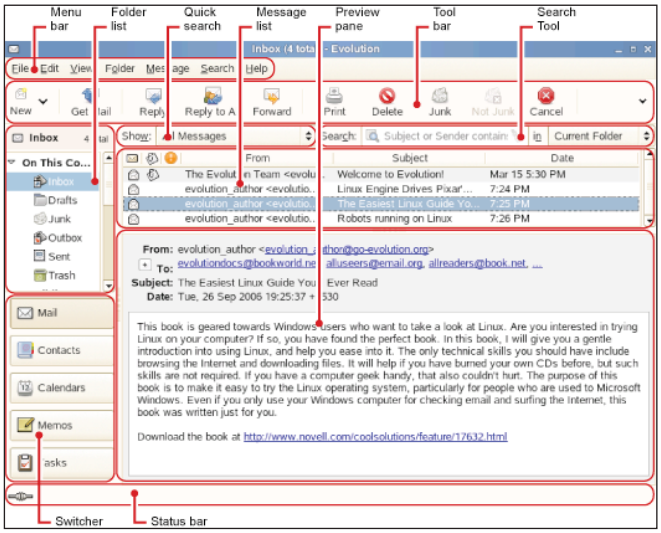
- Click OK, then click Apply.

Evolution opens with your new account created.

5.3.3 Using Evolution: An Overview

Now you are ready to start using Evolution. The main Evolution window is shown in Figure 5.16.

Figure 5.16: Main Evolution Window



- ***Menu Bar:*** The menu bar gives you access to most Evolution features.
- ***Folder List:*** The folder list gives you a list of the available folders for each account. To see the contents of a folder, click the folder name and the contents are displayed in the e-mail list.
- ***Quick Search:*** It displays all the messages that match the in-built criteria's you selected from the drop-down list at the top left side, just above the message pane.
- ***Toolbar:*** The toolbar gives you fast and easy access to the frequently used features in each component.
- ***Search Tool:*** The search tool lets you search your e-mail, contacts, calendar, and tasks to easily find what you're looking for.
- ***Message List:*** The message list displays a list of e-mail that you have received. To view an e-mail in the preview pane, click the e-mail in the e-mail list.
- ***Side bar:*** The side bar lets you switch between folders and between Evolution tools. At the bottom of the side bar is the switcher that let you switch Evolution tools, and above that is a list of all the available folders for the current tool.
- ***Switcher:*** The switcher at the bottom of the side bar lets you switch between the Evolution tools - mail, contacts, Calendars, Memos and Tasks.
- ***Status bar:*** The status bar periodically displays a message, or tells you the progress of a task. This most often happens when you're checking or sending e-mail. These progress queues are shown in the previous figure.
- ***Preview Pane:*** The preview pane displays the contents of the e-mail that is selected in the e-mail list.

Notes

The Menu Bar

Contents of menu bar always provide every possible action for any given view of your data. If you are looking at your Inbox, most of the menu items relate to e-mail.

Did u know? Some content relates to other components of Evolution and some, particularly in the File menu, relates to the application as a whole.

- ***File:*** Anything related to a file or to the operations of the application is listed in this menu, such as creating things, saving them to disk, printing them, and quitting the program itself.
- ***Edit:*** Holds useful tools that help you edit text and move it around. Lets you access the settings and configuration options in the Edit menu.
- ***View:*** Lets you decide how Evolution should look. Some of the features control the appearance of Evolution as a whole, and others the way a particular kind of information appears.
- ***Folder:*** Holds actions that can be performed on folders. You can find things like copy, rename, delete, and so on.
- ***Message:*** Holds actions that can be applied to a message. If there is only one target for the action, such as replying to a message, you can normally find it in the Message menu.
- ***Search:*** Lets you search for messages, or for phrases within a message. You can also see previous searches you have made. In addition to the Search menu, there is a text entry box in the toolbar that you can use to search for messages. You can also create a search folder from a search.
- ***Help:*** Opens the Evolution Help files.

E-Mail

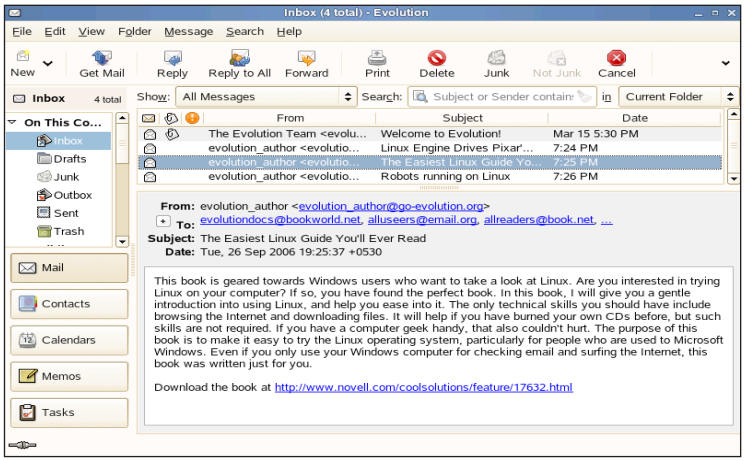
Evolution e-mail is similar to other e-mail programs in many ways:

- It can send and receive e-mail in HTML or as plain text, and makes it easy to send and receive multiple file attachments.
- It supports multiple e-mail sources, including IMAP, POP3, and local mbox or mh spools and files created by other e-mail programs.
- It can sort and organize your e-mail in a wide variety of ways with folders, searches, and filters.
- It lets you guard your privacy with encryption.

Nevertheless, Evolution has some significant differences as compared to other e-mail programs. First, it is built to manage very large amounts of e-mail. The junk e-mail, message filtering and searching functions were built for speed and efficiency. There's also the search folder, an advanced organizational feature not found in some e-mail clients. If you get a lot of e-mail, or if you keep every message you get in case you need to refer to it later, you'll find this feature especially useful. Here's a quick explanation of what's happening in your main Evolution e-mail window.

Notes

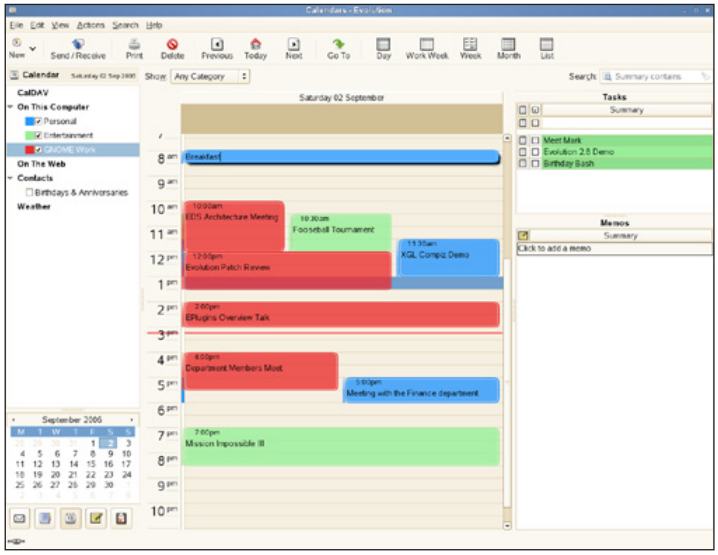
Figure 5.17: Evolution e-mail Window



The Calendar

In order to start using the calendar, click Calendar in the switcher. By default, the calendar shows today's schedule on a ruled background. At the left bottom, there is a monthly calendar you can use to switch days. At the upper right, there is a Task list, where you can keep a list of tasks separate from your calendar appointments. Below that, there is a Memo list.

Figure 5.18: Calender



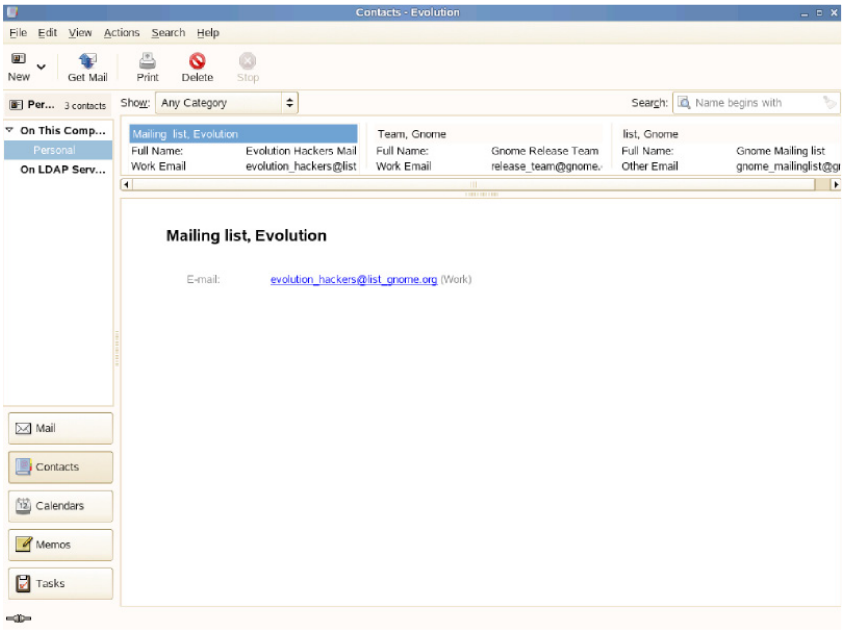
The Contacts

All the functions of phone book or address book or are handled by evolution contacts. However, it is easier to update Evolution than it is to change an actual paper book, in part because Evolution can synchronize with Palm OS(thrdmrk) devices and use LDAP directories on a network.

Notes

To use the contacts tool, click Contacts in the switcher. By default, the display shows all your contacts in alphabetical order, in a minicard view. You can select other views from the View menu, and adjust the width of the columns by clicking and dragging the gray column dividers.

Figure 5.19: Contacts



Self Assessment

Fill in the blanks:

11. ..... can help you work in a group by handling e-mail, address, and other contact information, and one or more calendars.
12. The ..... option lets you determine where you get your e-mail.
13. ..... server uses the Sendmail program to send mail from your system.
14. ..... sends mail using an outbound mail server.
15. The Evolution ..... can handle all of the functions of an address book or phone book.

5.4 Summary

- OpenOffice.org is an office suite developed by the OpenOffice.org project (www.openoffice.org). OpenOffice.org is similar to major office suites such as Microsoft Office.
- KDE Office is a very complete Office suite that is, beginning with 1.3.x, getting close to being an advanced and usable Office program. It just isn't there yet.
- There are several GNOME Office applications available including AbiWord, Gnumeric, Evince, and Evolution.
- Mozilla Firefox has the basic bookmark functions, typical menu setup, customisable toolbars, and multiple tabs for browsing sessions.

Notes

- To expand the functions of Mozilla Firefox there are lots of add-ons to choose from.
- The Mozilla Mail client program provides a simple interface for using e-mail in Red Hat Linux.
- Gaim is an Instant Messaging client program that runs in Linux.
- Personal Information Management typically include email clients deluxe, but they usually also include at least an address book, calendar, task manager, and some sort of system for taking notes.

5.5 Keywords

AbiWord: AbiWord is an open source word processor that aims to be a complete cross-platform solution, running on Mac, Unix, and Windows, as well as Linux.

Gaim: Gaim is an Instant Messaging client program that runs in Linux.

GnuCash: GnuCash is a personal finance application for managing accounts, stocks, and expenses.

Gnumeric: Gnumeric is a professional-level GNOME spreadsheet meant to replace commercial spreadsheets.

Mozilla Firefox: Mozilla Firefox is the most popular web browser on the Internet which is highly customisable.

Mozilla Mail client: The Mozilla Mail client program provides a simple interface for using e-mail in Red Hat Linux.

OpenOffice.org: OpenOffice.org is an office suite developed by the OpenOffice.org project (www.openoffice.org).

PIM: Personal Information Management typically include email clients deluxe, but they usually also include at least an address book, calendar, task manager, and some sort of system for taking notes.

5.6 Review Questions

1. What are the office applications? Explain.
2. Explain the use of opencalc for spreadsheets.
3. Explain the concept of Internet applications.
4. Make distinction between Mozilla Web Browser and Mozilla Mail Client.
5. Describe the GAIM instant messenger application.
6. Describe the various GNOME Office applications.
7. Explain how to present information with OpenOffice.org Impress.
8. Explain the use of Mozilla Firefox Web Browser Bookmarks.
9. Discuss the concept of personal information management.
10. Discuss the steps included in Starting Evolution for the First Time.

Notes

Answers: Self Assessment

1. KWord
2. AbiWord
3. Gnumeric
4. Accessories
5. Dia
6. Navigation
7. Status
8. Bookmark
9. Mail client
10. Instant Messaging
11. Evolution
12. Receiving E-mail
13. Sendmail
14. SMTP
15. Contacts

5.7 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley

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

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

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

Online links

http://www.techrepublic.com/blog/10things/10-must-have-linux-office-applications/423

http://linuxlibrary.org/linux-desktop-internet-applications/

http://linuxreviews.org/software/office_suits/

http://arstechnica.com/information-technology/2008/03/creating-rich-internet-applications-on-linux-with-webkit/

Notes

Unit 6: Everyday Applications-II

CONTENTS

Objectives

Introduction

6.1 Multimedia Applications

6.1.1 Using a Digital Camera

6.1.2 Playing Audio CDs

6.1.3 Playing Sound Files

6.1.4 Burning a CD

6.1.5 Using Graphics and Imaging Apps

6.2 System Applications

6.2.1 A Simple Text Editor

6.2.2 File Management Application

6.2.3 File Compression and Archiving

6.2.4 Emulators

6.3 Summary

6.4 Keywords

6.5 Review Questions

6.6 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss Multimedia Applications
- Explain various Graphics and Imaging Applications
- Explain System Applications

Introduction

In this unit, we will discuss various multimedia applications such as using digital cameras, playing audio CDs and burning CD-ROMs. We will also discuss the concept of using Graphics and Imaging Apps. System applications of Linux include a simple text editor, file management application, file compression and archiving application, and emulators such as VMWare and Wine. Here we will discuss the application of File Compression and Archiving by using File Roller. We will also explain the process of decompressing and unarchiving with File Roller.

6.1 Multimedia Applications

Most of the Linux distributions consists of some multimedia applications. These applications include multimedia audio players and CD players. It also include applications for using digital cameras and burning CD-ROMs. To play some other multimedia files (such as MPEG video),

Notes

you may have to download and install additional software on your Linux system. A few typical multimedia tasks and the applications that can be used to perform these tasks are discussed below:

- ***Using digital cameras:*** A digital camera tool is used to download photos from your digital camera in Linux.
- ***Playing audio CDs:*** Use one of many audio CD players that come with Linux.
- ***Playing sound files:*** Use Rhythmbox or XMMS multimedia audio players. You can also download other players from the Internet.
- ***Burning a CD:*** Use a CD burner, like K3b, to burn audio and data CDs.

6.1.1 Using a Digital Camera

Most of the Linux distributions include a digital camera application that can be used to download pictures from digital cameras.

Example: SUSE and Xandros come with digiKam, which works with many makes and models of digital cameras.

The cameras can connect to the Universal Serial Bus (USB) port or the serial port. It actually depends on the model.

Follow the steps given below to use digiKam with your digital camera:

1. Connect your digital camera to the serial port or USB port (whichever interface the camera supports) and turn on the camera.
2. Start digiKam. digiKam can be seen in the main menu under the Graphics or Images submenu. The first time you open digiKam, you are required to specify a default location to store images and select a number of other configuration preferences.
3. From the digiKam menu, select Settings»Configure digiKam.

   A configuration dialog box appears.
4. Click the Cameras tab in the dialog box, and then click Auto Detect.

   If your camera is supported and the camera is configured to be in Picture Transfer Protocol (PTP)mode, the camera is detected. If not, you can get the photos from your camera by using an alternate method described after these steps.
5. Select your camera model from the Camera menu.

   A new window appears and, after a short while, displays the photos in the camera.
6. Click the thumbnails to choose the images you want to download; then select Camera → Download to download the images.

   digiKam downloads the images. These files can be saved in a folder and the photos can be edited in The GIMP or your favorite photo editor.

If digiKam doesn't identify your digital camera, don't despair. You can still access the digital camera's storage media (compact flash card, for example) as a USB mass storage device, provided your camera supports USB mass storage. To access the images on your USB digital camera, use the following steps.

1. Read the camera manual and use the menu options of the camera to set the USB mode to Mass Storage.

Notes

If USB mass storage is not supported in your camera, this procedure cannot be used to access the photos. If the camera supports the Picture Transfer Protocol mode, you can use digiKam to download the pictures.

2. Connect your digital camera to the USB port by using the cable that came with the camera. Then turn on the camera.

   Linux detects the camera and opens the contents of the camera in a file manager window.

3. Click to select photos and copy them to your hard drive by dragging and dropping them into a selected folder.

4. Close the file manager window, disconnect the USB cable from the PC, and turn off the camera.

Whether we use a digital camera tool like digiKam or access our camera like any other storage device, it becomes easier to get pictures onto the computer.

6.1.2 Playing Audio CDs

All Linux distributions takes place in either KDE or GNOME CD player applications. In order to play an audio CD, we need a sound card, and that sound card must be configured to work in Linux.

In some distributions, an audio CD can be inserted into the drive. This will provide a dialog box and asks whether you want to play the CD with the CD player. If this dialog box doesn't appear, locate an audio CD player by choosing Applications➪Sound and Video from the main menu.

The KDE CD player exhibits the CD title and the current track name. An active Internet connection is required to download song information from the CD database. After the CD player downloads information about a particular CD, it caches that information in a local database for future use. The CD player user interface is intuitive, and you can figure it out easily.

Did u know? One good feature of CD player user interface is that you can select a track by title.

6.1.3 Playing Sound Files

To open and play sound files (such as MP3 files), we can use Rhythmbox or XMMS. Users having huge MP3 music libraries generally like Rhythmbox as it can assist in organizing their music files. You can start Rhythmbox by choosing the music player application from the main menu in several distributions, including Debian and Fedora. When you first start Rhythmbox, it displays an assistant that prompts you for the location of your music files so that Rhythmbox can manage your music library.

After identifying the locations of music files, Rhythmbox begins and shows the library in an systematized way. You can then select music and play it.

Another music player that can play different types of sound files is XMMS. XMMS can be started by selecting the audio player application from the main menu. After XMMS begins, you can open a sound file by selecting Window Menu»Play File or by pressing L. Then select one or more music files from the Load File dialog box. Click the Play button, and XMMS starts playing the sound file.

Some free Linux distributions does not allow you to play MP3 files. This is because the MP3 decoder is not included. Nevertheless, MP3 playing functions well in Debian, Knoppix, SUSE, and Xandros. Due to legal reasons, the versions of Rhythmbox and XMMS in Fedora don't include

Notes

the code needed to play MP3 files, so you have to translate MP3s into a supported format, such as WAV, before you can play them. You can, however, download the source code for Rhythmbox and XMMS and build the applications with MP3 support.

Notes The Ogg Vorbis format can also be used for compressed audio files as Ogg Vorbis is a patent- and royalty-free format.

6.1.4 Burning a CD

Today, most GUI file managers have the ability to burn CDs.

Example: Nautilus File Manager and Xandros File Manager have incorporated features to burn CDs.

Linux distributions also come with standalone GUI programs that enable you to easily burn CDs and DVDs.

For example, K3b is a popular CD/DVD burning application for KDE that's available in Knoppix and SUSE.

Most of the CD burning applications are easy to use. Basically, we collect the files that we want to burn to the CD or DVD and then the burning process is started. Obviously, for this to work, your PC must have a CD or DVD burner installed.

The K3b window's upper part is for browsing the file system to choose what you want to burn onto a CD or DVD. The upper-left corner displays the CD writer device installed.

To burn a CD, we begin with one of the projects displayed in K3b window's lower part.

Example: New Audio CD Project, or New Data DVD Project.

Finally, you are required to add files and burn the project to the CD or DVD. This is done by selecting Project → Burn or pressing Ctrl+B. For an audio CD, you can drag and drop MP3 files as well as audio tracks.

To burn CDs, K3b needs the external command-line programs cdrecord and cdrdao. K3b also needs the growisofs program to burn DVDs.

If an error is encountered regarding missing cdrdao , ensure that your system is connected to the Internet and then type apt-get install cdrdao to install it.

6.1.5 Using Graphics and Imaging Apps

In order to work with images and graphics, you can use graphics and imaging applications. Two most popular applications are

- ***The GIMP (GNU Image Manipulation Program):*** The GIMP is a program used for viewing and performing image manipulation tasks, like image composition, photo retouching, and image creation.
- ***Gnome Ghostview (GGv):*** It is a graphical application which is capable of displaying PostScript files.

The GIMP

Notes

The GIMP is considered as an image manipulation program which is written by Peter Mattis and Spencer Kimball. It is released under the GNU General Public License (GPL). Most Linux distributions come with this program, even though you may have to specifically choose a package to install it. The GIMP is comparable to other image manipulation programs, such as Adobe Photoshop and Corel PHOTO-PAINT.

In order to try out The GIMP, look under the Graphics category in the main menu. When you start The GIMP, a window is displayed with copyright as well as license information. Click the Continue button to proceed with the installation. The next screen shows the directories to be created when you proceed with a personal installation of The GIMP.

If The GIMP is not found under the Graphics category, select Add/Remove Software from the System Settings menu and install it from there.

The GIMP installation includes forming a directory in home directory and placing a number of files there. This directory basically holds information about any changes to user preferences you may make to The GIMP. Go ahead and click the Continue button at the bottom of the window. The GIMP creates the necessary directories, copies the necessary files to those directories, and guides you through a series of dialog boxes to complete the installation.

Click the Continue button on the completion of installation. Then, any plug-ins are loaded by The GIMP. Plug-ins are the external modules that improve its functionality. It displays a startup window that shows a message about each plug-in as it loads. After finishing the startup, The GIMP displays a tip of the day in a window. You can browse the tips and click the Close button to close the Tip window. At the same time, The GIMP displays a number of other windows, including a main navigation window titled The GIMP, a Toolbox window (usually on the left side), a Brush Selection window, and a Layers, Channels, Paths window. The center navigation window gives you access to new images to work with, image editing functions, and a number of effect filters that you can apply to the image. The Toolbox window lets you quickly select a number of important image manipulation tools.

In order to open an image file in The GIMP, select File→Open. It will provide the Load Image dialog box. You can change directories and select the image file that you want to open. The GIMP can read all common image file formats, such as GIF, JPEG, TIFF, PCX, BMP, PNG, and PostScript. After you select the file and click OK, The GIMP loads the image into a new window.

There are many buttons in the toolbox which represent the tools used by you to edit the image and apply special effects. You can get pop-up help on each tool button by placing the mouse pointer over the button. You can select a tool by clicking the tool button, and you can apply that tool's effects to the image to see what it does.

On right-clicking the image window, a pop-up menu is displayed by The GIMP. This pop-up menu has most of the options from the GIMP's top toolbar.

Caution You can do much more than just load and view images with The GIMP. You can also choose Help➪GIMP Online➪User Manual Website to access the online documentation for The GIMP. (You need an Internet connection for this command to work.)

You can do much more than just load and view images with The GIMP. You can also choose Help➪GIMP Online➪User Manual Website to access the online documentation for The GIMP. (You need an Internet connection for this command to work.)

Notes

GNOME Ghostview

A graphical application used for for viewing and printing PostScript or PDF documents is GNOME Ghostview. For a long document, you can view and print selected pages. You can also view the document at various levels of magnification by zooming in or out.

In order to run GNOME Ghostview, select Graphics → PostScript Viewer from GUI desktop. The GNOME Ghostview application window appears. In addition to the menu bar and toolbar along the top edge, a vertical divide splits the main display area of the window into two parts.

To load and view a PostScript document in GNOME Ghostview, select File→Open. Alternatively, click the Open icon on the toolbar. GNOME Ghostview displays a File-Selection dialog box. Use this dialog box to navigate the file system and select a PostScript file.

To open the chosen file, click the Open File button in the File-Selection dialog box. GNOME Ghostview opens the chosen file, processes its contents, and shows the output in its window.

Notes GNOME Ghostview is useful for viewing different types of documents that come in PostScript format. Usually, these files have the .ps extension. You can also open PDF files — which typically have .pdf extensions — in GNOME Ghostview.

Task Make distinction between GIMP and GNOME Ghostview.

Self Assessment

Fill in the blanks:

1. A ..... tool is used to download photos from your digital camera in Linux.
2. Depending on the model, the cameras can connect to the serial port or the ..... port.
3. Linux detects the camera and opens the contents of the camera in a ..... window.
4. The ..... CD player displays the title of the CD and the name of the current track.
5. To .....a CD, you start with one of the projects shown in the lower part of the K3b window.
6. The ..... is a program for viewing and performing image manipulation tasks, such as photo retouching, image composition, and image creation.
7. ..... is a graphical application capable of displaying PostScript files.
8. ..... is useful for viewing various kinds of documents that come in PostScript format.

6.2 System Applications

The various system applications provided by Linux are discussed below.

Notes

6.2.1 A Simple Text Editor

The manner in which tasks are completed mainly by means of GUI-based applications draws a big attention of many users. UNIX is related with command line input. The things became simpler only when a graphical user interface front end was offered by X-Window system. The simple text editor is one of the applications that lend itself to this technique of input. The editors like OpenWriter can be used to edit these type of documents. However, the functionality as well as formatting offered with applications such as Open Writer is actually overkill for configuration files, etc. Here gedit is of much use.

The gedit application is analogous to Wordpad or Notepad. Gedit is considered as an efficient text editor having a nice GUI. In order to launch gedit, select Main Menu →Accessories →Text editor. Alternatively, we can type gedit in the command line.

6.2.2 File Management Application

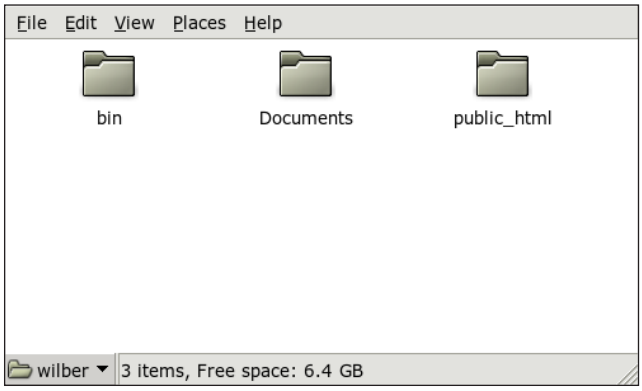
A file manager application is a major part of the desktop environment. It allows us to easily create, access, and manage all files on the system. Traditional file management in Linux would have been done via the command line, requiring some deeper knowledge of several commands to list, create, delete, or edit files and their properties. A file manager provides a graphical and more intuitive way to handle these tasks.

Nautilus is considered as the file manager and viewer of GNOME. Here we will discuss some basic functions of Nautilus and some tips on its configuration.

In Figure 6.1, we have shown the standard window of Nautilus. The default view of a folder's content is the icon view just featuring an icon and the filename for each file. If configured accordingly, a preview of the file's content can be provided. When you double-click a folder icon, a new Nautilus window opens, displaying the folder's content.

If you like browser-like file navigation, use Nautilus browser interface. Right-click the folder and select 'Browse Folder'. A new Nautilus window opens, providing the normal functionality, but with a browser's look and feel. To navigate folders and files, you can now use the 'Back', 'Forward', and 'Up' buttons as you would do in a Web browser.

Figure 6.1: Nautilus Standard Window



Source: http://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.3/suselinux-userguide_en/sec.gnomenew.nautilus.html

Nautilus supports drag and drop for file management. We can simply drag any file from the desktop to an open Nautilus window and drop to its final destination. To move files between directories, open the source directory containing the file to move. Select 'File'+'Open Location', enter the path to the target directory, then drag the files to move to the Nautilus window holding

Notes

the target directory. Files and folders can be moved to and from an open Nautilus window and the desktop.

In order to create several copies of a file, use 'Edit'+'Duplicate'. For simple cut, copy, and paste of files, use the 'Edit' menu or right-click the file icon then selecting the appropriate item from the context menu that appears. To rename a file, right-click it and select 'Rename'.

Nautilus provide support to file browsing across a network. In order to connect to a remote server, click 'File'+'Connect to Server'. Then, you are prompted for the type of server to which to connect and some additional information, like the name of the folder to access, the port number, and the username to use. When you leave this dialog with 'Connect', the remote folder is displayed as part of the 'Places'panel menu and appears as a desktop icon. For any future connections, just select the appropriate item from the 'Places' menu and provide the necessary authentication to log in to these network folders. To close these connections, right-click the desktop icon and select 'Unmount Volume'.

Nautilus includes basic CD burning functionality. If you created a directory holding data you want to back up by just burning it to a CD, click 'Places'+'CD Creator' and drag the folder holding the relevant data onto the 'CD/DVD Creator' window. Select 'File'+'Write to Disc' to copy the data to CD or DVD.

6.2.3 File Compression and Archiving

It is beneficial to store a group of files in one file for transfer to another directory, for transfer to another computer, or for easy backup. It is also useful to compress large files; compressed files take up less disk space and download faster via the Internet.

Now let us understand the difference between *archive file* and a *compressed file*. An archive file is defined as a collection of files and directories stored in one file. The archive file is not compressed — it uses the same amount of disk space as all the individual files and directories combined. A compressed file is defined as a collection of files and directories that are stored in one file *and* stored in a way that uses less disk space than all the individual files and directories combined. If disk space is a concern, compress rarely-used files, or place all such files in a single archive file and compress it.

Caution An archive file is not compressed, but a compressed file can be an archive file.

Using File Roller

File Roller is a graphical utility included in Red Hat Enterprise Linux. File Roller can compress, decompress, and archive files in common Unix and Linux formats. It has a simple interface and extensive help documentation. To start File Roller, select Archive Manager from the Applications (the main menu on the panel) → System Tools sub-menu. File Roller is also integrated into the desktop environment and Nautilus.

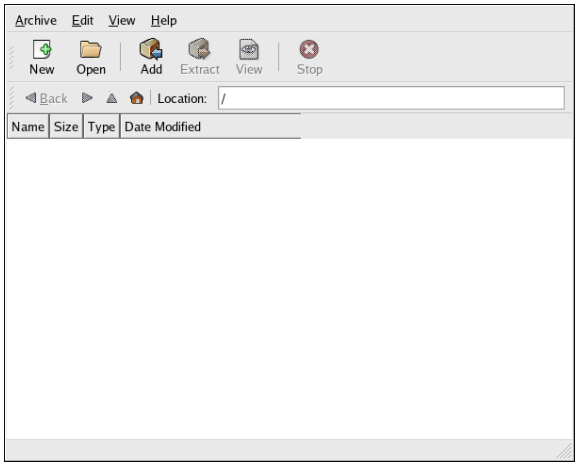
On using a file manager (for example, Nautilus), you can double-click the file you wish to unarchive or decompress to start File Roller. The File Roller browser window appears with the decompressed/unarchived file in a folder for you to extract or browse.

Decompressing and Unarchiving with File Roller

To unarchive and/or decompress a file, click the Open button on the main toolbar. A file menu pops up, allowing you to choose the archive you wish to manipulate.

Notes

Figure 6.2: File Roller in Action



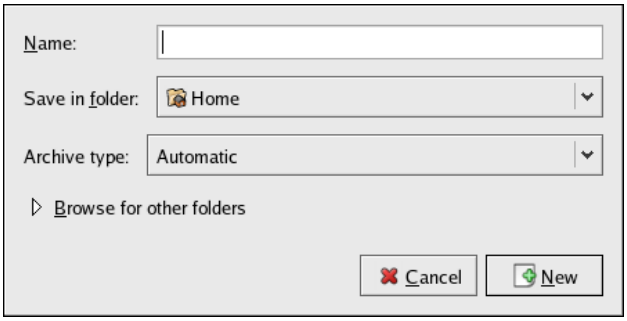
Source: http://www.centos.org/docs/4/html/rhel-sbs-en-4/s1-managing-compressing-archiving.html

Example: If you have a file called foo.tar.gz located in your home directory, highlight the file and click OK. The file appears in the main File Roller browser window as a folder, which you can navigate by double-clicking the folder icon.

All directory and subdirectory structures are preserved by File Roller. This is a convenient way if you are searching a particular file in the archive. You can extract individual files or entire archives by clicking the Extract button, choosing the directory in which to save the unarchived files, and clicking OK.

Creating Archives with File Roller

Figure 6.3: Creating an Archive with File Roller



Source: http://www.centos.org/docs/4/html/rhel-sbs-en-4/s1-managing-compressing-archiving.html

By using File Roller , we can create archives of our files and directories. To create a new archive, click New on the toolbar. A file browser pops up, allowing you to specify an archive name and the compression technique.

Example: You may choose a Tar Compressed with gzip (.tar.gz) format from the drop-down menu and type the name of the archive file you want to create. Click OK and your new archive is ready to be filled with files and directories.

Notes

To add files to your new archive, click Add, which opens a browser window that you can navigate to find the file or directory to add to the archive. Click Add when you are finished, and click Archive => Close to close the archive.

6.2.4 Emulators

There are many different emulators for numerous platforms. The emulators that simulate 8-bit home computers such as Sinclair ZX Spectrum are considered as one of the most common emulators. Emulators are obtainable for simulating complex operating systems, for example, MS Windows.

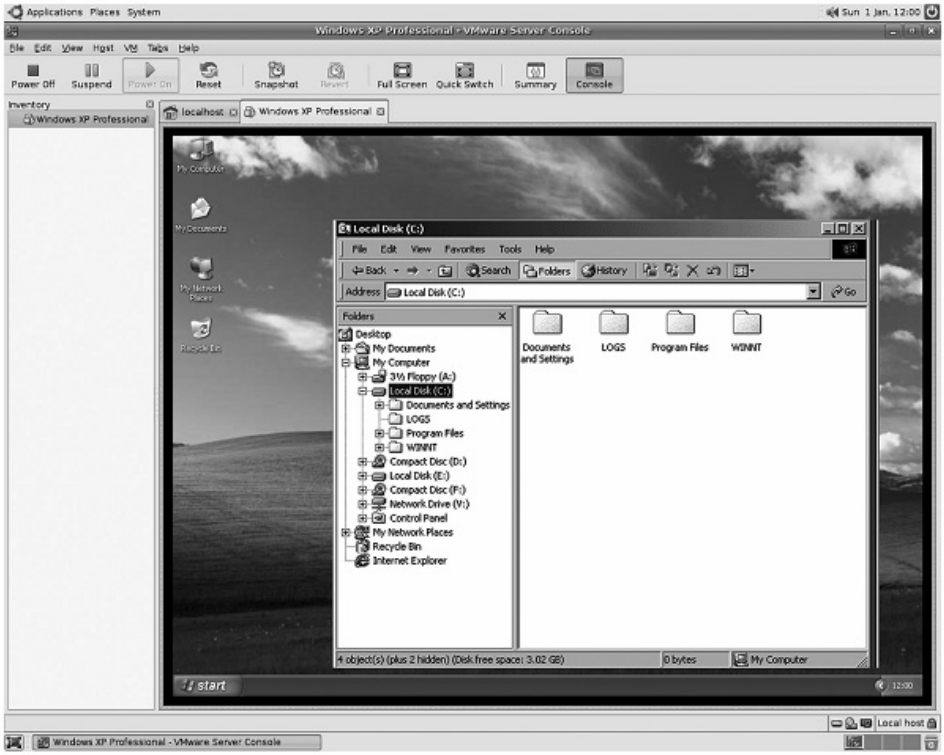
Some of the emulators are discussed below.

VMWare

This application is considered as an enterprise class virtual machine. It is capable of running numerous versions of an operating system on a single computer. For instance, numerous versions of Windows operating system can be executed on a single Linux machine.

Example: We have shown below an example which shows Windows XP running under VMWare on a Linux box. This is shown in Figure 6.4.

Figure 6.4: The VMWare Application



Wine

An application implementing the Windows API on top of native X window system is known as Wine. The native X window system is utilized by the graphical environments of KDE and GNOME.

Notes

Did u know? As compared to emulator, it is more of a compatibility layer. However, Windows applications are not allowed to run.

Task Differentiate between VMWare and Wine.

Self Assessment

Fill in the blanks:

9. A ..... application enables you to easily to create, access, and manage all files on your system.
10. ..... is the file manager and viewer of GNOME.
11. An ..... is a collection of files and directories stored in one file.
12. A ..... is a collection of files and directories that are stored in one file *and* stored in a way that uses less disk space than all the individual files and directories combined.
13. ..... can compress, decompress, and archive files in common Unix and Linux formats.
14. To create a new archive, click ..... on the toolbar.
15. ..... application is capable of running numerous versions of an operating system on a single computer.
16. An application implementing the Windows API on top of native X window system is known as .....

6.3 Summary

- Most Linux distributions come with a digital camera application that you can use to download pictures from digital cameras.
- All Linux distributions come with either the GNOME or KDE CD player applications.
- The GIMP (GNU Image Manipulation Program) is a program for viewing and performing image manipulation tasks, such as photo retouching, image composition, and image creation.
- Gnome Ghostview (GGv) is a graphical application capable of displaying PostScript files.
- A central part of your desktop environment is a file manager application, enabling you easily to create, access, and manage all files on your system. Nautilus is the file manager and viewer of GNOME.
- An archive file is a collection of files and directories stored in one file.
- A compressed file is a collection of files and directories that are stored in one file *and* stored in a way that uses less disk space than all the individual files and directories combined.
- File Roller can compress, decompress, and archive files in common Unix and Linux formats. It has a simple interface and extensive help documentation.

Notes

6.4 Keywords

Archive file: An archive file is a collection of files and directories stored in one file.

Compressed file: A compressed file is a collection of files and directories that are stored in one file *and* stored in a way that uses less disk space than all the individual files and directories combined.

File Roller: File Roller can compress, decompress, and archive files in common Unix and Linux formats.

GIMP: The GIMP (GNU Image Manipulation Program) is a program for viewing and performing image manipulation tasks, such as photo retouching, image composition, and image creation.

Gnome Ghostview: Gnome Ghostview (GGv) is a graphical application capable of displaying PostScript files.

Nautilus: Nautilus is the file manager and viewer of GNOME.

VMWare: VMWare is an application which is considered as an enterprise class virtual machine which is capable of running numerous versions of an operating system on a single computer.

Wine: Wine is an application implementing the Windows API on top of native X window system.

6.5 Review Questions

1. Explain various multimedia applications.
2. Describe the steps used for using digiKam with your digital camera.
3. Illustrate the concept of playing an audio CD.
4. Discuss the process of burning a CD. Illustrate with example.
5. Explain various graphics and imaging applications to work with images and graphics.
6. Describe various system applications provided by Linux.
7. Compare and contrast archive file and a compressed file.
8. Discussd the use of File Roller in compressing, decompressing, and archiving files in common Unix and Linux formats.
9. Describe the steps for creating archives with file roller.
10. Explain the concept of VMWare application with example.

Answers: Self Assessment

1.	digital camera	2.	Universal Serial Bus (USB)
3.	File manager	4.	KDE
5.	Burn	6.	GIMP
7.	Gnome Ghostview (GGv)	8.	GNOME Ghostview
9.	file manager	10.	Nautilus
11.	archive file	12.	compressed file
13.	File Roller	14.	new
15.	VMWare	16.	Wine

6.6 Further Readings

Notes

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.linuxjournal.com/article/2519

http://www.hackosis.com/top-20-free-linux-multimedia-applications/

http://www.brighthub.com/computing/linux/articles/49725.aspx

http://www.doudoulinux.org/web/english/documentation-7/applications-13/article/multimedia-applications.html

Notes

Unit 7: The Shell

CONTENTS

Objectives

Introduction

7.1 Shell as a Command Line Interface

7.1.1 Shell Variables

7.2 Types of Shell

7.2.1 BASH (Bourne-Again SHell)

7.2.2 C Shell (csh)

7.2.3 Korn Shell (ksh)

7.2.4 TCSH

7.3 Built-in Programs and External Programs

7.4 Summary

7.5 Keywords

7.6 Review Questions

7.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain the concept of shell
- Discuss different types of shell
- Discuss built-in programs and external programs

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. The shell is a user program or it is an environment provided for user interaction. It is a command language interpreter that executes commands read from the standard input device (keyboard) or from a file. Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc.

7.1 Shell as a Command Line Interface

The shell is defined as a program that takes commands from the keyboard and provides them to the operating system to perform. Earlier, shell was the only user interface available on a Unix computer. These days, we have graphical user interfaces (GUIs) in addition to command line interfaces (CLIs) such as the shell. The command interpreter is the interface between the user and the operating system, hence the name "shell".

Notes

Thus, we can say that the shell acts as an intermediary among the operating system and the user. The role of shell is to read the command line, interpret its meaning, carry out the command, and then return the result through the outputs.

The shell is considered as an executable file which is liable for interpreting the commands, transmitting them to the system, and returning the result. There are several shells. The most common shell is sh ("Bourne shell"), bash ("Bourne again shell"), csh ("C Shell"), Tcsh ("Tenex C shell"), ksh ("Korn shell"), and zsh ("Zero shell"). We have discussed these shells in this unit.

Did u know? Generally, the name of the different shells matches the name of the executable.

Every user has a default shell. This default shell will be launched on opening of a command prompt. We specify default shell in the dans configuration file /etc/passwd in the last field of the line corresponding to the user. It is possible to change the shell during a session by simply executing the corresponding executable file.

Example:

/bin/bash

We initialize shell by reading its overall configuration (in a file of the directory /etc/), followed by reading the user's own configuration (in a hidden file whose name begins with a dot, situated in the basic user directory, that is, home/user_name/.configuration_file). Then, a prompt is displayed as follows:

```
machine:/directory/current$
```

By default, the prompt (for most shells) include the machine name, followed by two points (:), the current directory, and then a character signifying the type of user connected:

- "$" specifies a normal user
- "#" specifies the administrator, called "root"

We can define a command line as a chain of characters which represents a command corresponding to an executable file of the system or rather a shell command and optional arguments (parameters):

```
ls -al /home/jf/
```

In the above command, ls signifies the name of the command, -al et /home/jean-francois/ are arguments. Arguments starting with - are known as options. Generally, for each command, there is a certain number of options which can be detailed by typing one of the following commands:

```
command --help

command -?

man command
```

The shell gives an interface to Linux. In this interface, commands can be typed or entered by using the keyboard. It is called the command line interface (CLI). To find out your current shell type following command:

echo $SHELL

```
ps $$

ps -p $$
```

Notes

The following example output shows that we are using bash shell:

```
  PID TTY          TIME CMD
13931 pts/4    00:00:00 bash
```

7.1.1 Shell Variables

A shell variable is a means of citing a character or numeric value. And unlike formal programming languages, a shell script doesn't require you to declare a type for your variables. Thus, you could assign a number to the variable

```
stuff
```

and then make use of it again in the same script in order to hold a string of characters.

Example: To access the value (contents) of a variable, prefix it with a dollar sign.

```
stuff=5
stuff='chocolate truffles'
```

Don't put any spaces before or after the equal sign, or it will produce an error. To assign a string including spaces, it is required to put quotation marks around the string.

Notes This is to note that there are several distinct ways to use quotations marks in a shell script.

Now let us discuss some differences between single quotation marks, double quotation marks, and the backslash character:

Single quotation marks will always get you exactly what's inside the quotation marks. Any characters that might otherwise have special meaning to the shell (like the dollar sign or the backslash) are treated literally.

Use double quotation marks when you want to assign a string including special characters the shell should act on.

We use backslash to escape a single character (such as $ or *) that might otherwise be considered as a special character by the shell.

Now we will discuss some examples that illustrate when to use each method of quoting.

Example:

howdy='Good Morning $USER !'

```
echo $howdy
Good Morning $USER !
howdy="Good Morning $USER !"
echo $howdy
Good Morning hermie !
```

In the first case, the howdy variable value would perhaps not be what you required. The single quotation marks caused Bash to not treat $USER as a variable. In the second case, the results look

Notes

much better. The double quotation marks allowed Bash to substitute the value of $USER, which is set automatically when you log in, in the string.

Example: This example presents a common error:

```
costmsg="Price is $5.00"

echo $costmsg

Actual result: Price is .00
```

Here, we thought quote the string, however the dollar sign tells Bash to use the value in the $5 variable, which is not what we required. We can easily solve the problem by prefixing the dollar sign with a backslash, as shown here:

```
$ costmsg="Price is \$5.00"

$ echo $costmsg

Actual result: Price is $5.00
```

Arguments and Other Special Variables

The values that you pass to a shell script are called arguments. Every value on the command line after the name of the script will be allocated to the special variables $1, $2, $3, and so on. The name of the currently running script is stored in the $0 variable.

Some other special variables that are useful in script writing are:

- $# The number of arguments
- $* The entire argument string
- $? The return code from the last command issued

Now we will show some examples working with arguments and other special variables.

Example: Create an executable script called testvars containing these lines:

```
echo "My name is $0"

echo "First arg is: $1"

echo "Second arg is: $2"

echo "I got a total of $# arguments."

echo "The full argument string was: $*"
```

Now if you run this script, here's what you'll see:

```
$ ./testvars birds have lips

My name is testvars

First arg is: birds

Second arg is: have

I got a total of 3 arguments.

The full argument string was: birds have lips
```

Notes

Self Assessment

Fill in the blanks:

1. The ..... is a program that takes your commands from the keyboard and gives them to the operating system to perform.
2. "#" specifies the administrator, called "....."..
3. A ..... is a chain of characters representing a command which corresponds to an executable file of the system.
4. A ..... in a shell script is a means of referencing a numeric or character value.
5. The ..... is used to escape a single character that might otherwise be treated as a special character by the shell.

7.2 Types of Shell

There are many different types of shells in the Linux operating system. These command-line interfaces provide powerful environments for software development and system maintenance. Though shells have many commands in common, each type has unique features. Over time, individual programmers come to prefer one type of shell over another; some develop new, enhanced shells based on previous ones.

Table 7.1 shows the different shells available in Linux:

Table 7.1: Different Types of Shell

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).

In order to find all the available shells in your system, we type the following command:

```
$ cat /etc/shells
```

Caution Every shell does the same job, however each understands different command syntax and offers different built-in functions.

These shells are discussed as below.

7.2.1 BASH (Bourne-Again SHell)

BASH, Bourne-again shell, is considered a free software UNIX shell which is written for the GNU Project. BASH the name is a pun on the name of the Bourne shell (sh), an early and

Notes

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.

It is the default shell on most systems which are built on top of the Linux kernel and on Mac OS X. Also, it can be run on most Unix-like operating systems.Bash has been ported to MS Windows using Subsystem for UNIX-based Applications (SUA), or POSIX emulation provided by Cygwin and MSYS.

Did u know? BASH has been ported to MS-DOS by the DJGPP project and to Novell NetWare.

7.2.2 C Shell (csh)

Bill Joy developed a shell for the BSD Unix system. This shell is known as C shell (csh). Originally, it was derived from the 6th Edition Unix /bin/sh , 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.

7.2.3 Korn Shell (ksh)

In the early 80's, David Korn (AT&T Bell Laboratories) developed a unix shell known as Korn shell (ksh). 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.

By means of ksh, we can edit the command line in a WYSIWYG fashion by hitting the suitable 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.

Since 2000, Korn shell has been an open source software, originally under a license peculiar to AT&T. However, 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.

Even though many improvements were added by the ksh93 version, some vendors still ship their own version of the older ksh88 as /bin/ksh, sometimes with extensions ship ksh88, all other Unix vendors migrated to ksh93 and even Linux distributions started shipping ksh93.

Notes There ksh93 consists of two modified versions 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.

Notes

SKsh is an AmigaOS version, that provides various Amiga-specific features like ARexx interoperability.

Another commercial ksh reimplementation is MKS Inc.'s MKS Korn shell. It was included with Microsoft's Services for Unix (SFU) up to version 2.0. As per David Korn, the MKS Korn shell was not completely compatible with his own Korn shell implementation in 1998.

When the SFU Version 3.0 was introduced, Microsoft has substituted 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.

7.2.4 TCSH

TCSH is considered as an improved version of the Berkeley UNIX C shell. However, it is a completely compatible version of 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.

We can call TCSH a programming language with conditional statements. In tcsh, the 't' occurs from the T in TENEX. TENEX is an operating system which encouraged 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 editions of Mac OS X shipped with tcsh as the default shell, however the default for new accounts is bash as of 10.3.

Caution Iowa State's implementation of MIT's Project Athena (Project Vincent) by default uses tcsh as the default shell, even though users can change this.

TCSH is considered as the default shell of FreeBSD and its descendants such as PC-BSD, DragonFly BSD, and DesktopBSD.

Task Make distinction between ksh and tcsh shell.

Self Assessment

Fill in the blanks:

6. ..... is the default shell on most systems built on top of the Linux kernel as well as on Mac OS X.
7. The ..... is a Unix shell developed by Bill Joy for the BSD Unix system.
8. The ..... shell is backwards-compatible with the Bourne shell and includes many features of the C shell as well.
9. ..... is a command language interpreter usable both as an interactive login shell and a shell script command processor.
10. ..... provides access to the Tk widget toolkit.

Notes

7.3 Built-in Programs and External Programs

All the Linux commands can be classified into two types:

- Internal commands
- External commands

The commands which are directly executed by the shell are known as internal commands. These are built-ins in the shell. Internal commands do not depend on paths as they are not coded in files. Dissimilar to external commands, when an internal command is executed, no process is created. Commands like pwd, cd ,echo comes under the category of the internal commands.

The files present in the $PATH are considered as external commands. This is to note that, in Linux, everything is represented in the form of files. The files do not get executed if they are not present in the path specified by the $PATH variable. Likewise, if the files are available but are not present int the path specified by $PATH, the commands cannot be executed. Another point worth noting is that every time an external command gets executed, a new process gets spawned. Commands like ls,cp etc. comes in this category.

Now let us differentiate between internal and external commands.

There is a Linux command 'type', which determines how its arguments would be interpreted if they were utilised as a command name. We can use this command to identify among the two types of commands. Execute the command : type <command>

In the case when the output states that the command is shell built-in, it is an internal command. Or else, if the output states that the command is present in /bin, then it is an external command.

Example: cd and pwd are examples of internal commands.

```
1.  savita@Amrita:~$ type cd
2.  cd is a shell builtin
3.  savita@Amrita:~$ type pwd
4.  pwd is a shell builtin
5.  savita@Amrita:~$
```

cp and mv are examples of external commands.

```
1.  savita@Amrita:~$ type cp
2.  cp is /bin/cp
3.  savita@Amrita:~$ type mv
4.  mv is /bin/mv
5.  savita@Amrita:~$
```

Nevertheless, there might be certain commands having files in /bin directory and simultaneously, are also built-ins in the shell. Then, the first preference would always be given to the internal command. It means that even if the corresponding file did not exist, it would be executed by the shell as a built-in.

Example: Linux command echo.

Table 7.2 shows some of the differences between internal and external commands:

Notes

Table 7.2: Differences between internal and external commands

Table 7.2: Differences between Internal and External Commands

External Linux Commands	Internal Linux Commands
External commands are executed by the kernel.	Internal commands are executed by the shell.
A separate process is spawned every time a new external command is executed.	No new process is created.
These are separate files in /bin directory. The execution of these commands happens through the execution of their corresponding files in /bin directory.	These are built-ins in the shell. The execution of these commands happens through the execution of their corresponding files in /bin directory.
A few examples are cp, mv, etc.	Some examples are cd, pwd, etc.

Task Illustrate the execution of external commands.

Self Assessment

Fill in the blanks:

11. ..... commands are those which are directly executed by the shell.
12. ..... commands are files present in the $PATH.
13. If the output states that the command is shell ....., it is an internal command.
14. The execution of internal commands happens through the execution of their corresponding files in ..... directory.
15. External linux commands are executed by the .....

7.4 Summary

- The shell is a program that takes your commands from the keyboard and gives them to the operating system to perform.
- The shell provides an interface to Linux where you can type or enter commands using the keyboard. It is known as the command line interface (CLI).
- A variable in a shell script is a means of referencing a numeric or character value.
- 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 which added many feature improvements over the Bourne shell, such as aliases and command history.
- The Korn shell (ksh) is a Unix shell which is backwards-compatible with the Bourne shell and includes many features of the C shell as well.

Notes

- TCSH is a command language interpreter usable both as an interactive login shell and a shell script command processor.
- Internal commands are those which are directly executed by the shell. External commands are files present in the $PATH.

7.5 Keywords

Arguments: Arguments are the values you pass to a shell script.

BASH: BASH is a free software UNIX shell written for the GNU Project.

csh: The C shell (csh) is a Unix shell which added many feature improvements over the Bourne shell, such as aliases and command history.

External commands: External commands are files present in the $PATH.

Internal commands: Internal commands are those which are directly executed by the shell.

ksh: The Korn shell (ksh) is a Unix shell which is backwards-compatible with the Bourne shell and includes many features of the C shell as well.

Shell: The shell is an executable file responsible for interpreting the commands, transmitting them to the system, and returning the result.

TCSH: TCSH is a command language interpreter usable both as an interactive login shell and a shell script command processor.

Variable: A variable in a shell script is a means of referencing a numeric or character value.

7.6 Review Questions

1. Explain how shell acts as an intermediary between the operating system and the user.
2. Describe the concept of shell variables with example.
3. Make distinction between single quotation marks and double quotation marks.
4. What are the different types of shell? Explain.
5. Compare and contrast C shell and Korn shell.
6. TCSH is a command language interpreter usable both as an interactive login shell and a shell script command processor. Comment.
7. What are the built-in programs in Linux? Discuss.
8. Illustrate the difference between internal and external commands.
9. Discuss some special variables that are useful in script writing.
10. Explain the features of Korn Shell.

Answers: Self Assessment

1. Shell
2. Root
3. command line
4. Variable
5. backslash
6. Bash
7. C shell (csh)
8. Korn

Notes

9. TCSH
10. TKSH
11. Internal
12. External
13. built-in
14. /bin
15. kernel

7.7 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.dis.uniroma1.it/~bordino/shell-tutorial.pdf

http://www.linuxforum.com/threads/1541-Important-Concepts-For-Linux-Beginners-%E2%80%93-Shells-and-Utilities

http://www.tldp.org/LDP/gs/node5.html

http://docstore.mik.ua/orelly/linux/lnut/ch06_01.htm

Notes

Unit 8: Shell Commands

CONTENTS

Objectives

Introduction

8.1 Common Shell Commands

8.1.1 Basic Commands

8.1.2 Editor

8.1.3 User Account

8.1.4 Network Commands

8.1.5 Achieve Commands

8.1.6 Help Commands

8.1.7 Process Commands

8.1.8 Other Commands

8.2 Special Keys and Shortcuts

8.3 Command Line Syntax

8.4 Summary

8.5 Keywords

8.6 Review Questions

8.7 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss common shell commands
- Discuss special keys and shortcuts
- Explain command line syntax

Introduction

At the core of each modern Linux computer is the "terminal." The terminal developed from the text-based computer terminals of the 1960s and 1970s, which themselves replaced punch cards as the main method to interact with a computer. It's also known as the command shell, or just "shell." In this unit, we will discuss some commonly used shell commands. Also we will discuss some shortcuts and command line syntax.

8.1 Common Shell Commands

Linux consists of various commands and each command has its own set of options. All the commands perform different operations. Now let us categorise all these commands as per their function. By doing this, you will know the basic commands and have some measure of control over the Linux command line.

Notes

After discussing these commands, you will be able to perform different function according to your requirement in the Linux command line.

8.1.1 Basic Commands

The basic commands are divided into two sub-categories, that is, Directory commands and File commands.

Directory Commands

Some of the basic directory commands are defined below:

- pwd: Print working directory
- mkdir: Create directories
- cd: Change the current directory
- rmdir : Remove directories

File Commands

Some of the basic file commands are defined below:

- ls : List directory contents
- rm : Remove files
- cp : Copy files from a source to the same or different target(s)
- mv : Move file to different targets
- cat : Read one or more files and print them to standard output. If you need to view contents of a short file, cat is recommended
- cmp: Compare two files byte by byte
- wc: Print the number of new lines, words, and bytes in files
- du : Estimate disk usage of each file and recursively for directories
- find: Search for files in directory hierarchy, e.g. find notes.txt
- grep: Print lines matching a pattern, e.g. grep –i topic notes.txt (topic is the pattern)
- sort: Sort lines of text files

8.1.2 Editor

Every Linux program is considered as an executable file.

Example: The cp command is provided by the file in /bin/sh which holds the list of machine instructions.

Likewise, if you are installing a package, let's say vsftpd, your focus will be modifying its configuration file, vsftpd.conf present in /etc directory. In this case, we use editors. We use 'Vim' frequently.

Did u know? Vim is an advanced text editor that occurs a more complete feature as compared to the 'Vi' text editor. The other text editors are: nano, vi, kate (KDE Advanced Text Editor), gedit (graphical user interface). For example, vim rabi.c (vim filename).

Notes

8.1.3 User Account

In Linux, you can access to almost each and every file (configuration files, system, text files etc.) with no restriction and interruption. You need to be very careful while doing work as the root (super user). Thus, managing user accounts and groups is an important part of a system administrator.

Example: An organization XYZ consists of three departments, that is, Marketing, Technical and Account. Each department have 3–4 employees. The organization demands you to verify the users of marketing, technical and account departments so that the employees of each group will be able to view his/her own department file.

Given a situation, if you are familiar with the process of managing these accounts, you can easily set permissions for the users mentioned above. If not, The commands given below are essential for the task.

- useradd : This command is used for creating user account.

  This command can be executed by administrators only. On debian, you should use adduser. For other options like adding expiry date, home directory etc., refer man useradd.

- passwd : This command is used for changing user password.

  If the user has set password before, he/she will be prompted for the first password whereas superuser is permitted to bypass the step so that forgotten passwords may be changed.

- usermod : This command is used to modify user account.
- userdel : This command is used to delete a user account and related files. This command can be executed by administrators only.
- groupadd : This command is used to create a new group.
- groupdel : This command is used delete the group and entries referring to the group.
- groupmod : This command is used to modify a group definition on the system.
- chmod : This command is used to modify properties for users.
- chown : This command is used to change file owner and group.
- chgrp : This command is used to change group ownership.

8.1.4 Network Commands

Linux is mainly recognized for its use in servers. In 2009, it held a server market share ranging between 20–40%.

Caution It is necessary to know the commands to check the ip address, download files from the net, get DNS, etc.

Various network commands are discussed below:

- wget: This command is a non-interactive network downloader.
- Even if a download fails because of a network problem, it will keep trying again and again until the complete file has been retrieved. The server will instruct to continue to download from where it it left off.
- $ wget url-for-file.

Notes

- ping: send ICMP ECHO_REQUEST to network hosts, you will get back ICMP packet if the host responds. We use this command if there is a doubt whether the computer is connected or not.
- $ ping IP or host name.
- hostname : This command is used to show or set the system's host name.
- dnsdomainname : This command is used to show the system's DNS domain name.
- netstat: This command displays the status of ports i.e. which ports are open, closed, waiting for connections. It displays the contents of /proc/net file.
- ifconfig: This command is used to configure a network interface, or to display their current configuration. It is also useful to get the information regarding IP address, Subnet Mask, set remote IP address, Netmask etc.
- ifup : This command is used to bring a network interface up.
- ifdown : This command is used to take a network interface down.

8.1.5 Achieve Commands

Let us suppose that you want to install a package from its source code. You observed that the source code of the package is archived in a file xxx.tar. In this condition, the command-line utility 'tar' proves to be a critical resource for you. The 'tar' is perhaps the most popular Linux backup utility. If the 'tar' file is compressed with the compression utility like 'bzip' or 'gzip', the resulting file is the famous 'tarballs' which is a common way to deliver software installation archives.

tar: It is an archiving program which is designed to store and extract files from an archive known as a tarfile.

Options:

-c : create a new archive

-r : append files to the end of an archive

-t : list the contents of an archive

-u : only append files that are newer than copy in archive

-x : extract files from an archive

-C : change to directory Dir

-j : filter archive through bzip2, use to decompress .bz2 files

-v : verbosely list files processed

-f : use archive file

-z : filter the archive through gzip

Examples:

`tar -xvf test.tar` (extract foo.tar to the current location)

`tar -xvzf test.tar.gz` (extract gzipped test.tar.gz)

`tar -cvf test.tar foo/` (compress the contents of foo folder to foo.tar)

Notes

Task Describe the method to deliver software installation archives.

8.1.6 Help Commands

There are manual pages for almost every command of Linux. Manual pages can be accessed by means of man command. The man command provides documentation of the command. If you type:

$ man ls

This will result in the manual page of ls with its name, synopsis, description, author, copyright, etc.

Did u know? There is a manual page for the man command itself.

If you want to know a brief description of the command, use -help option with the command.

$ ls -help

Also, info command can be used to have a quick overview of the command.

$ info ls

Remembering all the commands in Linux together with all its options is a very tough job. Thus it is recommended to memorize the command and options which are used frequently and leave the rest to the HELP commands.

8.1.7 Process Commands

In order to execute a command in the background, it is necessary to place an ampersand(&) on the command line at the end of the command. A user job number (placed in brackets) and a system process number are displayed.

Notes The number by which the system identifies the job is known as a system process number whereas the number by which the user identifies the job is known as a user job number.

$ sudo cp -rf * ~/ss &

[1] 9144

$

- **jobs:** This command is used to list the jobs being run at the background.

  ```
  $jobs
  [1]-  Running   sudo  cp  -rf  *  ~/ss  &
  [2]+  Running   sudo  cp  -rf  *  ~/yy &
  ```

  The '+' sign signifies that the job is currently being processed , '-' sign signifies the upcoming jobs to be executed. The '% ' is used with the job number and it references a job.

Notes

- **fg:** On using this commad, a process running in the background will be processed in the foreground.

```
$ fg % 2
cat  *.cpp  > mytext
$
```

- **kill:** This command cancels a job running in the background. It takes argument either the user job number or the system process number.

```
$jobs
[1]   +  Running    cp  *.c  > mytext
[2]  - Running    cp  *.dat >>mytext
$kill %2
```

- **bg:** This command places a suspended job in the background.

```
$ cat *.cpp > mytext
^Z
$bg
```

Caution Ctrl + Z will suspend the process running at the moment.

- **ps:** This command reports a snapshot of the current processes.
- **top:** This command is used to display Linux tasks.
- **at:** This command executes commands at a specified time.

```
$ at  8:00
at >  echo "HI" > /dev/tty1
```

In order to return to the command line, press 'ctrl + d' . This will display the message in tty1 at 8'o clock.

- To view the schedule: $ atq
- To cancel a job: $atrm 5 [job ID]
- **crontab:** crontab is a file containing the schedule of entries to run at specified times.
- **shutdown:** This command is used to bring the system down.
  - -r: This option requests that the system be rebooted after it has been brought down.
  - -c: This option is used to cancels a running shutdown.

Task Compare and contrast system process number and user job number.

8.1.8 Other Commands

Some other commands are defined as below:

- **whoami:** This command displays the login name of the current effective user.
- **logname:** This command is used to print user´s login name.

Notes

- **quota:** This command is used to display disk usage and limits, e.g., $ quota –v
- **su:** This command is used to switch to super user or change user ID.
- **which:** This command is used returns the pathnames of the files which would be executed in the current environment.

  Type $which ls, you will get /bin/ls.

Self Assessment

Fill in the blanks:

1. ..... command is used to change the current directory.
2. ..... command is used to print the number of new lines, words, and bytes in files.
3. 'Vim' is an advanced ..... that comes with a more complete feature than the 'Vi' text editor.
4. ..... command displays the status of ports i.e. which ports are open, closed, waiting for connections.
5. You can access the manual pages using ..... command.
6. A ..... number is the number by which the system identifies the job.
7. .....command places a suspended job in the background.
8. ..... command displays the login name of the current effective user.

8.2 Special Keys and Shortcuts

As we know, BASH is considered as the default shell on most Linux operating systems. You plan to spend a lot of time at the command line, then it is important for you to get familiar with some hotkeys. These shortcuts will save much of your time if you learn them.

Table 8.1 shows the various shortcuts used in Linux.

Table 8.1: Shortcuts Used in Linux

Shortcut	Description
Ctrl + A	Go to the beginning of the line you are currently typing on.
Ctrl + E	Go to the end of the line you are currently typing on.
Ctrl + L	Clears the Screen, similar to the clear command.
Ctrl + U	Clears the line before the cursor position. If you are at the end of the line, clears the entire line.
Ctrl + H	Same as backspace.
Ctrl + R	Let's you search through previously used commands.
Ctrl + C	Kill whatever you are running.
Ctrl + D	Exit the current shell.
Ctrl + Z	Puts whatever you are running into a suspended background process. e.g. restores it.

Contd....

Notes

Ctrl + W	Delete the word before the cursor.
Ctrl + K	Clear the line after the cursor.
Ctrl + T	Swap the last two characters before the cursor.
Esc + T	Swap the last two words before the cursor.
Alt + F	Move cursor forward one word on the current line.
Alt + B	Move cursor backward one word on the current line.
Complete recent commands with "!"	Try this: Type "!" followed by the first couple of letters of a recent command and press ENTER! For example, type: find /usr/bin -type f -name m* and now type: !fi
Tab	If you type a partial command or filename that the shell recognizes, you can have it automatically completed for you if you press the TAB key. Try typing the first few characters of your favourite Linux command, then hit TAB a couple of times to see what happens.
"history" command	Show your complete command history.
Up/Down Arrow Keys	Scroll through your most recent commands. You can scroll back to an old command, hit ENTER, and execute the command without having to re-type it.
Scrolling the screen with Shift- PageUp and Page Down	Scroll back and forward through your terminal.

Notes This is to note that some of these commands may not work if you are accessing BASH through a telnet/ssh session, or depending on how you have your keys mapped.

Self Assessment

Fill in the blanks:

9 The default shell on most Linux operating systems is called .....

10. A shortcut Ctrl + T is used to Swap the last two ..... before the cursor.

11. Shortcut ..... clears the line before the cursor position.

8.3 Command Line Syntax

Commands can be run by themselves, or you can pass in additional arguments to make them do different things. Typical command syntax is given below:

command [-argument] [-argument] [--argument] [file]

Notes

Examples:

ls: This command list files in current directory

ls -l: It lists files in "long" format

ls -l --color: It shows colourized output

cat filename: It shows contents of a file

cat -n filename: It shows contents of a file, with line numbers

Let us discuss some in detail. Let us consider two commands pwd and ls.

- ***The pwd Command:*** The command ;pwd' stands for present working directory. This command shows the name and location of the current directory, which is the directory (also known as a folder on some operating systems) in which the user is currently working. To use this, type the word pwd at the keyboard and then press the ENTER key:

  ```
  pwd
  ```

  If the terminal window or console has just been opened, then the monitor screen will show /home followed by another forward slash and then the name of the user's home directory. Remenber that the user's home directory is usually the same as the user name.

  Example: If the user has a user name of miky, the line would say /home/miky. This is because a user begins working at the command line in its home directory, home directories are located in the directory named /home, and a user's home directory usually has the same name as the user name.

- ***The ls Command:*** Another most basic and frequently used commands on Unix-like operating systems is ls. This command is equivalent to DIR on MS-DOS systems. Also, it lists the contents of a directory. When used just by itself, it offers a list of the names of the objects (that is, files, directories and links) in the current directory, i.e.,

  ```
  ls
  ```

  You can see that various names are listed, but no additional information is provided about them. This is often sufficient. However, in most of the cases it is desired to get additional information about each directory object. Getting such information can be easily attained by using ls together with one or more of its options. We can define an option as a letter (or in some cases a word) that follows the command after a space and that tells the command how to behave. We can use various options together, and they are usually preceded directly by a hyphen (without any intervening space). The options available can vary as per the specific command.

  Example: The -a option tells ls to show all objects in the directory, inclusive of hidden files or hidden directories. Hidden objects are those whose names are normally not visible either in the command line or when examining the contents of a directory in a GUI. Their names are preceded directly by a period, such as .file1. Thus to see the names of all objects in the current directory, the following command should be typed in and then the ENTER key pressed:

  ```
  ls -a
  ```

  Many commands have several options, and ls is no exception. But, only some of them are often used. Another generally used option for ls is –l. This option provides a long listing, i.e., it provides much information about each object in addition to just its name. This additional information includes the type of object (e.g., file, directory or link), its permissions (i.e., who has access to it for reading, writing and/or executing), its owner (which is by default the same as its creator), and the date and time of creation.

Notes

When used together with both its -a and -l options, we will write the command as follows:

```
ls -al
```

Another common option for ls is –s. This option shows the size of each file in kilobytes. This option is often used together with the above two options as

```
ls -als
```

Similar to most of the commands, ls can accept input data, which is known as an argument. In the case of ls, arguments are the names of directories about which it is desired to get information. Any number of directories can be listed as arguments. They all follow ls and any options, and they are separated by at least one blank space.

Example: If you need to see the names of all the objects in both the directories /bin and /home, the following command would be used:

```
ls -a /bin /home
```

As is the case with /home, /bin is a standard first tier directory in the root directory, which is the single directory that includes all other directories and their subdirectories on a Unix-like operating system and which is represented by a forward slash (/). Similar to all other first-tier directories in the root directory, their names start with a forward slash, in part so that they will not be confused with directories with similar names that are not first tier directories of the root directory.

Self Assessment

Fill in the blanks:

12. ..... command shows the name and location of the current directory.
13. ..... command, when used just by itself, it provides a list of the names of the objects in the current directory.
14. ..... option tells ls to show all objects in the directory, inclusive of hidden files or hidden directories.
15. ls can accept input data known as an .....

8.4 Summary

- Linux has more than 650 commands and every command has its own set of options all performing different operations.
- Every Linux program is an executable file.
- In Linux, you can say that using the 'root' account is like having the powers of God. You will have access to almost each and every file (configuration files, system, text files etc.) with no interruption and restriction.
- Linux is predominantly known for its use in servers.
- The 'tar' is probably the most popular Linux backup utility.
- You can access the manual pages using man command. The man command offers documentation of the command.
- In order to execute a command in the background, place an ampersand(&) on the command line at the end of the command. A user job number (placed in brackets) and a system process number are displayed.

- Commands can be run by themselves, or you can pass in additional arguments to make them do different things.

8.5 Keywords

cd: This command is used to switch to another directory.

ls: The command ls is used to the contents of a directory.

mkdir: The command mkdir is used to create a new directory.

netstat: It displays the status of ports ie. which ports are open, closed, waiting for connections.

pwd: The command pwd is used to display the full path name of the current directory.

System process number: A system process number is the number by which the system identifies the job.

tar: It is an archiving program designed to store and extract files from an archive known as a tarfile.

User job number: A user job number is the number by which the user identifies the job.

8.6 Review Questions

1. What are the common shell commands?
2. Discuss various directory commands and file system.
3. Describe the commands used for managing User Account.
4. Make distinction between userdel and groupdel command.
5. What are network commands? Illustrate.
6. Explain an archiving program which is designed to store and extract files from an archive.
7. Illustrate the use of kill command with example.
8. What are the different special keys and shortcuts of shell programing?
9. What is the command line syntax? Discuss with example.
10. Illustrate the use of whoami command with example.

Answers: Self Assessment

1. Cd
2. Wc
3. text editor
4. Netstat
5. man
6. system process
7. bg
8. Whoami
9. Bash
10. Characters
11. Ctrl + U
12. Pwd
13. ls
14. -a
15. argument

Notes

8.7 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Online links

http://www.fortystones.com/40-linux-shell-commands-beginners/

http://www.freeos.com/guides/lsst/

http://vic.gedris.org/Manual-ShellIntro/1.2/ShellIntro.pdf

http://www.kau.edu.sa/Files/830/Files/60761_Linux.pdf

Notes

Unit 9: The File System

CONTENTS

Objectives
Introduction
9.1 Concept of the File System
9.2 Anatomy of a File
9.2.1 Mounting File Systems
9.2.2 The Linux File System Locations
9.2.3 The Root File System /
9.2.4 The Variable Data Location /var
9.2.5 The Userland Location /usr
9.2.6 The Home Location /home
9.2.7 Permissions
9.2.8 Attributes
9.3 File Search Utilities
9.3.1 Locate Command
9.3.2 Find Command
9.3.3 GNOME
9.4 Navigating File System
9.4.1 Commands for Directory Navigation
9.5 Summary
9.6 Keywords
9.7 Review Questions
9.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Explain the concept of file system
- Discuss anatomy of a file
- Discuss file search utilities
- Explain the concept of navigating file system

Introduction

The Linux file system is considered as a hierarchically structured tree. Here, each location has its different meaning. The structure of file system is standardized through the file system hierarchy standard. Lately however, more and more distributions are making a small change towards their

Notes

file system layout (but all consistent) so the standard is in need of updating. Of course, a file system is always stored on media (be it a hard drive, a CD or a memory fragment). In this unit, we will discuss the anatomy of a file, various file search utilities and the concept of navigating file system.

9.1 Concept of the File System

We can define a file system as the way in which the computer stores and retrieves all your files. These files include your programs, documents, help files, music, games, etc. In the Windows world we have the concept of files and folders. A folder (also known as a directory) is nothing more than a container for different files so that you can organise them better. In Linux, the same concept holds true. That is, you have files, and you have folders in which these files are organised.

In Linux, files are stored according to the function performed by them. Alternatively, all help files for all programs will go into one folder made just for help files, all the executable (.exe) files will go into one folder for executable programs, all programs configuration files will go into a folder meant for configuration files. This layout has a few significant advantages as you always know where to look for a particular file.

Example: To find the configuration file for a program, you need to find it in the actual program's installation directory.

In Linux, everything is configurable to the smallest level. Thus if you want to install a program and store all its files in one folder, you can, but it will just make things complicated. Also, it will miss out on the benefits of a file system that groups files by the function they perform rather than randomly.

Linux uses a hierarchical file system. Alternatively, there is no concept of 'drives' such as c: or d:. Everything begins from what is known as the '/' directory (root directory). This is the top most level of the file system and all folders are placed at some level from here. As a result of files being stored according to their function on any Linux system, you will see many of the same folders. These are 'standard' folders that have been pre-designated for a particular purpose.

Example: The 'bin' directory will store all executable programs.

Notes Remember that in Linux a forward slash (e.g.: /bin) is used. This means that you are telling the system where the directory is in relation to the root or top level folder.

Self Assessment

Fill in the blanks:

1. A file system is the way in which the computer stores and retrieves all your .....
2. A ..... is a container for different files so that you can organise them better.
3. Linux uses an ..... file system.

9.2 Anatomy of a File

We consider file system as a tree-shaped structure. The root of the tree is identified by the slash character: "/". The root of the tree is coincidentally known as the file system root, but it is always

Notes

portrayed as being above all other. It is the highest place you can go to. Beneath it are almost always only directories:

```
~$ cd /
~$ ls -F
bin/     home/     opt/      srv/     var/
boot/    lib/      proc/     sys/
dev/     media/    root/     tmp/
etc/     mnt/      sbin/     usr/
```

The ls -F commands displays the content of the root location. However, it appends an additional character to special files. For instance, it appends a "/" to directories, an "@" to symbolic links and a "*" to executable files. The benefit is that you can easily see what type of files you have. A popular method of representing the file system is through a tree.

Example: An example would be for the top level:

```
/
+- bin/
+- boot/
+- dev/
+- etc/
+- home/
+- lib/
+- media/
+- mnt/
+- opt/
+- proc/
+- root/
+- sbin/
+- srv/
+- sys/
+- tmp/
+- usr/
`- var/
```

This is to note that he more you descend, the larger the tree becomes and it will soon be too difficult to put it on a single view. Still, the tree format is a good method of presenting the file system as it shows exactly how the file system looks like.

```
/
+- bin/
+- ...
+- home/
|  +- thomas/
|  |  +- Documents/
```

Notes

```
|  |  +- Movies/
|  |  +- Music/
|  |  +- Pictures/         <-- You are here
|  |  |  `- Backgrounds/
|  |  `- opentasks.txt
|  +- jane/
|  `- jack/
+- lib/
+- ...
`- var/
```

Before we explain the various locations, let's first consider how the file system is stored on one (or more) media.

9.2.1 Mounting File Systems

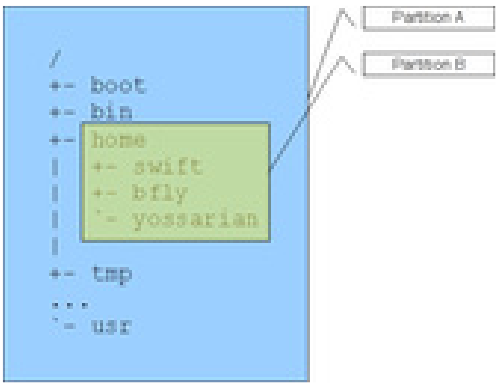
The root of a file system is mostly stored on a partition of a disk. In many cases you would want to combine multiple partitions for a single file system. Combining one partition with the file system is known as mounting a file system. Your file system is always seen as a tree structure, however parts of a tree (a branch) can be located on a different partition, disk or even other medium (network storage, USB stick , DVD, etc).

Mounting

Let us suppose that the root of your file system is stored on one partition and the files of all the users are stored on another. This would signify that /, and everything below it, is on one partition except /home and everything below that, which is on a second one.

Example: Figure 9.1 show the two partitions used for the file system structure.

Figure 9.1: Two Partitions Used for the File System Structure



Source: http://swift.siphos.be/linux_sea/linuxfs.html

In case of mounting, you are required to identify a location of the file system as being a mount point (in Figure 9.1, /home is the mount point) under which each file is actually stored on a different location (in Figure 9.1, everything below /home is on the second partition). The partition you "mount" to the file system doesn't need to know where it is mounted on. In fact, it doesn't.

Notes

You can mount the users' home directories at /home (which is preferable) but you could very well mount it at /srv/export/systems/remote/disk/users.

The mount command by itself, without any arguments, displays a list of mounted file systems:

Example:

```
$ mount
/dev/sda8 on /             type ext3 (rw,noatime)
proc      on /proc         type proc (rw)
sysfs     on /sys          type sysfs (rw,nosuid,nodev,noexec,relatime)
udev      on /dev          type tmpfs (rw,nosuid,relatime,size=10240k,mode=755)
devpts    on /dev/pts     type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
/dev/sda7 on /home         type ext3 (rw,noatime)
none      on /dev/shm      type tmpfs (rw)
/dev/sda1 on /mnt/data     type ext3 (rw,noatime)
usbfs     on /proc/bus/usb type usbfs (rw,noexec,nosuid,devmode=0664,devgid=85)
```

The example discussed above, even though bloated with various other file systems we know nothing about yet, tells us that the file system can be seen as follows:

```
/             (on /dev/sda8)
+- ...
+- dev/       (special: "udev")
|  +- pts     (special: "devpts")
|  `- shm     (special: "none")
+- proc/      (special: "proc")
|  `- bus/
|     `- usb/ (special: "usbfs")
+- sys/       (special: "sys")
+- home/      (on /dev/sda7)
`- mnt/
   `- data/   (on /dev/sda1)
```

Overlooking the special mounts, it can be seen that the root of the file system is on device /dev/sda8. From /home onwards, the file system is stored on /dev/sda7 and from /mnt/dataonwards, the file system is stored on /dev/sda1.

Mounting allows programs to be uncertain about where your data is structured. From an application (or user) point of view, the file system is one tree. Under the hood, the file system structure can be on a single partition, but also on a dozen partitions, network storage, removable media and more.

File Systems

Every medium which can enclose files is internally structured. The appearance of this structure is part of the file system it uses. Windows users might remember that originally, Microsoft Windows used FAT16 and later on FAT32 before they all migrated to one of the many NTFS revisions currently in use by Microsoft Windows. Well, all these are in fact file systems, and Linux has its own set as well.

Notes

In Linux, it's the partitions are not required to have one possible file system. As long as it understands it and the file system supports things like permissions and ownership, you are free to select any file system. Actually, during most distribution installations, you are asked which file system to choose. The following is a small list of popular file systems around, each with a brief explanation on its advantages and disadvantages.

- The ext2 file system is Linux' old, yet still used file system. It symbolizes extended 2 file system and is quite simple. It has been in use almost since the birth of Linux and is quite resilient against file system fragmentation – although this is true for almost all Linux file systems. It is however slowly being replaced by journalled file systems.
- An improvement on the ext2 file system is the ext3 file system. It adds, amongst other things, the concept of journalling.
- An improvement on the ext3 file system is the ext4 file system. It adds, amongst other things, support for very large file systems/files, extents, pre-allocation and delayed allocation and more. The ext4 file system is backwards compatible with ext3 as long as you do not use extents. Ext4 is frequently seen as the default file system of choice amongst administrators and distributions.
- The reiserfs file system is written from scratch. It also offers journalling, however speed is its main focus. The file system provides quick access to locations with hundreds of files inside and keeps the disk footprint for small files small. Even though quite popular a few years back, the file system has been seeing a lack of support through its popular years and is not frequently advised by distributions any more.

Did u know? The successor of reiserfs, reiser4, is still quite premature and is, due to the imprisonment of the main developer Hans Reiser, not being developed that actively any more.

- The btrfs file system is considered as a promising file system. It addresses concerns regarding huge storage backend volumes, multi-device spanning, snapshotting and more. Although its primary target was enterprise usage, it also offers interesting features to home users such as online grow/shrink (both on file system as well as underlying storage level), object-level redundancy, transparent compression and cloning.

A journal of file system keeps track of file write operations. It first performs the write (like adding new files or changing the content of files) in a journal first. Then, it performs the write on the file system itself after which it removes the entry from the journal. This set of operations ensures that, if at any point the file system operation is interrupted (for instance through a power failure), the file system is able to recover when it is back up and running by either replaying the journal or removing the incomplete entry: as such, the file system is always at a consistent state.

Generally, it is not possible to switch among file systems (except ext2 <> ext3) but as most file systems are mature enough you do not need to panic "to chose the correct file system".

Now, if we take a look at our previous mount output again, observe that there is a part of the line that says which "type" a mount has. Well, this type is the file system used for that particular mount.

```
$ mount
/dev/sda8 on /            type ext3 (rw,noatime)
proc      on /proc        type proc (rw)
sysfs     on /sys         type sysfs (rw,nosuid,nodev,noexec,relatime)
udev      on /dev         type tmpfs (rw,nosuid,relatime,size=10240k,mode=755)
devpts    on /dev/pts     type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620)
```

Notes

```
/dev/sda7 on /home         type ext3 (rw,noatime)
none      on /dev/shm      type tmpfs (rw)
/dev/sda1 on /mnt/data     type ext3 (rw,noatime)
usbfs     on /proc/bus/usb type usbfs (rw,noexec,nosuid,devmode=0664,devgid=85)
```

It can be seen that all partitions are typed as ext3. But what are those other file systems that you notice? These are discussed below.

- ***Proc:*** It is a special file system which doesn't exist on a device. However, it is a kind of gateway to the Linux kernel. Everything you see below /proc is something the kernel displays the moment you read it. It is a way to communicate with the kernel (and vice versa) using a very simple interface: file reading and file writing, something well supported.

  proc is known to be a pseudo file system: it does not contain real files, but runtime information.

- ***Sysfs:*** It is a special file system similar to proc. It doesn't exist on a device, and is a sort of gateway to the Linux kernel. It varies from proc in the way it is programmed as well as structured: sysfs is more structured and tailored towards computer-based parsing of the files and directories, whereas proc is more structured and tailored towards human-based reading/writing to the files and directories.

  The idea is that proc will eventually disappear and be fully replaced by the sysfs file system.

  Similar to /proc, sysfs is known to be a pseudo file system.

- ***Tmpfs:*** It is a temporary file system. The contents of this file system is stored in memory and not on a persistent disk. As such, its storage is generally very quick.

  In Linux, we use tmpfs for things like the device files in /dev and /tmp.

- ***Devpts:*** It is a pseudo file system similar to proc as well as sysfs. It includes device files used for terminal emulation. Earlier, those device files were created statically, which caused most distributions to allocate a lot of terminal emulation device files (as it is difficult to know how many of those emulations a user would start at any point in time). To manage those device files better, a pseudo file system is developed that creates and destroys the device files as they are needed.

- ***Usbfs:*** It is also a pseudo file system and we can compare it with devpts. It also includes files which are created or destroyed as USB devices are added or removed from the system. However, unlike devpts, it doesn't create device files, but pseudo files that can be used to interact with the USB device.

Task Compare and contrast tmpfs and sysfs.

Partitions and Disks

Every hardware device of the Linux system is represented by a device file inside the /dev location. Partitions and disks are no exception.

Example: As an example, let us take a serial ATA hard disk.

A SATA disk driver internally uses the SCSI layer to represent and access data. As such, a SATA device is represented as a SCSI device. The first SATA disk on your system is represented as /dev/sda, its first partition as /dev/sda1. You could read sda1 backwards as: "1st partition (1) on the first (a) scsi device (sd)".

Notes

```
~$ ls -l /dev/sda1
brw-rw---- 1 root disk 8,  1  Nov 12  10:10  /dev/sda1
```

A regular ATA disk (or DVD-ROM) would be represented by /dev/hda (hd stood for hard disk but is now seen as the identification of an ATA device).

```
$ ls -l /dev/hda
brw-rw---- 1 root cdrom 3, 0  Apr 23  21:00 /dev/hda
```

On a default installation, the device manager (which is called udev) creates the device files as it encounters the hardware. For instance, on my system, the partitions for my first SATA device can be listed as follows:

```
$ ls -l /dev/sda*
brw-r----- 1 root disk 8, 0 Sep 30 18:11 /dev/sda
brw-r----- 1 root disk 8, 1 Sep 30 18:11 /dev/sda1
brw-r----- 1 root disk 8, 2 Sep 30 18:11 /dev/sda2
brw-r----- 1 root disk 8, 5 Sep 30 18:11 /dev/sda5
brw-r----- 1 root disk 8, 6 Sep 30 18:11 /dev/sda6
brw-r----- 1 root disk 8, 7 Sep 30 18:11 /dev/sda7
brw-r----- 1 root disk 8, 8 Sep 30 18:11 /dev/sda8
```

The 'mount' Command and the fstab file

The mount command performs the act of mounting a medium to the file system. To perform well, it needs some information, such as the mount point, the file system type, the device and optionally some mounting options.

Example: The mount command to mount /dev/sda7, housing an ext3 file system, to /home, would be:

```
# mount -t ext3 /dev/sda7 /home
```

It can be seen that the act of mounting a file system is like "attaching" a certain storage somewhere on the file system, effectively expanding the file system with more files, directories and information.

Nevertheless, if there are various different partitions in your system, it would be a joke to have to enter the commands every time over and over again. This is one of the reasons why Linux has a file system definition file called /etc/fstab.

Caution The fstab file includes all the information mount could need in order to successfully mount a device.

An example fstab is shown below:

```
/dev/sda8  /          ext3    defaults,noatime      0 0
/dev/sda5  none       swap    sw                    0 0
/dev/sda6  /boot      ext2    noauto,noatime        0 0
/dev/sda7  /home      ext3    defaults,noatime      0 0
/dev/sdb1  /media/usb auto    user,noauto,gid=users 0 0
```

Notes

The file is structured as follows:

1. The device to mount (also supports labels).
2. The location to mount the device to (mount point).
3. The file system type, or auto if you want Linux to automatically detect the file system.
4. Additional options (use "defaults" if you don't want any specific option), such as noatime (don't register access times to the file system to improve performance) and users (allow regular users to mount/umount the device).
5. Dump-number (you can leave this at 0).
6. File check order (you can leave this at 0 as well).

The previous mount command example is not necessary any more (as the mount is performed automatically) but in case the mount has not been done already, the command is simplified to:

```
# mount /home
```

If you ever need to remove a medium from the file system, use the umount command:

```
# umount /home
```

This is of particular interest for removable media.

Example: If you want to access a CD or DVD (or even USB stick), you need to mount the media on the file system first before you can access it. Likewise, before you can remove the media from your system, you first need to unmount it:

```
# mount /media/dvd
```

(The DVD is now mounted and accessible)

```
# umount /media/dvd
```

(The DVD is now not available on the file system any more and can be removed from the tray)

Obviously, modern Linux operating systems have tools in place which automatically mount removable media on the file system and unmount it when they are removed. Linux does not offer such tool by default though.

Swap Location

Thgere can be a a partition dedicated for paging. Linux uses this partition when there is inadequate physical memory to keep all information regarding running processes (and their resources). When this is the case, the operating system will start putting information (which it hopes will not be used soon) on the disk, freeing up physical memory.

Instead of a file system usable by end users, the swap partition holds a particular file system for memory purposes and is recognized as a swap partition in the partition table:

```
# fdisk -l /dev/sda
Disk /dev/sda: 60.0 GB, 60011642880 bytes
255 heads, 63 sectors/track, 7296 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x8504eb57
   Device Boot  Start    End      Blocks   Id  System
/dev/sda1   *       1   1275    10241406   83  Linux
/dev/sda2        1276   7296    48363682+   5  Extended
```

Notes

```
/dev/sda5        1276   1525     2008093+  82  Linux swap / Solaris
/dev/sda6        1526   1532       56196   83  Linux
/dev/sda7        1533   2778    10008463+  83  Linux
/dev/sda8        2779   7296    36290803+  83  Linux
```

The swap partition is pointed by through the /etc/fstab file and enabled at boot-up.

In order to view the currently active swap partitions, view the content of the /proc/swaps file or run the swapon -s command:

cat /proc/swaps

Filename Type Size Used Priority

/dev/sda5 partition 2008084 0 -1

9.2.2 The Linux File System Locations

As we know, each location on the Linux file system has its distinct meaning. We've already discussed some of them without explicitly telling that those are standard locations, like /home which houses the local users' home directories.

System Required Locations

These locations cannot be placed on another file system medium. This is because those locations are needed by the mount command itself to function properly:

- /bin usually contains executable programs needed to bring the system up and running. Recently however, more and more distributions are moving all applications towards /usr /bin and are using symbolic links to transition towards this new structure.
- /etc contains all the configuration files for the system (not the user-specific configurations).
- /lib usually contains the system libraries necessary to successfully boot the system and run the commands which are located inside /bin. Recently however, these files are also being migrated towards /usr/lib.
- /sbin, just like /bin, contains executable programs. However, whereas /bin has programs which users can use as well, /sbin contains programs solely for system administrative purposes.

Userland Locations

These locations contain the files for the regular operation of a system. These can be stored on separate media if you want, but if you do, you will need to setup an initial ram disk to boot your system with. The location for the userland locations is/usr.

- /usr is the root of the userland locations (and usually the mount point of the separate medium).
- /usr/X11R6 contains all the files necessary for the graphical window server (X11); they are subdivided in binaries (bin/), libraries (lib/) and header definitions (/include) for programs relying on the X11 system.
- /usr/bin contains all the executable programs.
- /usr/lib contains all the libraries for the above mentioned programs.
- /usr/share contains all the application data for the various applications (such as graphical elements, documentation, etc.)

Notes

- /usr/local is often a separate mount as well, containing programs specific to the local system (the /usr might be shared across different systems in large environments).
- /usr/sbin is, like /usr/bin, a location for executable programs, but just like /bin and /sbin, /usr/sbin contains programs for system administrative purposes only.

General Locations

Everything else which might be placed on a separate medium are considered as general locations.

- /home contains the home directories of all the local users.
- /boot contains the static boot-related files, not actually necessary once the system is booted (for instance, it includes the bootloader configuration and kernel image).
- /media contains the mount points for the various detachable storage (like USB disks, DVDs, ...).
- /mnt is a location for temporarily mounted media (read: not worth the trouble of defining them in fstab).
- /opt contains add-on packages and is usually used to install applications into which are not provided by your package manager natively (as those should reside in /usr) or build specific to the local system (/usr/local).
- /tmp contains temporary files for the system tools. The location can be cleansed at boot up.
- /var contains data that changes in size, such as log files, caches, etc.

Special Kernel-provided File Systems

Some file system locations are not actually stored on a partition or disk, but are created and managed on-the-fly by the Linux kernel.

- /proc contains information about the running system, kernel and processes
- /sys contains information about the available hardware and kernel tasks
- /dev nowadays can be provided by the Linux kernel as well, offering a small set of default device
- nodes (so that the system can boot until udev is ready to take over control of /dev)

9.2.3 The Root File System /

As we know, the root file system / is considered as the parent of the whole file system. The root file system / is the first file system that is mounted when the kernel boots, and your system will not function appropriately if the kernel detects corruption on this file system. Also, due to the nature of the boot process, this file system will eventually become writeable (as the boot process needs to store its state information, etc.).

Some root file system locations are strongly advised to remain on the root file system. These locations are:

- /bin and /sbin as these contain the binaries (commands) or links to binaries that are needed to get a system up to the point it can mount other file systems. Although this functionality is gradually becoming less and less so, it would still break systems if you make separate mounts for these (small) locations.

Notes

- /lib as this contains the libraries that are needed by the commands in /bin.
- /etc as this contains the systems' configuration files, including those that are needed during the boot-up of the system.

Example: A main example of a configuration file inside /etc is fstab (which contains information about the other file systems to mount at boot time).

9.2.4 The Variable Data Location /var

The var location contains variable data. This location is expected to be used frequently throughout the life time of your installation. It contains log files, cache data, temporary files, etc.

This is the only reason to give /var its own separate file system. That is, by using a dedicated file system, it is ensured that flooding the /var location doesn't damage the root file system (as it is on a different file system).

9.2.5 The Userland Location /usr

This location includes the day-to-day application files of the system. A specific property of the location is that, if you are not updating your system, it should be left unmodified. In other words, you should be able to have only read-only access to the /usr location.

Thus, some larger installations use a network-mounted, read-only /usr location. Having /usr on a separate file system also has other benefits:

- If you are performing system administration tasks, you could unmount /usr so that end users don't run any programs they shouldn't during the administrative window.
- By placing /usr (and some other locations) on separate media, you keep your root file system small which lowers the chance of having a root file system corruption that will make booting impossible.
- You can use a file system that is optimized for fast reading (writing doesn't require specific response times).

9.2.6 The Home Location /home

This location includes the home directories of the end user. Inside these directories, these users have full write access. Outside these directories, users usually have read-only rights (or even no rights at all). The structure inside a home directory is also not bound to specific rules. In effect, the users' home directory is the users' sole responsibility.

This also signifies that users have the means of filling up their home location as they see fit, possibly flooding the root file system if /home isn't on a separate partition. For this reason, using a separate file system for /home is a good thing.

Another benefit of using a separate file system for /home is when you would decide to switch distributions. That is, your /home file system can be reused for other Linux distributions.

9.2.7 Permissions

By default, Linux supports a DAC (discretionary access control) permission system. In DAC permission system, privileges are based on the file ownership as well as user identity. However, projects exist that enable mandatory access control (MAC) on Linux, which bases privileges on roles and where the administrator can force security policies on files and processes.

Notes

As most MAC-based security projects (like RSBAC, LIDS and grSecurity) are not part of the default Linux kernel yet, we will talk about the standard, discretionary access control mechanism used by almost all Linux distributions.

Read, Write and Execute

The Linux file system provides support to numerous permission flags for every file or directory. A flag should be seen as a feature or privilege that is either enabled or disabled and is set independently of the other flags. The most used flags on a file system are the read (r), write (w) and execute (x) flags. Their meaning differs a bit based on the target.

Nevertheless, supporting these flags wouldn't make a system secure. That is, you want to mix these privileges based on who works with the file. For example, the system configuration files should only be writeable by the administrator(s); some might not even be readable by the users.

In order to enable this, Linux supports three different types of privilege destinations:

- the owner of the file (1st group of privileges)
- the group owner of the file (2nd group of privileges)
- everybody else (3rd group of privileges)

In this manner, you can place one set of privileges for the file owner, another set for the group (which signifies that everybody who is member of the group is matched against these privileges) and a third one set for everybody else.

In case of a file,

- the read privilege informs the system that the file can be read (viewed)
- the write privilege informs the system that the file can be written to (edited)
- the execute privilege informs the system that the file is a command which can be executed

For example, see the output of the ls -l command:

```
$ ls -l /etc/fstab
-rw-r--r-- 1 root root 905 Nov 21 09:10 /etc/fstab
```

In the above example, the fstab file is writeable by the root user (rw-) and readable by anyone else (r--).

In case of a directory,

- the read privilege informs the system that the directory's content can be viewed
- the write privilege informs the system that the directory's content can be changed (files or directories can be added or removed)
- the execute privilege informs the system that you are able to jump inside the directory (using the cd command)

Example: See the output of the ls -ld command:

$ ls -ld /etc/cron.daily

drwxr-x--- 2 root root 4096 Nov 26 18:17 /etc/cron.daily/

In the above example, the cron.daily directory is viewable (r), writeable (w) and "enterable" (x) by the root user.

Notes

Caution People in the root group have view- and enter rights (r-x). On the other hand, all other people have no rights to view, write or enter the directory (---).

In order to view the privileges on a file, the long listing format support of the ls command can be used.

Example: To view the permissions on the systems' passwd file (which contains the user account information):

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root root 3108 Dec 26 14:41 /etc/passwd
```

This file's permissions are read/write rights for the root user and read rights for everybody else.

The first character in the permission output shows the type of the file:

- '-': regular file
- 'd': a directory
- 'l': a symbolic link
- 'b': a block device (like /dev/sda1)
- 'c': a character device (like /dev/console)
- 'p': a named pipe
- 's': a unix domain socket

The remaining permission output is divided in three parts:

- one for the file owner,
- one for the file owning group, and
- one for all the rest.

So, in the given example, we can read the output '-rw-r--r--' as:

1. the file is a regular file.
2. the owner (root - see third field of the output) has read-write rights.
3. the members of the owning group (also root - see fourth field of the output) have read rights.
4. everybody else has read rights.

Example: Another example would be the privileges of the /var/log/sandbox directory. In this case, we also use ls' -d argument to make sure ls shows the information on the directory rather than its contents:

```
$ ls -ld /var/log/sandbox
drwxrwx--- 2 root portage 4096 Jul 14 18:47 /var/log/sandbox
```

In this case:

- the file is a directory
- the owner (root) has read, write and execute rights

Notes

- the members of the owning group (portage) also have read, write and execute rights
- everybody else can't do anything (no read, no execute and certainly no write rights)

Another method to obtain the access rights is to use the stat command.

Example:

```
$ stat /etc/passwd
  File: `/etc/passwd'
  Size: 3678        Blocks: 8          IO Block: 4096   regular file
Device: 808h/2056d  Inode: 3984335     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2013-03-18 21:46:06.000000000 +0100
Modify: 2013-03-18 21:46:06.000000000 +0100
Change: 2013-03-18 21:46:06.000000000 +0100
```

In the output of the stat command, you notice the same access flags as we identified before (-rw-r--r-- in this case), but also a number. This number identifies the same rights in a mort short-hand notation.

To be able to read the number, you need to know the values of each right:

- execute rights gets the number 1
- write rights gets the number 2
- read rights gets the number 4

To get the access rights of a particular group, add the numbers together.

For a file with privileges (-rw-r--r--), this gives the number 644:

- 6 = 4 + 2, meaning read and write rights for the owner
- 4 = 4, meaning read rights for the group
- 4 = 4, meaning read rights for everybody else

The first 0 that we notice in stats' output identifies the file as having no very specific privileges.

There are a few specific privileges inside Linux as well.

The restricted deletion flag, or sticky bit, has been identified before. When set on a directory, it prevents people with write access to the directory, but not to the file, to delete the file (by default, write access to a directory means that you can delete files inside that directory regardless of their ownership). The most well-known use for this flag is for the /tmp location:

Example:

```
$ stat /tmp
  File: `/tmp'
  Size: 28672       Blocks: 56         IO Block: 4096   directory
Device: 808h/2056d Inode: 3096577     Links: 759
Access: (1777/drwxrwxrwt)  Uid: ( 0/    root)   Gid: (    0/    root)
Access: 2010-01-10 17:44:04.000000000 +0100
Modify: 2013-04-02 00:04:36.000000000 +0200
Change: 2013-04-02 00:04:36.000000000 +0200
```

Notes

The setuid or setgid flag is another specific privilege that is identified before. When set on an executable (non-script!), the executable is executed with the rights of the owner (setuid) or owning group (setgid) instead of with the rights of the person that is executing it. That does mean that people with no root privileges can still execute commands with root privileges if those commands have the setgid flag set. For this reason, the number of executables with the setuid/setgid bit set need to be limited and well audited for possible security exposures.

Example:

/bin/mount is a nice example of this flag:

```
$ stat /bin/mount
  File: `/bin/mount'
  Size: 59688      Blocks: 128        IO Block: 4096   regular file
Device: 808h/2056d Inode: 262481      Links: 1
Access: (4711/-rws--x--x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2010-02-06 13:50:35.000000000 +0100
Modify: 2013-01-02 13:50:35.000000000 +0100
Change: 2013-01-02 13:50:43.000000000 +0100
```

In order to change the privileges of a file or directory, it is required to use the chmod command (change mode). Its syntax is easy enough to remember well. First, the target permissions:

- 'u' for user,
- 'g' for group, and
- 'o' for everybody else (others).

Then you can set (=), add (+) or remove (-) privileges.

Example: To make /etc/passwd writeable for the members of the owning group:

```
# chmod g+w /etc/passwd
```

Privileges can also be combined.

Example: If you want to remove write privileges for the owning group and remove read privileges for the others:

```
# chmod g-w,o-r /etc/passwd
```

Finally, the numeric notation can also be used required:

Example:

```
# chmod 644 /etc/passwd
```

When you need to change the ownership of a file or directory, use the chown (change owner) or chgrp (change group) command.

Example: To change the owner of a file to the user "jack":

```
# chown jack template.txt
```

If you want to change the owner of a file, you need to be root. It will not help if you are the current owner. This is not true for the group though: if you are a member of the target group, you can change the owning group.

Notes

Example:

```
$ ls -l bar
-rw-r--r-- 1 swift  users   0 May 13 20:41 bar
$ chgrp dialout bar
$ ls -l bar
-rw-r--r-- 1 swift  dialout 0 May 13 20:41 bar
```

If you want to change the owner and group, a single chown command can be used. That is, just separate the target owner and group with a colon.

Example:

```
# chown jack:dialout template.txt
```

9.2.8 Attributes

Some file systems provides permission to add more attributes to files. These attributes might have impact on the permissions/usage of these files, or on how the operating system works with these files. Not many distributions use these attributes, because not all file systems support them.

Listing and Modifying Attributes

In order to view the attributes of a file, the lsattr command (list attributes) is used; to modify the attributes, chattr (change attributes) command is used. Now let us create an example file:

```
# touch /tmp/foo
# chattr +asS /tmp/foo
```

Now let's see what lsattr has to say:

```
# lsattr /tmp/foo
s-S--a---------  /tmp/foo
```

Not a big surprise, given the chattr command before. But what does it mean? Well, man chattr gives us the information we need, but here is it in short-hand:

- **s:** when the file is deleted, its blocks are zeroed and written back to disk (unlike regular files where only the reference to the file is deleted).
- **S:** when changes are made to the file, the changes are immediately synchronized to disk (no memory caching allowed).
- **a:** the file can only be appended (data is added to the file); changes are not allowed to existing content. Very useful for log files.

Did u know? Another very interesting attribute is the immutable flag (i) that doesn't allow the file to be deleted, changed, modified, renamed or moved.

Self Assessment

Fill in the blanks:

4. The ..... command by itself, without any arguments, shows you a list of mounted file systems.

Notes

5. ..... is a special file system which doesn't exist on a device, but is a sort of gateway to the Linux kernel.
6. ..... is a pseudo file system which contains device files used for terminal emulation.
7. A SATA disk driver internally uses the ..... layer to represent and access data.
8. If you ever need to remove a medium from the file system, use the..... command.
9. The ..... file system is the parent of the entire file system.

9.3 File Search Utilities

Mostly, the file you want to locate is inside your home directory. However, sometimes you want to locate a particular file somewhere on your whole system. Fortunately, there are a few commands at your disposal to do so.

9.3.1 Locate Command

This command is used to manage and use a database of files. This will assist you to find a particular file. Before using this command, you first need to install it (the package is called sys-apps/mlocate) and then create the file database. Also, this database is not automatically brought up to date while you modify your system, so you'll need to run this command (which is the same for creating a new database or updating an existing one) every now and then:

```
# updated
```

We can keep this database up to date by using the system scheduler (known as cron). When your database is build and somewhat up to date, you can locate any particular file on your filesystem using locate.

Example:

```
# locate make.conf
/etc/portage/make.conf
(...)
/usr/portage/local/layman/make.conf
```

It can be seen that the locate command returns all files it has found where the string (in this case, "make.conf") is used in the filename, even when the file name is different.

Notes The name mlocate is the name of the project that maintains the package. In earlier days, the package of choice for the locate functionality was slocate.

9.3.2 Find Command

This command is considered as a very significant and powerful command which, unlike locate,. only returns live information. Thus, a database is not used. This makes searches with find somewhat slow, but find's power isn't speed, but the options you can give to find a particular file.

Notes

Regular Find Patterns

The most simple find construct is to locate a particular file inside one or more directories.

Example: To find files or directories inside /etc whose name is dhcpd.conf (exact matches):

```
$ find /etc -name dhcpd.conf
/etc/dhcp/dhcpd.conf
```

To find files (not directories) where dhcpd is in the filename, also inside /etc directory:

```
$ find /etc -type f -name '*dhcpd*'
/etc/conf.d/dhcpd
/etc/init.d/dhcpd
/etc/udhcpd.conf
/etc/dhcp/dhcpd.conf
```

To find files in the /etc directory who have been modified within the last 7 days (read: "less than 7 days ago"):

```
$ find /etc -type f -mtime -7
/etc/mtab
/etc/adjtime
/etc/wifi-radar.conf
/etc/genkernel.conf
```

You can even find files based on their ownership.

For example, find the files in /etc that do not belong to the root user:

```
$ find /etc -type f -not -user root
```

Combining Find Patterns

You can also combine find patterns.

Example: Find files modified within the last 7 days but whose name does not contain .conf:

```
$ find /etc -type f -mtime -7 -not -name '*.conf'
/etc/mtab
/etc/adjtime
```

Or, find the same files, but the name should also not be mtab:

```
$ find /etc -type f -mtime -7 -not \( -name '*.conf' -or -name mtab)
/etc/adjtime
```

Working with the results

With find, you can also perform tasks on the results.

Example: To view the "ls -l" output against the files that find finds, you can add the -exec option. The string after -exec should contain two special character sequences:

Notes

A character sequence '{}' represents the file found by the find command. The command given to the -exec option is executed and '{}' is substituted with the filename.

A character sequence \; ends the command in the -exec clause.

```
$ find /etc -type f -mtime -7 -exec ls -l '{}' \;
```

On the Internet, you'll also find the following construction:

```
$ find /etc -type f -mtime -7 | xargs ls -l '{}'
```

This gives the same result, however its behaviour is slightly different.

When we use -exec, the find command executes the command for each file it encounters. The xargs construction will attempt to execute the command as little as possible, based on the argument limits.

Example: If the find command returns 10000 files, the command given to -exec is executed 10000 times, once for every file.

With xargs, the command might be executed only a few dozen times. This is possible because xargs appends multiple files for a single command as it assumes that the command given can cope with multiple files.

Example: Run for find -exec:

```
ls -l file1
ls -l file2
...
ls -l file10000
```

Example run for xargs:

```
ls -l file1 file2 ... file4210
ls -l file4211 file4212 ... file9172
ls -l file9173 file9174 ... file10000
```

9.3.3 GNOME

GNOME Search Tool is a utility which is used for finding files on your system. To do a basic search, type a filename or a partial filename, with or without wildcards.

Basically, the Gnome Search Tool is a GUI version of locate and find. When doing a basic search, the locate command is used first by default. Then the slower but more thorough find command is used.

This program can be started by typing the following command at shell prompt:

```
$ gnome-search-tool &
```

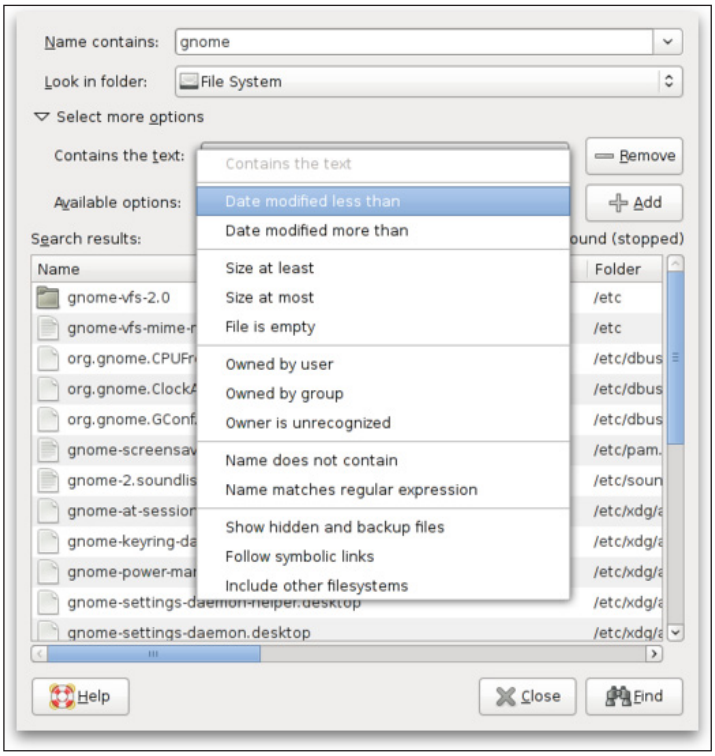
On the other hand, you can find the Gnome Search Tool by going to Places > Search For Files

The case sensitivity of the search relies on your operating system.

Example: On Linux, the find, grep, and locate commands support the -i option, so all searches are case-insensitive.

Notes

Figure 9.2: Gnome Search Tool



Source: http://www.nixtutor.com/linux/finding-files-in-linux/

Self Assessment

Fill in the blanks:

10. The ..... command manages and uses a database of files to help you find a particular file.
11. The ..... command is a very powerful command which only returns live information.
12. When using ....., the find command executes the command for every file it encounters.
13. The Gnome Search Tool is basically a ..... version of locate and find.

9.4 Navigating File System

Even though GUI file managers like Konqueror (in KDE) or Nautilus (in GNOME) are easy to use, they can be used only in the presence of working GUI desktop. Sometimes, you may not have a graphical environment to run a graphical file manager.

Example: You may be logged in through a text terminal, or X may not be working on your system. In those situations, you have to rely on Linux commands to work with files and directories. You can always use Linux commands, even in the graphical environment — all you have to do is open a terminal window and type the Linux commands.

Now we will describe some Linux commands for moving around the Linux file system.

Notes

9.4.1 Commands for Directory Navigation

In case of Linux, when you log in as root, then /root is your home directory. For other users, the home directory is usually in the /home directory. My home directory (when we log in as edulaney) is /home/edulaney. This information is stored in the /etc/passwd file.

By default, only you are allowed to save files in your home directory, and only you can create subdirectories in your home directory in order to further organise your files.

The concept of a current directory is supported by Linux. Current directory is the directory on which every file and directory command operate.

Example: After you log in, your current directory is the home directory.

To see the current directory, type the pwd command.

To change the current directory, use the cd command.

Example: To change the current directory to /usr/lib, type the following:

```
cd /usr/lib
```

Then, to change the directory to the cups subdirectory in /usr/lib, type this command:

```
cd cups
```

Now, on using pwd command shows /usr/lib/cups as the current directory.

Thus, you can refer to a directory's name in two methods:

- ***Absolute pathname:*** An example of absolute pathname is /usr/lib, which is an exact directory in the directory tree. Absolute pathname is considered as the complete mailing address for a package that the postal service will deliver to your next-door neighbor.
- ***Relative directory name:*** An example of relative directory name is cups. It represents the cups subdirectory of the current directory, whatever that may be. The relative directory name gives the postal carrier directions from your house to the one next door so the carrier can deliver the package.

Typing cd cups in /usr/lib will change the current directory to /usr/lib/cups. However, if we type the same command in /home/edulaney, the shell tries to change the current directory to /home/edulaney/cups.

In order to change the current directory back to your home directory, use the cd command without any arguments. No matter where you are, typing cd at the shell prompt brings you back home!

Now let discuss he tilde character (~). It refers to your home directory. So, the current directory can be changed to your home directory by using the command cd ~. You can also refer to another user's home directory. This is done by appending the name of that user to the tilde. For instance, cd ~superman changes the current directory to the home directory of superman.

Now we will discuss the concept of a single dot (.) and two dots (. .) They also have special meanings. A single dot (.) specifies the current directory, whereas two dots (. .) specifies the parent directory.

For example, if the current directory is /usr/share, you go one level up to /usr by typing the following:

```
cd . .
```

Notes

Task Explain tilde character (~) with example.

Self Assessment

Fill in the blanks:

14. To change the ..... directory, use the cd command.
15. The tilde character (~) refers to your ..... directory.

9.5 Summary

- A file system is nothing more than the way the computer stores and retrieves all your files. These files include your documents, programs, help files, games, music etc.
- The act of mounting requires that you identify a location of the file system as being a mount point under which every file is actually stored on a different location.
- The system required locations are locations you cannot place on another file system medium because those locations are required by the mount command itself to function properly.
- Userland locations are the locations which contain the files for the regular operation of a system.
- The root file system / is the first file system that is mounted when the kernel boots, and your system will not function properly if the kernel detects corruption on this file system.
- The var location contains variable data. You should expect this location to be used frequently during the life time of your installation.
- The usr location contains the systems' day-to-day application files.
- The/home location contains the end users' home directories.
- Linux supports what is called a discretionary access control (DAC) permission system where privileges are based on the file ownership and user identity.
- With all the locations, it might be difficult to locate a particular file. There are a few commands which are used to search a file.

9.6 Keywords

/home location: The /home location contains the end users' home directories.

Find: The find command is a powerful command which only returns live information.

GNOME Search Tool: GNOME Search Tool is a utility for finding files on your system.

Locate: The locate command manages and uses a database of files to help you find a particular file.

Mount: The act of mounting a medium to the file system is performed by the mount command.

Root File System /: The root file system / is the first file system that is mounted when the kernel boots, and your system will not function properly if the kernel detects corruption on this file system.

Notes

Userland Locations: Userland locations are the locations which contain the files for the regular operation of a system.

Var Location: The var location contains variable data.

9.7 Review Questions

1. Discuss the concept of Mounting File Systems.
2. Describe how SATA disk driver represent and access data.
3. Explain the use of mount command with example.
4. Discuss the standard locations of Linux File system.
5. Discuss the advantages of having /usr on a separate file system.
6. Explain the method used to obtain the access rights.
7. Illustrate the concept of locate command with example.
8. Make distinction between locate command and find command.
9. How to use GNOME Search Tool for finding files on your system? Illustrate.
10. Discuss the commands used for directory navigation.

Answers: Self Assessment

1.	Files	2.	Folder
3.	hierarchical	4.	Mount
5.	proc	6.	Devpts
7.	SCSI	8.	umount
9.	root	10.	locate
11.	find	12.	-exec
13.	GUI	14.	Current
15.	home		

9.8 Further Readings

Books

Christopher Negus, *Linux Bible*, Wiley.

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.

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

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

Notes

Online links

http://tldp.org/LDP/intro-linux/html/sect_03_01.html

http://tldp.org/LDP/tlk/fs/filesystem.html

http://www.cyberciti.biz/tips/understanding-unixlinux-file-system-part-i.html

http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/Linux-Filesystem-Hierarchy.pdf

Notes

Unit 10: Servers

CONTENTS

Objectives

Introduction

10.1 DNS Server

10.1.1 Configuring DNS Server

10.1.2 Downloading and Installing the BIND Packages

10.2 FTP Server

10.2.1 Installing and Configuring FTP Server Using vsftpd

10.3 Apache Web Server

10.3.1 Installing, and Running Apache

10.4 DHCP Server

10.4.1 How to Get DHCP Started

10.4.2 Installing DHCP

10.4.3 Configuring DHCP

10.5 Summary

10.6 Keywords

10.7 Review Questions

10.8 Further Readings

Objectives

After studying this unit, you will be able to:

- Discuss the concept of DNS server
- Explain FTP server
- Describe the concept of Apache web server
- Explain the process of installing and configuring DHCP servers

Introduction

By the commencement of the 90s home PCs were lastly powerful enough to run a full blown UNIX. Linus Torvalds, a young man studying computer science at the University of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and punctually started to code. From the start, it was Linus' goal to have a free system that was completely compliant with the original UNIX. Samba is a strong network service for file and print sharing that works on the mainstream of operating systems available today. Vsftpd is a GPL approved FTP server for UNIX systems, including Linux. It is secure and extremely fast. It is stable. Below, we will see evidence supporting all three assertions. Apache is the most extensively

Notes

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. Dynamic Host Configuration Protocol (DHCP) repeatedly assigns IP addresses and other network configuration information (subnetmask, broadcast address, etc.) to computers on a network. DNS acts as a directory service for all of these systems and let you to specify each one by its hostname.

10.1 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).

10.1.1 Configuring DNS Server

Domain Name System (DNS) converts the name of a Web site.

Example: Domain name system converts (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, not the Web site's name, is 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.

Notes

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.

When to Use a DNS Caching Name Server

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. Setting up a caching DNS server is fairly straightforward and works whether or not your ISP provides you with a static or dynamic Internet IP address.

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.

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.

Notes

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.

Task Illustrate how to use Dynamic DNS in your own domain.

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 determining 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 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
  ```

Notes

```
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 :\>
```

10.1.2 Downloading and Installing the BIND Packages

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`.

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 10.1: 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 server name 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]#
```

Notes

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.

Table 10.2: 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:

- Forward zone file definitions list files to map domains to IP addresses.
- 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.

Fortunately BIND comes with samples of all the primary files you need. Table 10.3 explains their names and purpose in more detail.

Table 10.3: 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.

Example: 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.

Example: In this example, we allow queries on any interface.

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

Example: 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.

Task While it is not required, it is a good practice to configure your DNS server's named.conf file to support BIND views.

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.

Notes

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:
   - localhost: Refers to the DNS server itself.
   - localnets: Refers to all the networks to which the DNS server is directly connected.
   - 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 we use for our home network. All the statements below were inserted after the options and controls sections in the file. We 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.

   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.

Example: 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; };
```

Notes

```
 file "my-site.zone";
};
```

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";

};
```

Self Assessment

Fill in the blanks:

1. ..... converts the name of a Web site.
2. ..... is a group that sustains the DNS-related software suite that runs under Linux.
3. A ..... DNS name server is used as a reference only.
4. ..... refers to all the networks to which the DNS server is directly connected.

10.2 FTP Server

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 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:

- Anonymous
- 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.

Caution 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.

Notes

10.2.1 Installing and Configuring FTP Server using vsftpd

In this section, we will discuss the concept of vsftpd – FTP Server Installation and configuration.

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

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.

Did u know? vsftpd supports IPv6, SSL, locking users to their home directories and many other advanced features.

vsftpd Features

VSFTPD offer security, performance and constancy over other servers. A quick list of vsftpd features:

- Virtual IP configurations
- Virtual users

- Run as standalone or inetd / xinetd operation
- Per-user configuration
- Bandwidth throttling
- Per-source-IP configurability
- Per-source-IP limits
- IPv6 ready
- Encryption support through SSL integration
- And much more.

Self Assessment

Fill in the blanks:

5. ..... is a TCP protocol for uploading and downloading files between computers.
6. FTP ..... constantly listens for FTP requests from remote clients.

10.3 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.

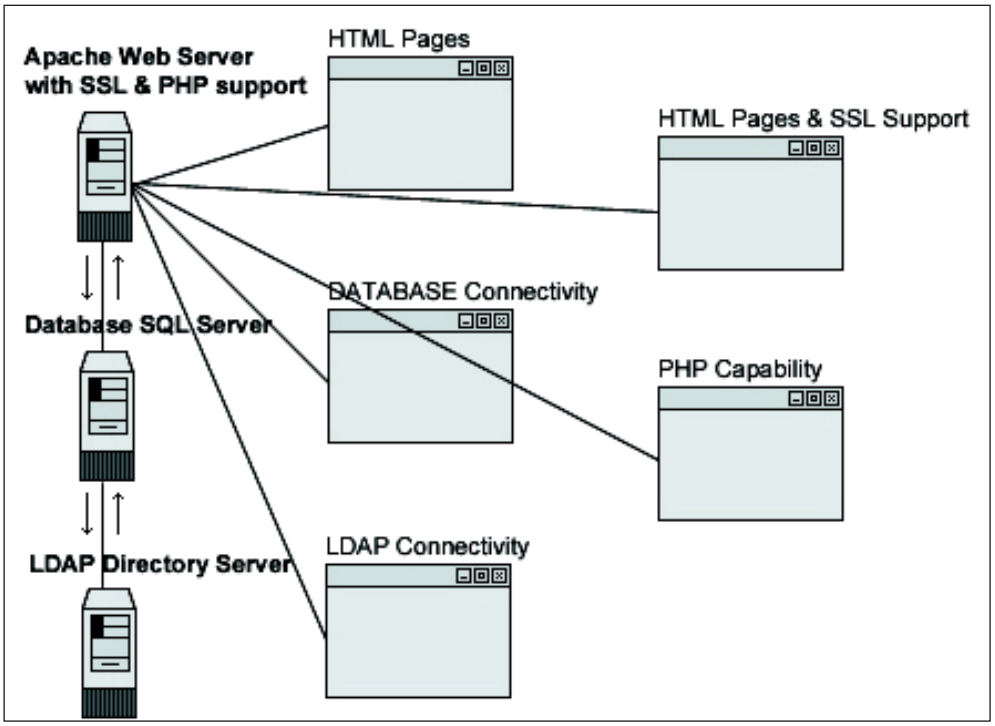
Finally, it's relatively fast. We say "relatively" as it's relative to what you're doing with it. If you're hosting a Web site with mostly static content, Apache is a fireball. If of desktop-centric Linux such as Caldera, it's rare that you won't have the Apache server installed. During your Linux installation, if you see an option for "Web services" makes sure to select it so that Apache will be installed. If you're new to the Unix/Linux/BSD world, we should warn you about something. There are two types of installation packages – source and binary.

Notes 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.

Notes

Figure 10.1: Apache Web Server



Some features of Apache include:

- ***Powerful:*** Apache's performance and reliability is well-known.
- ***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 will be talking about some of these features in future articles.
- ***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.
- ***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!
- ***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.

Task Analyze the difference between web server and apache web server.

10.3.1 Installing, and Running 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

Notes

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.

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!

These Installation Instructions Assume:

- Commands are Unix-compatible.
- The source path is /var/tmp, *other paths are possible*.
- Installations were tested on Red Hat Linux 6.1 and 6.2.
- All steps in the installation will happen in super-user account root.
- Apache version number is 1.3.12.
- Mod_SSL version number is 2.6.4-1.3.12.
- Mod_Perl version number is 1.24.
- Mod_PHP version number is 4.0.0.

Table 10.4 enlists the required action points for installing the Apache Web server.

Table 10.4: 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.

- OpenSSL should be previously installed on your system if you want Apache and SSL encryption support.

Notes

- PosgreSQL supposed to be already installed on your system if you want Apache and PostgreSQL database connectivity support.
- MM should be already installed on your system if you want Apache and MM high-performance RAM-based session cache support.
- OpenLDAP should be already installed on your system if you want Apache and LDAP directory connectivity support.
- 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.

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
```

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 we 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

Build Apache

As with any source installation, you'll then require to build the installation:

```
make
make install
```

Notes

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.

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.

Self Assessment

Fill in the blanks:

7. ..... option specifies the directory where the Apache files will be installed.
8. The module ..... is used to map different parts of the URL tree.
9. The module ..... is used to allow for strong cryptography using SSL.

10.4 DHCP Server

The Dynamic Host Configuration Protocol (DHCP) is nothing new to many Linux users. The majority of people have a DHCP client installed on their computers so they can connect to the Internet via cable or DSL modems. This allows them to have a dynamically assigned IP address every time they connect to their ISP, usually just by booting the system if they have a dedicated cable or DSL connection. This is an easy way for ISPs to hand out semi-permanent IP addresses to their clients without giving everyone a real static IP address. In fact, if you leave your computer on most of the time, you may end up with the same IP address for a very long time since it works on an IP address lease/renewal basis.

However, DHCP is far more adaptable than this usage might imply. It can be used in any corporate environment where laptops come and go and computers are turned on and off or are changed around on a semi-regular basis. For most system administrators, dealing with network changes, IP address changes, and so forth, is one of their most time-consuming tasks. Fortunately, DHCP allows most system administrators to deal with a large networked environment with a greater degree of ease.

DHCP was designed to give all possible TCP/IP configuration parameters to client computers using the client/server model. Because it includes every configuration option defined in the Requirements for Internet Hosts RFC, there is no need for a system administrator to configure TCP/IP on a user's desktop. It is all done by the interaction between the DHCP client and the DHCP server.

Usually if you have a cable modem or DSL, you get your home PC's IP address dynamically assigned from your service provider. If you install a home cable/DSL router between your

Notes

modem and home network, your PC will most likely get its IP address at boot time from the home router instead.

Did u know? You can choose to disable the DHCP server feature on your home router and set up a Linux box as the DHCP server.

10.4.1 How to Get DHCP Started

To get DHCP started:

1. Some older Fedora/RedHat versions of the DHCP server will fail unless there is an existing dhcpd.leases file. Use the command touch /var/lib/dhcp/dhcpd.leases to create the file if it does not exist.

   ```
   [root@bigboy tmp]# touch /var/lib/dhcp/dhcpd.leases
   ```

2. Use the chkconfig command to get DHCP configured to start at boot:

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

   With Debian/Ubuntu the equivalent command for the dhcp3-server package would be:

   ```
   root@u-bigboy:/tmp# sysv-rc-conf dhcp3-server on
   ```

3. Use the service command to instruct the /etc/init.d/dhcpd script to start/stop/restart DHCP after booting

   ```
   [root@bigboy tmp]# service dhcpd start
   [root@bigboy tmp]# service dhcpd stop
   [root@bigboy tmp]# service dhcpd restart
   ```

   With Debian / Ubuntu the equivalent commands would be:

   ```
   root@u-bigboy:/tmp# /etc/init.d/dhcp*-server start
   root@u-bigboy:/tmp# /etc/init.d/dhcp*-server stop
   root@u-bigboy:/tmp# /etc/init.d/dhcp*-server restart
   ```

4. Keep in mind to restart the DHCP process every time you make a change to the conf file for the changes to take effect on the running process. You also can test whether the DHCP process is running with the following command; you should get a response of plain old process ID numbers:

   ```
   [root@bigboy tmp]# pgrep dhcpd
   ```

5. Lastly, always remember to set your PC to get its IP address via DHCP.

10.4.2 Installing DHCP

Once you have the source tarball or the RPM/DEB package for your distribution, install it. For RPM/DEB package users, this process is easy and straightforward. To install from the source code, untar the source code and enter the dhcp_2.0 subdirectory, then type:./configure make install.

If you have a earlier version of DHCP installed on your system, be sure to remove it prior to running make install so that you do not mix man pages, configuration files, or binaries.

Download and Install the DHCP Package

Most RedHat and Fedora Linux software products are obtainable in the RPM format. Downloading and installing RPMs aren't hard. When searching for the file, remember that the DHCP server

Notes

RPM's filename usually starts with the word dhcp followed by a version number like this: dhcp-3.0.1rc14-1.i386.rpm.

With Debian / Ubuntu the package name may contain a version number. Use the `dpkg --list | grep dhcp` command to get a list of all your dhcp packages and use the output to infer what the DHCP server package name would be. In this case we can guess that the package name should be `dhcp3-server`.

```
root@u-bigboy:/tmp# dpkg --list | grep dhcp
ii dhcp3-client 3.0.3-6ubuntu7 DHCP Client
ii dhcp3-common 3.0.3-6ubuntu7 Files used by all the dhcp3* packages
root@u-bigboy:/tmp#
```

The /etc/dhcpd.conf File

When DHCP starts, it reads the file /etc/dhcpd.conf. It utilizes the commands here to configure your network. The standard DHCP RPM package doesn't automatically install a /etc/dhcpd.conf file, but you can find a sample copy of dhcpd.conf in the following directory which you can always use as a guide.

```
/usr/share/doc/dhcp-<version-number>/dhcpd.conf.sample
```

You have to copy the trial dhcpd.conf file to the /etc directory and then you'll have to edit it. Here is the command to do the copying for the version 3.0p11 RPM file:

```
[root@bigboy tmp]# cp /usr/share/doc/dhcp-3.0pl1/dhcpd.conf.sample
/etc/dhcpd.conf
```

With Debian / Ubuntu the configuration file name is /etc/dhcp*/dhcpd.conf and has the similar syntax as that used by Redhat / Fedora.

Here is a rapid explanation of the dhcpd.conf file: Most importantly, there must be a subnet section for each interface on your Linux box.

```
ddns-update-style interim
ignore client-updates
subnet 192.168.1.0 netmask 255.255.255.0 {
 # The range of IP addresses the server
 # will issue to DHCP enabled PC clients
 # booting up on the network
 range 192.168.1.201 192.168.1.220;
 # Set the amount of time in seconds that
 # a client may keep the IP address
 default-lease-time 86400;
 max-lease-time 86400;
 # Set the default gateway to be used by
 # the PC clients
 option routers 192.168.1.1;
 # Don't forward DHCP requests from this
 # NIC interface to any other NIC
 # interfaces
```

Notes

```
 option ip-forwarding off;
 # Set the broadcast address and subnet mask
 # to be used by the DHCP clients
 option broadcast-address 192.168.1.255;
 option subnet-mask 255.255.255.0;
 # Set the NTP server to be used by the
 # DHCP clients
 option ntp-servers 192.168.1.100;
 # Set the DNS server to be used by the
 # DHCP clients
 option domain-name-servers 192.168.1.100;
 # If you specify a WINS server for your Windows clients,
 # you need to include the following option in the dhcpd.conf file:
 option netbios-name-servers 192.168.1.100;
 # You can also assign specific IP addresses based on the clients'
 # ethernet MAC address as follows (Host's name is "laser-printer":
 host laser-printer {
 hardware ethernet 08:00:2b:4c:59:23;
 fixed-address 192.168.1.222;
 }
}
#
# List an unused interface here
#
subnet 192.168.2.0 netmask 255.255.255.0 {
}
```

There are many more options statements you can make use to configure DHCP. These include telling the DHCP clients where to go for services such as finger and IRC. Check the dhcp-options man page after you do your install:

```
[root@bigboy tmp]# man dhcp-options
```

The host statement seen in the sample dhcpd.conf file can be very valuable. Some devices such as network printers default to getting their IP addresses using DHCP, but users need to access them by a fixed IP address to print their documents. This statement can be used to always provide specific IP address to DHCP queries from a predefined a NIC MAC address. This can help to reduce systems administration overhead.

10.4.3 Configuring DHCP

The configuration file for your DHCP server is the /etc/dhcpd.conf file. This file has numerous configuration commands that operate the server and provide configuration information to the clients.

A DHCP server can supply service to individual hosts through static address assignments (useful for servers) or to an entire subnet of hosts through dynamic address assignments. The configuration file uses host and subnet statements that identify the client systems.

Notes

Example: Let us consider a statement: host ns `{ hardware ethernet 12:34:56:78:AB:CD; fixed_address 192.168.1.5;}`

This statement defines the host name, Ethernet address, and IP address of the client. Using this statement, any time the client with the matching Ethernet address connects to the server, the server will return the defined host name and the defined static IP address.

Example: Let us consider a subnet statement: `subnet 192.168.1.0 netmask 255.255.255.0 {range 192.168.1.10 192.168.1.200;}`

This statement describes the network we are providing DHCP service for, in this case 192.168.1.0. It also says that the IP addresses it is allowed to lease are from 192.168.1.10 to 192.168.1.200.

The range clause defines the low and high values of the IP addresses the DHCP server is allowed to lease.

Caution Range clause must always be used within a subnet statement, and the range defined must be within the address space of the defined subnet.

Usually if you have a cable modem or DSL, you get your home PC's IP address dynamically assigned from your service provider. If you install a home cable/DSL router between your modem and home network, your PC will most likely get its IP address at boot time from the home router instead. You can choose to disable the DHCP server feature on your home router and set up a Linux box as the DHCP server.

Configuration Problems

There are a few uncertain blocks you may encounter after your DHCP server is configured and running. Primarily, if you provide DHCP services to Microsoft Windows clients, you may encounter problems with the limited broadcast address. If Windows clients do not see DHCP messages from the server while other clients, such as Linux clients, do, you may need to define a specific route for the limited broadcast address on your Linux server. To do so, add the following to your /etc/hosts file:

```
255.255.255.255 lim_broad
```

Then, add a route for the limited broadcast address by using this command:

```
route add _host lim_broad dev eth0
```

To make this a more permanent change, add the preceding command to your /etc/rc.d/rc.local file. For Red Hat and Caldera users, the above commands have been added for you in the stock DHCP init script, /etc/rc.d/init.d/dhcpd.

If you have problems running dhcpd on numerous network interfaces, you most likely have an older kernel. Be sure you are using a kernel version of 2.0.31 or higher. If you get a Protocol Not Configured error, be sure that your kernel was compiled with the CONFIG_PACKET and CONF_FILTER options turned on.

DHCP Configuration Options

There are a lot of configuration options available to you in the /etc/dhcpd.conf file. All of these options can be specified within a host or subnet statement or within a group statement. The group statement can be used to supply configuration options to a group of host or subnet statements. Also, you can use options outside of group, subnet, or host statements if you wish to apply those options to every single system and network you define. Because of this flexibility, you can completely tailor your configuration to your specific needs.

Notes

The initial statements you should be concerned with are the allow and deny statements. These statements control how dhcpd handles client requests. The three allow/deny statements are as follows:

- **booting:** This statement is used within a host statement to allow or deny the exact host from obtaining any configuration information from the DHCP server. By default, hosts are allowed booting.
- **bootp:** This statement is used to inform the server whether or not it should act as both a BootP and DHCP server or just a DHCP server. By default, BootP clients are allowed, so you need only one server to handle both DHCP and BootP requests.
- **unknown clients:** This statement controls whether dhcpd will allow or deny clients for which it does not have detailed host entries. By default, this is allowed, as denying this feature takes away about 90 percent of the reason why anyone would want to use a DHCP server.

Example: To deny booting, you would use:

```
deny booting;
```

There are also many additional configuration options that control the server and the DHCP protocol. The more commonly used options are as follows:

- boot_unknown_clients [true|false]: If the value is false, simply clients that have a host statement are assigned an address. The default is true.
- default_lease_time [seconds]: This option describe the length of time, in seconds, for an IP address lease if the client does not request a specific lease length.
- dynamic_bootp_lease_cutoff [date]: This option defines a termination date for addresses assigned to BootP clients. By default, BootP clients are assigned a permanent address.
- dynamic_bootp_lease_length [seconds]: This option defines the length of time in seconds for an IP address lease for BootP clients. Note, though, that BootP clients do not renew address leases, so a client that does not boot and contact the server often enough will lose its lease.
- fixed_address [address[,address...]]: This option assigns a permanent IP address to a host as part of a host statement. More than one address can be assigned for a client that boots on more than one subnet.
- get_lease_hostnames [true|false]: If the value is true, dhcpd will execute a reverse lookup for every dynamically assigned address and send to the client the host name it gets from DNS. However, this can add a lot of extra overhead for servers on larger networks. By default, this value is false, and no reverse lookups are done.
- hardware ethernet [address]: This option defines the client's Ethernet address within a host statement. The DHCP server uses the Ethernet address to map host information to a specific client. For BootP clients, this is the only way dhcpd can map the information; however, DHCP clients can use other values in addition to the Ethernet address to identify themselves to the server. To obtain the Ethernet address of a Linux client, run ifconfig and check the Hwaddr field.
- max_lease_time [seconds]: This option defines the maximum length, in seconds, of a lease length. This is the maximum lease length a client may receive regardless of what it requests.
- range [dynamic_bootp] [low address] [high address]: This option defines the range of IP addresses available for the server to dynamically assign. The argument [dynamic_bootp] tells dhcpd to assign dynamic IP addresses to BootP clients as well as DHCP clients. By default, BootP clients are not assigned dynamic IP addresses because BootP was not designed for it.

Notes

- server_identifier [address]: This option defines the IP address of the server that is sent to clients. By default, the address of the server's network interface is used, so this should be used only if the server presents an incorrect address for some reason.
- server_name ["name"]: This option defines the host name of the server.
- use_host_dec1_name [true|false: This option defines whether the server will send the name provided on the host statement to the client as its host name.
- use_lease_addr_for_default_route [true|false]: This option sends the client its own address as the default route instead of sending the true default route.

The previous statements deal with how the server operates; however, there are more configuration options available for client operation. Some of the more common options are as follows:

- option broadcast_address [address]: Defines the broadcast address.
- option domain_name ["domain"]: Defines the domain name.
- option domain_name_servers [address_list]: Lists the addresses of the DNS name servers.
- option finger_server [address_list]: Lists the finger servers available to the client. Finger servers are typically used on sites that block finger traffic at the firewall.
- option host_name ["name"]: Defines the client's host name.
- option nis_domain ["name"]: Defines the name of the local NIS (Network Information Services) domain.
- option nis_servers [address_list]: Lists the addresses of the NIS servers.
- option nntp_server [address_list]: Lists the addresses of the NNTP servers the client is to use.
- option ntp_servers [address_list]: Lists the addresses of the NTP (Network Time Protocol) servers the client is to use.
- option pop_server [address_list]: Lists the addresses of the POP3 servers the client is to use.
- option routers [address_list]: Defines the default router.
- option smtp_server [address_list]: Lists the addresses of the SMTP servers the client is to use.
- option subnet_mask [mask]: Defines the subnet mask. If this option is undefined, the network mask from the subnet statement is used.
- option time_offset [value]: Defines the offset from Coordinated Universal Time of your time zone (i.e., _5 would mean Eastern Standard Time).
- option www_server [address_list]: Lists the addresses of the Web servers available to the client.

Classically, this would be used to define proxy Web servers the client is to use. There are many more configurable client options than those listed, but these are the ones most commonly used. For a full list of available options, please read the man pages for dhcpd.conf and dhcp_options.

Self Assessment

Fill in the blanks:

10. ..... was designed to give all possible TCP/IP configuration parameters to client computers using the client/server model.
11. When DHCP starts, it reads the file .....
12. The ..... statement can be used to supply configuration options to a group of host or subnet statements.

Notes

13. ..... statement is used to inform the server whether or not it should act as both a BootP and DHCP server or just a DHCP server.
14. server_name ["name"] option defines the ..... of the server.
15. ..... option defines the maximum length, in seconds, of a lease length.

10.5 Summary

- 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.
- 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.
- File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers. FTP works on a client/server model.
- vsftpd is an FTP daemon which is easy to install, set up, and maintain.
- Apache was first developed on the various Unix/Linux/BSD platforms, and then recently ported to Win32.
- Prefix=PREFIX option specifies the directory where the Apache files will be installed.
- The Dynamic Host Configuration Protocol (DHCP) is nothing new to many Linux users. Most people have a DHCP client installed on their computers so they can connect to the Internet via cable or DSL modems.
- The configuration file for your DHCP server is the /etc/dhcpd.conf file. This file has numerous configuration commands that operate the server and provide configuration information to the clients.

10.6 Keywords

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.

DHCP (Dynamic Host Configuration Protocol): It is a communications protocol that lets network administrators centrally manage and automate the assignment of Internet Protocol (IP) addresses in an organization's network.

FTP: File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers.

IIS: IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Windows NT and Windows 2000 Server operating systems.

IP Leasing: A DHCP client initiates a conversation with the server, and the server leases an IP address to the client for a configurable period of time.

PREFIX: This option specifies the directory where the Apache files will be installed.

Vsftpd: It is a GPL licensed FTP server for UNIX systems, including Linux. It is secure and extremely fast.

Web server: It is a program that runs on a host computer that serves up web sites.

10.7 Review Questions

1. Explain the concept of installing the FTP server.

Notes

2. Elucidate the main features of VSFTPD.
3. Describe the installation and configuration of your server for apache.
4. Discuss the running of apache as a console application and as a service.
5. Explain the installation of the DHCP server and what are the packages need to be installed with it.
6. Explain the configuration problems of the DHCP server.
7. How to install the DHCP server with the multiple NIC's?
8. Compare and contrast the host command and nslookup Command.
9. Discuss the differences In Fedora and Redhat DNS File Locations.
10. Discuss the steps used for downloading and Installing the DHCP Package.

Answers: Self Assessment

1. Domain Name System (DNS)
2. BIND
3. caching
4. Localnets
5. File Transfer Protocol (FTP)
6. Daemon
7. PREFIX
8. mod_alias
9. mod_ssl
10. DHCP
11. /etc/dhcpd.conf
12. Group
13. Bootp
14. host name
15. max_lease_time [seconds]

10.8 Further Readings

Books

Christopher Negus, Linux Bible, Wiley.

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.

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

Brian Ward, How Linux Works, No Starch Press.

Online links

http://www.centos.org/docs/5/pdf/Virtual_Server_Administration.pdf

https://speakerdeck.com/futureshocked/a-linux-server-administration-tutorial-for-beginners

http://www.yolinux.com/TUTORIALS/DHCP-Server.html

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch08_:_Configuring_the_DHCP_Server#.UW48LaKBmpc