

# Chapter 3

## Basic CLI Commands

This chapter introduces the new user to the basic commands that are required to navigate around a Unix / Linux system in order to perform fundamental tasks. Some of the fundamental requirements as an administrator are presented. With this chapter, the Lab Manual is used to practice the various commands. Initially, the commands used in this chapter will be issued as the administrator, future chapters will require the user to log in as a normal user.

### Concepts Learned in this Chapter

- Initial Logging on and off of a system
- Basic commands to navigate and function on a Unix / Linux system

## Table of Contents

|  |    |
|--|----|
| Basic CLI Commands.....                        | 1  |
| 3.1 Getting Started – Logging On .....         | 5  |
| 3.1.1 Command Line Interface .....             | 5  |
| 3.1.2 Login and Password .....                 | 5  |
| 3.1.4 Shutdown .....                           | 6  |
| 3.1.5 Rebooting .....                          | 6  |
| 3.2 Directory Structure .....                  | 7  |
| 3.2.1 Drive Designator.....                    | 7  |
| 3.2.2 Drive Designation.....                   | 7  |
| 3.2.3 Drive Partitions.....                    | 8  |
| 3.2.4 Mounting a Partition.....                | 8  |
| 3.3 Partition Requirements.....                | 8  |
| 3.3.1 Minimum Partition Requirements .....     | 8  |
| 3.3.2 Linux Directories .....                  | 9  |
| 3.3.3.1 Hard Drive Designation.....            | 12 |
| 3.3.4 Partition Management.....                | 12 |
| 3.3.4.1 fstab or File System Table.....        | 13 |
| 3.3.4.2 mtab or Mount Table.....               | 14 |
| 3.3.5 Floppy Drive Designation.....            | 14 |
| 3.3.6 Alternative Directory “Names” .....      | 15 |
| 3.3.7 Navigating the Directory Structure ..... | 15 |
| 3.3.7 Case Sensitivity .....                   | 16 |
| 3.3.8 CLI Prompt .....                         | 16 |
| 3.3.9 Path to Working Directory .....          | 16 |
| 3.4 Listing Directory Contents .....           | 16 |
| 3.4.1 Listing Files .....                      | 16 |
| 3.4.2 Listing File Attributes .....            | 17 |
| 3.4.2.1 Listing A File’s Attributes .....      | 17 |
| 3.4.2.2 File Type .....                        | 17 |
| 3.4.2.3 File Permissions .....                 | 18 |
| 3.4.2.4 File Links .....                       | 19 |
| 3.4.2.5 File Owner .....                       | 19 |
| 3.4.2.6 File Group Name .....                  | 19 |
| 3.4.2.7 File Size .....                        | 19 |
| 3.4.2.8 File Last Modification Date .....      | 20 |
| 3.4.2.9 File Name .....                        | 20 |
| 3.4.3 Listing Directories Only .....           | 20 |
| 3.4.4 Listing Hidden Files .....               | 20 |
| 3.4.5 Listing File Inode .....                 | 20 |
| 3.4.6 Combined Commands .....                  | 21 |
| 3.4.7 Additional Options .....                 | 21 |
| 3.5 Editing Files .....                        | 21 |
| 3.5.1 vi Text Editor .....                     | 22 |
| 3.5.2 emacs Text Editor .....                  | 25 |
| 3.5.3 joe Text Editor .....                    | 26 |

|   |    |
|---|----|
| 3.5.4 nano and pico Text Editors .....        | 26 |
| 3.5.5 cat .....                               | 27 |
| 3.5.6 touch .....                             | 27 |
| 3.6 Displaying a File's Contents .....        | 28 |
| 3.6.1 Viewing Text Files .....                | 28 |
| 3.6.1.1 cat .....                             | 28 |
| 3.6.1.2 more .....                            | 29 |
| 3.6.1.3 less .....                            | 29 |
| 3.6.1.4 head .....                            | 29 |
| 3.6.1.5 tail .....                            | 30 |
| 3.6.1.6 tac .....                             | 30 |
| 3.6.1.7 hexdump .....                         | 30 |
| 3.6.1.8 od .....                              | 30 |
| 3.6.1.9 xxd .....                             | 30 |
| 3.7 Adding Users .....                        | 31 |
| 3.7.1 Adding a User .....                     | 31 |
| 3.7.2 Assigning User Password .....           | 31 |
| 3.7.3 User Home Directory .....               | 33 |
| 3.7.4 Deleting a User .....                   | 33 |
| 3.8 Password File .....                       | 34 |
| 3.8.1 Passwd Data Entry .....                 | 34 |
| 3.8.2 Improving Password Security .....       | 35 |
| 3.8.2.1 Password File Conversion .....        | 35 |
| 3.8.2.2 Shadow File Contents .....            | 36 |
| 3.8.3 Random Password Generator .....         | 36 |
| 3.9 Creating Groups .....                     | 37 |
| 3.9.1 Creating a Group .....                  | 37 |
| 3.9.2 User Group Membership .....             | 38 |
| 3.10 Piping Utility .....                     | 38 |
| 3.11 Grep Utility .....                       | 39 |
| 3.12 Director Utility .....                   | 40 |
| 3.13 Copying Files .....                      | 41 |
| 3.14 Moving and Renaming Files .....          | 42 |
| 3.15 Creating and Deleting Directories .....  | 42 |
| 3.15.1 Creating a Directory.....              | 42 |
| 3.15.2 Deleting a Directory.....              | 42 |
| 3.16 Deleting Files .....                     | 43 |
| 3.16.1 rm.....                                | 43 |
| 3.16.2 shred.....                             | 43 |
| 3.17 Links .....                              | 44 |
| 3.18 Shell Interpreters .....                 | 45 |
| 3.19 System Help.....                         | 46 |
| 3.19.1 Man Pages .....                        | 47 |
| 3.19.2 Info Pages .....                       | 48 |
| 3.19.2.1 Navigating Through Info.....         | 49 |
| 3.19.3 Apropos Pages .....                    | 50 |
| 3.20 Removable Devices and Mount Points ..... | 50 |
| 3.20.1 Mount Command .....                    | 50 |

|   |    |
|---|----|
| 3.20.2 Mount or Media Directory.....            | 51 |
| 3.21 Background Processes .....                 | 52 |
| 3.22 Alternate Terminals .....                  | 53 |
| 3.23 Installing Applications .....              | 53 |
| 3.23.1 Red Hat Package Manager .....            | 53 |
| 3.23.2 Red Hat Updates.....                     | 54 |
| 3.23.3 Update Managers .....                    | 55 |
| 3.23.3.1 YUM .....                              | 55 |
| 3.23.3.2 Apt-Get.....                           | 56 |
| 3.23.4 Tarballs .....                           | 57 |
| 3.24 File Location Utilities .....              | 57 |
| 3.24.1 find .....                               | 57 |
| 3.24.2 locate .....                             | 58 |
| 3.24.3 which .....                              | 59 |
| 3.24.4 whereis .....                            | 59 |
| 3.24.6 glimpse .....                            | 59 |
| 3.25 Switch User .....                          | 60 |
| 3.26 Switch User by Command .....               | 61 |
| 3.27 Starting X Windows .....                   | 63 |
| 3.28 Printer Configuration , .....              | 64 |
| 3.28.1 Local Printer .....                      | 67 |
| 3.28.2 Remote Unix Printer .....                | 68 |
| 3.28.3 Remote Windows Printer .....             | 68 |
| 3.28.4 Remote Novel Printer .....               | 69 |
| 3.28.5 Jet Direct .....                         | 69 |
| 3.28.6 Configuration Utilizing CUPS .....       | 70 |
| 3.28.6.1 Switching to CUPS .....                | 70 |
| 3.28.6.2 CUPS System Files for Printer .....    | 71 |
| 3.28.6.3 Configuring a Printer using CUPS ..... | 71 |
| 3.29 Commands Used in this Chapter.....         | 72 |
| 3.30 Chapter Review Questions.....              | 74 |

### 3.1 **Getting Started – Logging On**

The very first time you start Linux, you will be prompted for a logon. This is the first level of security – the user must enter a valid username and password. As the Administrator, your Username is **root**. If you are another user, the Administrator will assign you a username. When you enter the password, no characters are echoed back to you, someone looking over your shoulder will not be able to see how many characters you type. The initial login will appear like the following:

***Fedora Core release 3 (Heideberg)  
Kernel 2.6.12-1.1381\_FC3 on an i686***

***steamed login: root  
Password: {your administrator's password}***

#### 3.1.1 **Command Line Interface**

It is assumed that during installation, you set up the system to boot into the **Command Line Interface (CLI)** rather than the X Windows GUI. Most users who are coming from the Microsoft Windows environment may never have worried about this mode and are unfamiliar with it. An action that is always required at the CLI is after entering a command – you must also hit the ENTER (RETURN) KEY. To start this exercise, a small reminder of this will be the @ symbol.

When you installed Linux, you established a password for the user known as **root**. At this time we are going to set up some simple rules:

1. Except for the exercises contained herein, never give your password out to anyone.
2. In the learning process, we will be making changes to our passwords, so we will be writing them down - OFTEN. You should normally not write down your password – someone might find it.

#### 3.1.2 **Login and Password**

When Linux or Unix first boots up we will be prompted for a user-id and password. At the login, as the administrator, enter:

***username: root***

You are then prompted for your password. Enter the password as you created it during installation. After we have entered the correct response, we will be given the CLI prompt. The default prompt, when using the Bash shell, appears as:

***[root @ hostname present-directory] #***

If we log on as the root administrator, the last character will be a “#”. If we log on as a normal user, then we will have the “\$”. The administrator is given the username of “**root**”. As a normal user, the prompt will appear as:

***[username @ hostname present-directory] \$***

### 3.1.3 Logout

Some times we need to log off of the host computer, but do not wish to shut down the computer. At the prompt, we issue the command:

**exit** or  
**logout**

The host will then prompt us for our username and password.

### 3.1.4 Shutdown

If we need to shut down the computer, we issue the command:

**shutdown -h now**

This tells the operating system to immediately halt (h) the operation (now). After we get tired of this, we will use a shortened version – **halt**.

We have the option to delay the shutdown, instead of specifying “now”, we can specify the number of minutes that the shutdown is to be delayed by. For instance, we could issue the command:

**shutdown -h 5**

This would set the system up to shutdown in 5 minutes. This might be useful if your host was on a battery backup (UPS) system, allowing the system to run for 5 minutes from the battery, and hoping that the power came back on in that time. If such should happen, we would be able to cancel the shutdown.

We also have an option to issue a message to everyone logged onto the system, informing them of the pending shutdown. We might issue the command

**shutdown -h 5 Please log off, shutting down in a few minutes!**

Additional commands that may be used to shut down the system include:

**halt** Immediately issues a shutdown command to terminate the system operation.

**poweroff** Immediately issue a shutdown command to terminate the system operation and to issue a command to the power supply to turn off power to those systems that support that functionality.

### 3.1.5 Rebooting

There is another requirement where we need to reboot the host. Although Linux and Unix are designed to minimize this, we sometimes do it anyways. To reboot, we issue the command

**shutdown -r now**

This tells the operating system to immediately reboot the system (now). After we get tired of this, we will use the shortened version – **reboot**. We can also use the “three-finger-salute” – **CTL-ALT-DEL**.

Additional commands include:

- k Send only a warning message
- c Cancel an already running shutdown that is time delayed

## **3.2 Directory Structure**

To understand how you navigate around the Linux system, you need to understand its structure, and the minimum requirements.

### **3.2.1 Drive Designator**

A hard drive is a long term storage device that may be written to and read from. In the Microsoft world that one is normally use to, a hard drive is assigned a letter designation, typically “C” for the first hard drive partition. If the drive is divided into smaller sections, or **logical partitions**, each partition is assigned a different letter. When a system has more than one hard drive, each drive / partition on the drive is assigned a letter designation. In the old “DOS” days, one would change between the different partitions by “**Changing Directory**”, or “**cd**”. Thus one would be able to access or read information on different partitions. It was sometimes necessary to divide a hard drive into multiple partitions because the DOS operating system was only able to support a partition of a limited size.

Unix and Linux handle how a hard drive is accessed in a different way. There is no such thing as a drive or partition designator. The drive or partition is part of a whole directory structure. Each drive may be divided into multiple partitions if desired – and this may be desirable or necessary in certain situations.

The directory structure in Unix and Linux is an inverted tree. The very top of the tree is called “**The Root**”, and is designated with the forward slash, “/”. Below the / are a number of directories, which will be discussed in more detail later. Each partition of the hard drive is assigned a location within the directory tree structure during the installation of the operating system or when a new hard drive is installed. The directory structure is shown below.

The number of partitions that is required depends on the hard drive size and user requirements. For a given system, the minimum number of partitions is two, but more may be required, depending upon the hard drive size and user requirements.

### **3.2.2 Drive Designation**

In Microsoft, each hard drive is given a designator. These are known as **Drive0**, **Drive1**, and so on. In Unix and Linux, each drive is given a designator, indicating its type, and a letter stating which one it is. Thus we have:

- hd Hard Drive
- fd Floppy Drive
- sd SCSI Drive
- hda First Hard Drive
- hdb Second Hard Drive
- fda First Floppy Drive
- sda First SCSI Drive
- sdb Second SCSI Drive

### 3.2.3 Drive Partitions

Each drive may be subdivided into multiple partitions. To keep track of each partition, starting with the number “1”, each partition is numbered on the specified drive. Thus if we have three partitions on the first hard drive, we would have the designations:

- hda1 First partition on Hard Drive A
- hda2 Second partition on Hard Drive A
- hda3 Third partition on Hard Drive A

The same format process is also used for SCSI drives.

### 3.2.4 Mounting a Partition

After a drive has been broken up into multiple partitions, each partition must be assigned a location within the directory structure. This is typically performed during the installation of the operating system, but may have to be assigned at a later time if a new hard drive is added. Add a new drive is covered in a later chapter.

Mounting a partition is subject to some limitations. There are some directories under the root (/) that are not allowed to be on a partition other than the partition on which the root is installed. Separate partitions are commonly used to support the subdirectories of:

|       |      |
|-------|------|
| /boot | /opt |
| /home | /var |

Most commonly, the /boot directory is on a separate partition if the primary partition (one that the root is mounted on) is greater than 5 Gigabytes. The /home directory is commonly put on a separate directory on a very active system, which allows control of the user's home directories and to protect the system in case the main drive should die, thus keeping the users active after the main partition has been restored. The /var directory contains most of the server data that is available to the public, and hence may be placed on a separate partition to protect it in case of a failure.

Of course, additional user defined partitions such as “/backup” and “/data” may be created by the administrator for network usage, if such is desired and appropriate.

## 3.3 Partition Requirements

During the installation of Linux, at least two and most likely three partitions will need to be created. Additional partitions may also be created depending upon the system or user requirements.

### 3.3.1 Minimum Partition Requirements

In a minimal system, the minimum number of partitions is two. These are the SWAP partition and the primary partition. If you have a hard drive greater than five Gbytes, you will need to have a minimum of three partitions for older versions, two for present versions. These partitions are:



1. **SWAP** This partition is reserved for the OS to temporarily store RAM information on the hard drive when you do not have sufficient RAM memory. Memory values are “swapped” back and forth to the hard drive as they are needed. The minimum size of the SWAP partition should be equal to twice amount of RAM on the system. If you have a large hard drive, make it larger.
2. **/ “the root”** This is the main partition where all of the information is maintained. Typically this is the remainder of the hard drive. We denote the “Native” directory with the forward slash “/”, and call it **the root**.
3. **/boot** This partition is required if the hard drive partition for **the root** is to be greater than 5 Gbytes. This is required only for earlier releases of Red Hat. Present versions of Linux no longer require this restriction, but it may be a good practice to perform.

Additional partitions may be created during the installation, or if an additional hard drive is added at a later date. For security and re-installation of a crashed system reasons, the /home directory is often placed on a separate partition. Several other standard partitions may also be set up on separate partitions if desired, but will not be discussed here.

### 3.3.2 Linux Directories

When using the DOS / Windows system, the first hard drive partition is designated as the **C:\** drive. When using a Unix or Linux system, a totally different method of assigning drives and partitions is utilized as defined by the FDS 2.2 specification.

Unix and Linux start with the primary partition on the designated drive. This is always designated as the **ROOT** partition, and given the designator “/”. We often refer to the top of the directory system as “**the root**”. During the installation of Linux, several subdirectories are installed. These are set up as:

|            |   |
|------------|---|
| /          | The ROOT directory that forms the base of the file system. All files and directories are logically contained in the root directory, regardless of their physical location.  |
| /bin       | Contains normal user Linux executable programs.   |
| /boot      | Contains the Linux kernel and other programs, including the system.map and boot.b files required to boot the system.  |
| /mnt       | Contains the path to other temporary physical devices such as the CD ROM drive and the Floppy drive. Before they may be utilized, the respective device must be mounted. In later versions of Fedora, the mounting of removable of media (Floppy and CDROM) has been moved to the /media directory. |
| /mnt/cdrom | Contains the file structure of the CD ROM after it has been mounted in the Red Hat 7 through 9 versions.  |

|               |   |
|---------------|---|
| /mnt/floppy   | Contains the file structure of the Floppy drive after it has been mounted in the Red Hat 7 through 9 versions.  |
| /media        | Contains the removable media in the Fedora Core distribution.   |
| /media/cdrom  | Contains the file structure of the CD ROM after it has been mounted in the Fedora Core distribution.  |
| /media/floppy | Contains the file structure of the Floppy drive after it has been mounted in the Fedora Core distribution.  |
| /dev          | Contains device driver files. Linux treats each device as being a special file and all such files are located in the device file. Common devices include the Communication Ports (Comm.), Parallel Ports and Sound Card.  |
| /etc          | Contains system configuration files and shell scripts for the Linux system.   |
| /etc/rc.d     | Contains the system initialization scripts used during bootup.  |
| /etc/X11      | Contains configuration of the X-Windows server and X scripts.   |
| /home         | Contains the subdirectories for all users that have been added to the system.   |
| /lib          | Contains the C Language (C and C++) and system libraries.   |
| /lost+found   | Contains the <b>files</b> that have been lost during an improper shutdown, if such should happen. Every partition has a <i>lost+found</i> directory.  |
| /proc         | A special directory that contains information about various aspects of the Linux system. This is a virtual directory or often called a RAM Drive, i.e., it does not really exist on the hard drive. It is used by the system to maintain the operation of Linux and is maintained in RAM. It is created every time the system is booted, and parameters set from the various configuration files. |
| /root         | The home directory of the <b>Root</b> administrator. Other users are located under the /home directory.   |
| /sbin         | Contains system executable files such as mount, halt, umount, and shutdown that are reserved for the administrator. Processes are critical to the operating system, and are normally run by the OS itself. Normal users do not have a need to access this directory.  |
| /tmp          | Contains files which are subject to being deleted on a reboot. A program may create temporary data files in this directory and delete them as appropriate. Do not confuse this with the SWAP partition, which is used by the system to temporarily remove processes from RAM until they are again required.   |
| /usr          | Contains the subdirectories of system programs and online documentation. All user-level binary files and non-vital system binaries  |

**/var** Contains various system administrative files, such as system spool, logging, mail, Internet access (ftp and http), printing and cron.

**/opt** A carry over directory from the UNIX days. It was used for unbundled software (download programs). The KDE desktop may utilize the space.

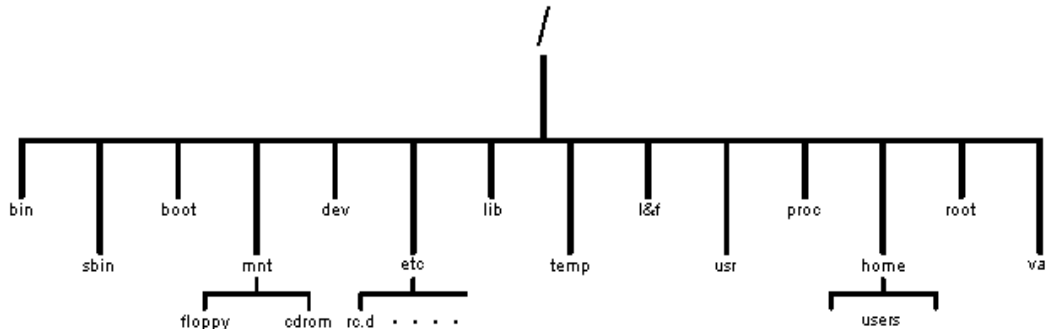


Figure 3.1: Basic Directory Tree

### 3.3.3 Hard Drive Partitions

As was previously noted, when you are installing Linux you will typically be required to create three directories (/ , /boot, SWAP), but it may be wise to create more in a productive system. Typical additional partitions might include:

**/home** User's home directory. If you desire to have multiple Linux systems, you can have just one home directory for all Operating Systems. Additionally, if you do need to reinstall, the users will still be retained.

**/var** This is where the system retains ongoing system logs, Internet files (web pages and ftp files for general access), and user email that has not yet been delivered. If there a system crash, the information would be retained. This partition contains extensive information, and thus should be of significant size (extreme minimum 2 G Byte).

**/usr** This is where common system and general application files are maintained. Again if you need to modify the system, you will have this retained for the new installation.

**/opt** This directory might be used for the installing of new programs.

As an example of the tree structure that includes multiple drive partitions, one might set up the drive structure as:

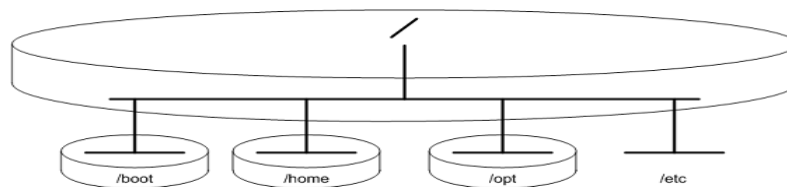


Figure 3.2: Hard Drive Partitions and Subdirectories

### 3.3.3.1 Hard Drive Designation

A typical computer is capable of supporting up to four IDE devices, which include Hard Drives, CDROM Drive, DVD Drive, Zip Drive, and other devices. If you open up a computer you typically find two ribbon cables (more modern system now have round cables), the first cable is referred to as the “**Primary**” and can support two drives. The second cable also supports two drives designated as “**Secondary**” interface. Each device is connected to either the Primary or the Secondary cable and given the designation of either “**Master**” or “**Slave**”. Within Unix and Linux, each drive interface is given a drive designator. For the IDE interface, each device is considered a Hard Drive, designated as either the “**Master**” or “**Slave**”. Thus, we give the designations as:

|            |                        |                          |
|------------|------------------------|--------------------------|
| <b>hda</b> | IDE Primary – Master   | hd stands for Hard Drive |
| <b>hdb</b> | IDE Primary – Slave    |                          |
| <b>hdc</b> | IDE Secondary – Master |                          |
| <b>hdd</b> | IDE Secondary – Slave  |                          |

If a hard drive has multiple partitions created on it, then each partition is specified with a numeric value, starting with 1. Even if only one partition exists on the drive, then it is designated as partition one. Thus if we have the first hard drive with three partitions, they would be designated as:

|             |  |
|-------------|--|
| <b>hda1</b> | First partition of the Primary Master drive  |
| <b>hda2</b> | Second partition of the Primary Master drive |
| <b>hda3</b> | Third partition of the Primary Master drive  |

If necessary, more than one IDE interface card may be installed into a computer, allowing a system to have up to eight or more IDE devices.

If the system is equipped with SCSI Drives, the same type of designation is provided, with a minor modification of designator name. SCSI devices are designated as:

|            |               |                           |
|------------|---------------|---------------------------|
| <b>sda</b> | SCSI Device 0 | sd stands for SCSI Device |
| <b>sdb</b> | SCSI Device 1 |                           |
| <b>sdc</b> | SCSI Device 2 |                           |
| <b>sdd</b> | SCSI Device 3 |                           |
| <b>sde</b> | SCSI Device 4 |                           |
| <b>sdf</b> | SCSI Device 5 |                           |
| ...        |               |                           |

Older SCSI interfaces support up to eight devices (device 7 is typically assigned to the SCSI card), while newer interfaces support up to sixteen devices.

### 3.3.4 Partition Management

In order for Unix / Linux to track the location of various partition, two files are utilized as tables during the boot process. These table files are **/etc/fstab** and **/etc/mtab**.

### 3.3.4.1 **fstab** or File System Table

The **fstab** file contains descriptive information about the various file systems that may be on your system. The following is an example of a **/etc/fstab** file content for an Red Hat 7 system. Newer distributions, specifically Fedora, now support the automatic detection of removable media, such as a USB Flash Drive. In this case, the utility

**fstab-sync** must be used to update the **/etc/fstab** file.

| Block Device    | Mount Point       | File Type | Options                        | Dumped | Check |
|-----------------|-------------------|-----------|--------------------------------|--------|-------|
| LABEL=/1        | /                 | ext3      | defaults                       | 1      | 1     |
| LABEL=/boot     | /boot             | ext3      | defaults                       | 1      | 2     |
| none            | /dev/pts          | devpts    | gid=5,mode=620                 | 0      | 0     |
| none            | /dev/shm          | tmpfs     | defaults                       | 0      | 0     |
| none            | /proc             | proc      | defaults                       | 0      | 0     |
| none            | /sys              | sysfs     | defaults                       | 0      | 0     |
| LABEL=SWAP-hda2 | swap              | swap      | defaults                       | 0      | 0     |
| /dev/hdc        | /media/cdrecorder | auto      | pamconsole,exec,noauto,managed | 0      | 0     |

#### First Column – **fs\_spec**

Specifies the block device or remote file system. An existing partition on the hard drive is specified with a **Label**. Other block devices utilize specific device drivers in order for them to function, such as **/dev/cdrom**.

#### Second Column – **fs\_file**

Specifies the type of file mount point of the device.

#### Third Column – **fs\_type**

Specifies the mount point of the device that has been created. Linux supports a wide range of file types, including all of the previous Unix types, Microsoft, IBM, and proprietary. To see all of the supported file systems, issue the command:

```
# fdisk /dev/hda
```

then click the **L** key to display the list of supported file types. Use caution when issuing this command – it is very powerful and can destroy your file system.

#### Fourth Column – **fs\_mtops**

Specifies the mount options that are associated with the device. A comma separated list of device options. Common options include:

|                 |   |
|-----------------|---|
| <b>defaults</b> | Standard options: rw, suid, dev, auto, nouser, and async  |
| <b>async</b>    | All I/O to the device is performed asynchronously.  |
| <b>suid</b>     | Allows <b>set-user-identifier</b> or <b>set-group-identifier</b> .  |
| <b>dev</b>      | Interpret character or block special characters.  |
| <b>auto</b>     | Normal mount on system boot.  |
| <b>nouser</b>   | Non-root user not allowed to mount the file system.   |
| <b>owner</b>    | Only the file system owner may mount the device. When booting, the owner of the file system is the <b>root</b> administrator.                         |
| <b>kudzu</b>    | Specifies that devices have been detected by the system. During bootup, the system automatically checks for new hardware utilizing the kudzu process. |
| <b>ro</b>       | Read Only.  |
| <b>rw</b>       | Read / Write access (default).  |

**Noauto** Device may only be mounted explicitly, that is, the device must be mounted using the **mount** command.

#### **Fifth Column – fs\_freq**

Specifies whether a file system is to be dumped for backup purposes.

**0** Full backup.

**1 – 9** Incremental backup, 9 being the lowest.

#### **Sixth Column – fs\_passno**

Used by the fsck utility to periodically check the file system for errors. The value specifies the order of checking, a “0” indicates to not check the file system.

### **3.3.4.2 mtab or Mount Table**

The **/etc/mtab** file may initially be hard to read, because it is not column formatted or comma delimited. If you display the file, it will look different from what is displayed below, as the printout has been formatted for easier reading. The file may be displayed using either the **cat** or **mount** command.

| <b>Mount Point</b> | <b>Mount Partition</b>   | <b>Mount Type</b> | <b>Access</b>     |   |   |   |
|--------------------|--------------------------|-------------------|-------------------|---|---|---|
| /dev/hda3          | /                        | ext3              | rw                | 0 | 0 |   |
| none               | /proc                    | proc              | rw                | 0 | 0 |   |
| none               | /sys                     | sysfs             | rw                | 0 | 0 |   |
| none               | /dev/pts                 | devpts            | rw,gid=5,mode=620 | 0 | 0 | 0 |
| usbfs              | /proc/bus/usb            | usbfs             | rw                | 0 | 0 |   |
| /dev/hda1          | /boot                    | ext3              | rw                | 0 | 0 |   |
| none               | /dev/shm                 | tmpfs             | rw                | 0 | 0 |   |
| none               | /proc/sys/fs/binfmt_misc | binfmt_misc       | rw                |   | 0 | 0 |
| sunrpc             | /var/lib/nfs/rpc_pipefs  | rpc_pipefs        | rw                |   | 0 | 0 |

#### **First Column – Mount Point**

Specifies the mount point for the device. hda is a hard drive, usbfs is a usb interface, none is a virtual drive.

#### **Second Column – Mount Partitions**

Specifies the name of the partition.

#### **Third Column – Mount Types**

Specifies the file type of the partition.

#### **Fourth Column – Accessed**

Specifies read / write access

#### **Fifth Column – Unknown**

Unknown at this time.

#### **Sixth Column - Unknown**

Unknown at this time.

### **3.3.5 Floppy Drive Designation**

In a similar manner to the Hard Drive, the floppy drive is also given a designation. We commonly only find one floppy device installed on today's system, because floppy drives are going out of usage, This unit is given the designator:

**fd0** First Floppy Device

### 3.3.6 Alternative Directory “Names”

Sometimes we refer to a directory with special characters. These are:

- . A single dot (period), represents the **present directory**.
- .. Double dot (periods), represents the **parent directory** of your present location. For example:

| <u>Directory</u> | <u>Parent</u> |
|------------------|---------------|
| /mnt/floppy      | /mnt          |
| /usr             | /             |
| /var/www/html    | /var/www      |

### 3.3.7 Navigating the Directory Structure

The basic method of changing locations from one directory to another is the **change directory** command. It is denoted as **cd**.

There are several different options for the **cd** command:

1. **cd dirname** Changes to the directory immediately below the present location, a **child** directory.
2. **cd /path/dirname** Changes to the directory specified by the **Fully Qualified Pathname**.
3. **cd** Changes to the username home directory.
4. **cd -** Changes to the previous directory. Excellent tool for switching back and forth between two directories.
5. **cd .** Changes to the present directory. Valid command, but it does not do anything.
6. **cd ..** Changes to the parent directory of your present location, that is, one level up.

Examples of the Change Directory include:

```
$ pwd
/home/jdoe
$ cd /var/log           change to specified Fully Qualified
Pathname
$ pwd
/var/log
$ cd                   change to user's home directory
$ pwd
/home/jdoe
$ cd -                change to previous directory
$ pwd
/var/log
$ cd ..              change to the parent directory
$ pwd
/var
```

### 3.3.7 Case Sensitivity

Something very important to note – Unix and Linux are **CASE SENSITIVE!**  
All of the following commands are different.

1. cd
2. cD
3. Cd
4. CD

In fact, only the command 1(cd) is valid, the rest will return an error.

### 3.3.8 CLI Prompt

After you have logged on, you are provided with a prompt. The default prompt for Red Hat and Fedora Core when using the Bash shell is in the form of:

**[username@hostname directory]\$** (if administrator, the \$ is a #)

where:

|           |                                      |
|-----------|--------------------------------------|
| username  | User ID name of logged on user.      |
| hostname  | The name of the Linux / Unix system. |
| directory | The present directory location.      |

If one is in his / her home directory, such as immediately after login, the system may show the tilde “~”, which is the symbol used to indicate the home directory.

### 3.3.9 Path to Working Directory

Although the prompt specifies your present directory location, it does not specify the full path. The command to display the full path is:

**pwd**

This immediately displays the full **Path** to the **Working Directory**.

## 3.4 Listing Directory Contents

Now that we have learned to maneuver around the different directories, we need to learn how to observe the contents.

### 3.4.1 Listing Files

To see the contents, we have one command, with various options. The command to list the contents of a directory is:

**ls**

The listing may appear similar to:

|            |      |          |         |       |      |
|------------|------|----------|---------|-------|------|
| arch       | date | gunzip   | mv      | sleep |      |
| ash        | dd   | gzip     | netstat |       | sort |
| ash.static | df   | hostname | nice    | sty   |      |
| ...        |      |          |         |       |      |

This is the similar to the DOS “dir” command.



The contents of a directory may consist of both files and additional directories. Issuing the command **ls** will list the contents of both. In some versions, the different types of files and directories may be displayed in color.

Whenever we use the term **list**, we imply the listing of a directory's contents.

### 3.4.2 Listing File Attributes

A file has a set of attributes that define file type, permissions, links, owner, group name, size, last modified date, and file name.

#### 3.4.2.1 Listing A File's Attributes

The first option that we need to use is:

```
ls -l
```

This provides a different format of listing the files. This appears as:

```
-rwxr-xr-x 1 root root 2872 Aug 26 2001 arch  
-rwxr-xr-x 1 root root 94364 Jun 24 2001 ash  
-rwxr-xr-x 1 root root 472492 Jun 24 2001 ash.static  
...
```

To explain what each of the above fields are, we will look at the entry that ends in "arch":

|                         |   |
|-------------------------|---|
| -rwxr-xr-x              | File type and permissions                 |
| -                       | File (-) or Directory (d) designator      |
| rwxr-xr-x               | File / Directory Permissions              |
| 1                       | Number of links or pointers to this file  |
| root (1 <sup>st</sup> ) | Owner or originator of the file           |
| root (2 <sup>nd</sup> ) | Group name for special access to the file |
| 2872                    | File size in Bytes                        |
| Aug 26 2001             | Date of last access to file               |
| arch                    | File name                                 |

This provides us a listing of all files, directories and links, with details of the attributes of such. At this time, we are reviewing only how to interpret the information, we will later go into more detail on how to modify the values.

#### 3.4.2.2 File Type

The very first character of the listing specifies the file type. There are five different types of files. These are:

1. Simple / Ordinary
2. Directory
3. Link
4. Device
5. Named Pipe (FIFO)

For the moment we only want to look at the first character of the string "-rwxr-xr-x". This tells us the following:

The first group of letters and dashes provides the file type and the operational attributes or **permissions**. Look at the sequence as one character plus three sets of 3 permissions characters:

```

F   r w x   r w x   r w x
      \ -1- /   \ -2- /   \ -3- /   field grouping

```

**F      File type**

**–**      normal file that may be read, written to, or executed  
**d**      directory file, a file that specifies additional files within a directory  
**l**      link to another file  
**c**      character device (serial port, keyboard, monitor, mouse)  
**b**      block device (hard drive, cd-drive, floppy, USB thumb drive)  
**p**      pipe

A file is a data object that contains various characters, be they readable text, an executable file, or a binary data file.

A directory is a special file that contains a list of additional files along with the file attributes.

A link is a pointer to another file in another directory. This will be explained later.

A character device transfers data to an interface for external transmission.

A block device is a storage device for retaining data for a long period.

A pipe is used to transfer data between processes within the kernel.

### 3.4.2.3      **File Permissions**

The next block of characters represent the File Permissions.

```

r w x   r w x   r w x
      \ -1- /   \ -2- /   \ -3- /   field grouping

```

The three groups of letters have like meaning, each with a different function. The **r w x** provides permission for:

|          |         |  |
|----------|---------|--|
| <b>r</b> | read    | The file may be opened for reading.  |
| <b>w</b> | write   | The file may be written to.  |
| <b>x</b> | execute | The file is executable, typically a binary executable program, but may also be an executable script. |

As appropriate, each **r w x** attribute is set with either a **0** or a **1**, where a 0 means the action is not allowed, and a 1 means it is allowed. A 0 will be displayed as a “–”, a 1 is displayed as an **r**, **w**, or **x** as appropriate.

Note that there are three groups of “rwxs”. These are set up to represent various users or entities on the system. These are set as:

| Group | Group Designator | User / Entity                          |
|-------|------------------|--|
| 1     | <b>u</b>         | <b>File owner / creator</b>            |
| 2     | <b>g</b>         | <b>User group name</b>                 |
| 3     | <b>o</b>         | <b>Others / world or everyone else</b> |

Each **r w x** represent the octal equivalent of the 1 / 0 binary combination of the rwx number set. This is a simple binary – octal conversion.

| <b>r</b> | <b>w</b> | <b>x</b> | <b>octal</b> | <b>Operational Status</b>                                   |
|----------|----------|----------|--------------|---|
| 0        | 0        | 0        | 0            | file is not accessible to anyone except root <sup>1</sup> , |
| 0        | 0        | 1        | 1            | file is executable only                                     |
| 0        | 1        | 0        | 2            | file is writeable only                                      |
| 0        | 1        | 1        | 3            | file is writeable and executable                            |
| 1        | 0        | 0        | 4            | file is read only   |
| 1        | 0        | 1        | 5            | file is readable and executable                             |
| 1        | 1        | 0        | 6            | file is readable and writeable                              |
| 1        | 1        | 1        | 7            | file is readable, writeable, and executable                 |

Another approach to remembering how to compute a value would be to use the following relationship:

**r = 4**  
**w = 2**  
**x = 1**

To calculate the value for a required permission, simply add the values for r, w, and x to find the octal value.

#### 3.4.2.4 File Links

A file may in fact point to another file. The numeric value immediately following the permissions is an indicator of how many other file names point to, or are linked to the file in question. If the value is a “1”, then the file is unique and no other files link to it. This is observed as:

r w x    r w x    r w x **1**

#### 3.4.2.5 File Owner

The file’s owner immediately follows the link value. This is the username of the file’s owner, typically “root” for system files. If you create a new file as a user other than the administrator, then the file will have your username as the owner.

#### 3.4.2.6 File Group Name

A file has a group of users associated with it. Multiple users may be assigned to a group. When a user is created on a system, a new group with the same name as the username is also created. Other users may be assigned to this group. The name following the file’s owner is the group name. Normally it is the same as the file’s owner, but may be modified to a different group name.

#### 3.4.2.7 File Size

A file’s size in Bytes immediately follows the group name. A file’s size may be 0 Bytes – a file in name only, to whatever size is required to maintain the information.

---

<sup>1</sup> Remember, root can do anything, something, or nothing – even if it is wrong!

### 3.4.2.8 File Last Modification Date

Following the file size is the last date and time that the file was modified. This field includes both the date and time, unless the file was last modified more than one year previously, in which case the time is replaced with the year of modification.

### 3.4.2.9 File Name

Finally, at the end of the listing is the file's name. The name may be up to 256 characters long, and contain any alpha-numeric character and the characters “ . - \_ , & ”. Spaces may also be included in a filename, but it is strongly discouraged. If a filename should include a space, it must be accessed by enclosing the file name in double quotation marks ( “ ” ).

### 3.4.3 Listing Directories Only

If you desire to list only the directories contained within a directory, use the command:

```
ls -F | grep /
```

This is identical to the **ls** command with the option to add a special character on to the end of each file, which specifies what type of file it is. In this case, one searches for only those file that contain a “/”. Recall that a file name may not contain the “/” character.

### 3.4.4 Listing Hidden Files

When we do a listing of a directory contents, we have seen the listing of files, but there may be more. Sometimes we find it necessary to have files within a directory that are not visible, or to be able to hide contents from the common user.

If we perform the command:

```
ls -a
```

Now we might observe something like:

```
.      Desktop      .first_start_kde
..     anaconda-ks.cfg  .freeciv
```

Notice that some of the files start with a “.”. This is a special character that specifies that the file is “hidden”.

```
.  file      This is a file that is our directory contents. When we do a ls
               command, we read the “dot” file.
.. file      This is a file that specifies the path to the parent directory, or
               the directory immediately above this directory.
```

### 3.4.5 Listing File Inode

The way Linux accesses the hard drive is via a database record. Each record is for one file, and contains information for:

```
inode      Record identifier
```

|       |                             |
|-------|-----------------------------|
| drive | Hard Drive pointer          |
| block | File location on hard drive |

From this, we can refer to each file's location by its inode value. To learn the file's inode value, we issue the command:

```
ls -li
996028    hosts.class
996030    inode
1061178   lsd
996029    test3
```

Unless a hard drive is nearing capacity, files are maintained in a continuous format. This means that if a file is modified, it is written out to the hard drive in a new location that allows for a continuous string of information, and the old location is freed for future use, rather than fragmented as MS Windows does.

Linux does not maintain files on the hard drive in the same manner as Microsoft. When MS writes a file, it finds the first available cluster and starts writing the file. If the file is larger than that cluster, it finds the next available cluster and writes to it. Thus files become fragmented and one must periodically defragment the partition / drive.

Linux, using the ext2 or ext3 file system, maintains files in a continuous segment on the hard drive. Every time you save the file, it writes it to a new section, then deletes the old, thus freeing up the area. Only when a partition on the drive becomes over 90% full will Linux start to fragment files – at that time you need to add more drive capacity to your system.

### 3.4.6 Combined Commands

We may combine the above commands to enhance our results, such as:

```
ls -lia
```

This will provide a full listing of all files, normal and hidden, with all attributes, including the inode.

### 3.4.7 Additional Options

A few of the many additional options for the ls command are:

- f List files in order on the disk
- F Show directory contents with the file-type suffix of:
  - / directory
  - \* executable
  - @ links
- p Show file type, identical to “-F”, except that executables are not marked.

## 3.5 Editing Files

There are many times when we will need to edit or create a file. This will be one of the most difficult labs because it has many variations.

**NOTE:** *We commonly denote the CTRL key with the carrot (^). This means to hold down the CTRL key and then strike the specified key.*

To start with, there are a number of different text editors.

1. vi The most well known – and probably the hardest to learn.
2. emacs A powerful text editor without a command line.
3. joe A simple editor with no command line at the bottom.
4. nano Simple editor with a command line at the bottom of the screen. Not included in the standard Red Hat. It has replaced pico in the Fedora Core release.
5. pico Basically the same as nano, with a command line at the bottom of the screen. Included in Red Hat.
6. cat Primarily a utility to display a file, but can be used to input a short text or script file.
7. touch A utility to create an empty file.

There are many more, but only these will be reviewed here. After review, you should select the text editor of your choice. In general, the future labs will use either the pico or nano text editor. The student that wishes to really be a top notch administrator, will use the vi editor.

### 3.5.1 vi Text Editor

The primary editor in Unix and Linux, although cursed by all new users, is a program called **vi**. It is not a text editor or word processor as you may be use to when creating a document in MS Word. It can more appropriately called a character / line editor, providing you with a simple, yet rich set of commands to edit text lines. **VI** is best used for creating / modifying scripts such as those to configure Unix or Linux. You might like to compare it to MS Notepad, but at the command line level of the operating system rather than a windows mode.

The commands are not necessarily easy to remember – except with a lot of practice. This is hopefully a simple explanation of the basic commands to use. If you want to be a true Unix guru – learn vi!

**VI** operates in one of two modes, Command Mode and Text-Entry Mode.

**Command Mode** is the entry of operational commands to **vi**. These are used to move to a text position, change to text entry mode, find, modify, rearrange, delete, substitute, search, and open / save / close a file. Text is not entered into a file in this mode.

**Text Mode** is the actual entry of characters / text. To move to different locations of the script, the user either uses the arrow keys, or changes back to the Command Mode.

To open an existing file in **vi**, type **vi filename**. To create a new file from **vi**, just type **vi**.

At this point, you are automatically in the command mode. At any time you wish to return to the command mode, press the **ESC** key, if you hear the computer beep at you, you are already in the command mode. All changes to the text are made to the buffer (RAM) version of the document, not to the version on the drive. There is no automatic save in **vi**.

To quit **vi**, you must be in the command mode, although there is one indirect (quick) method of bypassing the command mode and performing a quick write and exit.

- :q** Exits after making no changes to the buffer exits the program. If changes to the text had not been previously changed, it will refuse to allow you to exit the program.
- :q!** Exits and abandons all changes to the buffer since it was last saved. If changes to the text exist in the buffer, they are lost.
- :wq** or **:x** or **ZZ**  
Writes the contents of the document in the buffer to the drive and exits the program immediately.

To backup your work while editing a document, you need to perform a file write. Backup is not performed automatically, but must be done manually. In order to perform the backup, you must be in the command mode - hit the ESC key and

- :w** Writes the buffer to the file that **vi** is editing.
- :w filename**  
Writes the buffer to a new file with name filename. If you want the file to have an extension (which is not required in Linux), you need to specify it, such as filename.txt. This command will not permit you to overwrite a different existing file.
- :w! filename**  
Writes the buffer to an existing file, causing the old file to be overwritten. All information in the old filename is lost.

While in the command mode, you can move the cursor around the text with the following commands.

#### **Arrow Keys**

Moves the cursor around the text as directed by the arrow key keystrokes.

#### **Page Up / Page Down**

Moves the cursor around the text one screen at a time.

#### **l or spacebar**

Moves the cursor one position to the right. Same as the **→**.

#### **+ or Return / Enter**

Moves the cursor to the beginning of the next line.

**j** Moves the cursor to the line directly below preserving the horizontal position.

**–** Moves the cursor to the beginning of the line directly preceding.

**k** Moves the cursor to the line directly above preserving the horizontal position.

- h-*** Moves the cursor one character to the left. Same as <-.
- 0 (zero)*** Moves the cursor to the beginning of the existing line.
- \$*** Moves the cursor to the end of the existing line.
- w*** Moves the cursor one word to the right.
- b*** Moves the cursor to the beginning of the existing word to which the cursor is at.
- e*** Moves the cursor to the end of the existing word to which the cursor is at.
- H*** Moves the cursor to the first line of the lines presently displayed on the screen.
- M*** Moves the cursor to the middle of the lines presently displayed on the screen.
- I*** Moves the cursor to the last line of the lines presently displayed on the screen.
- G*** Moves the cursor to the end of the buffer or last line of text.
- IG*** Moves the cursor to the beginning of the buffer or the first line of the text.

To add text, you must be in the input mode. There are six ways of changing to this mode:

- a*** Append. Adds text after the current cursor position, or at the beginning if you are creating a new document.
- i*** Insert. Adds text to the document prior to the existing cursor position, or at the beginning if you are creating a new document.
- o*** Creates a new line in the insert mode directly below the current cursor position.
- A*** Appends text at the end of the current line.
- I*** Inserts text at the beginning of the current line.
- O*** Creates a new line in the insert mode directly above the beginning of the current line.

To delete text, you issue the following commands from the command mode:

- x*** Deletes character immediately to the right of the cursor position.
- dw*** Deletes word immediately to the right of the cursor position.
- d\$*** Deletes rest of line from the cursor position to the right.
- D\$*** Deletes line from the cursor position to the left.
- dd*** Deletes complete line for which the cursor is at.

To search for text, you issue the following commands from the command mode:

- /string*** Searches from the present cursor position to the end of the buffer.
- ?string*** Searches from the present cursor position to the beginning of the buffer.
- n*** Searches again in the current direction, used to continue a search after having finding a match.



**N** Searches again in the reverse direction.

To change or replace text, you issue the following commands from the command mode:

**r** Replaces a single character.  
**R** Replaces a sequence of characters.  
**cw** Changes the current word, from the cursor position to the end of the word.  
**cb** Changes the current word, from the beginning of the word to the character before the cursor position.  
**c\$** Changes a line, from the cursor position to the end of the line.  
**cc** Changes the entire line.

If you are in the insert / append mode, then use of the arrow keys, delete and backspace keys will move the cursor around the text.

This has been an elementary discussion of the **vi** editor. You are referred to other sources for additional information.

### 3.5.2 emacs Text Editor

Emacs is a very powerful text editor written in the language of LISP. It has an extensive help facility, but assumes that you previously know what you are doing. Use the key-stroke **CTRL-h** to enter the Help facility. For a tutorial, use the key-stroke **CTRL-h t**.

Commands are issued via either Control Characters – holding the CTRL key and the specified action key or Alternate Characters – holding the ALT key and the specified action key.

To terminate emacs, two keystrokes are required, CTRL-x then CTRL-c.

Common Commands:

|  |   |
|--|---|
| CTRL-v   | Move forward one screen   |
| ALT-v  | Move backward one screen  |
| CTRL-l   | Clear screen and redisplay all text (note, lower case L, not 1) |
| CTRL-p   | Move cursor to previous line                                    |
| CTRL-b   | Move cursor back one column space                               |
| CTRL-n   | Move cursor down to next line                                   |
| CTRL-f   | Move cursor forward one column space                            |
| (Note that the arrow keys also move the cursor around the text.) |   |
| ALT-f  | Moves cursor forward one word                                   |
| ALT-b  | Moves cursor backward one word                                  |
| CTRL-a   | Moves cursor to beginning of line                               |
| CTRL-e   | Moves cursor to end of line                                     |
| ALT-a  | Moves cursor to beginning of sentence                           |
| ALT-e  | Moves cursor to end of sentence                                 |
| ALT-<  | Moves cursor to beginning of text                               |
| ALT->  | Moves cursor to end of text                                     |
| CTRL-v   | Moves cursor back to where it was previously                    |
| ALT-v  | Moves cursor back to where it was previously                    |
| CTRL-g   | Terminates a command that is taking too long                    |

|           |  |
|-----------|--|
| CTRL-d    | Delete character to right of cursor                |
| <DEL>     | Delete character to left of cursor                 |
| ALT-d     | Delete word to right of cursor                     |
| ALT-<DEL> | Delete word to left of cursor                      |
| CTRL-k    | Delete line from the cursor to the end of the line |
| ALT-k     | Delete text to the end of the current sentence     |

Additional commands are available; the reader is directed to the emacs tutorial.

By default, emacs is in the insert mode. Start typing and you will see it on the screen from where the cursor was located. To delete a character to the left, use the DELETE key. The backspace may perform the same action.

If text extends to the next line, a continuation character “/” will appear in the far right-hand column and the text will continue on the next line.

Although emacs is a very powerful yet simple editor, one can observe that you need to memorize a few basic commands prior to jumping in.

It is important to learn some of the basic commands used in emacs, because it is the view editor for the **info** command. The important commands to learn in order to navigate info are:

|          |                              |
|----------|------------------------------|
| <b>u</b> | Move up to the previous page |
| <b>p</b> | Move to the previous page    |
| <b>n</b> | Move to the next page        |
| <b>q</b> | Terminate the info session   |

### 3.5.3 **joe Text Editor**

Joe is a text editor that uses the same basic properties of (now Corel) WordPerfect. It is a series of control characters, which are different from vi or emacs, are often familiar to many users. In any event, you must memorize the different commands to fully use the editor to any benefit.

### 3.5.4 **nano and pico Text Editors**

Nano and pico are nearly identical programs. They are very easy to use and operate and provide 90 percent of the commands in a command window at the bottom of the screen. All commands are CTRL key based and may be initiated from directly within the text without exiting to a command mode. Because the commands are at the bottom of the screen, they are not listed here. For simplicity of learning Linux, this is the recommended editor, but for those that wish to be more proficient at using either Unix or Linux, you need to learn vi.

In a simplified statement, this is an extremely easy and very flexible text editor, requiring no significant memorization.

When Red Hat is installed in an “everything” mode, pico is automatically installed, it is part of the pine package. Installing pico after the installation may be difficult, as it is embedded in application. Nano may be located and downloaded from the Internet. Later versions of Red Hat and Fedora Core include nano.

Several of the options for nano and pico include (^ means to hold down the CTRL key and at the same time click on the specified key):

|                 |  |
|-----------------|--|
| <b>^Shift-6</b> | Begins to mark a section of text for cutting out text                        |
| <b>^C</b>       | Reports the current cursor position as line # and character #                |
| <b>^G</b>       | Opens pico help text   |
| <b>^J</b>       | Justifies the selected text, similar to wordwrap in a word processor         |
| <b>^K</b>       | Cuts the selected text   |
| <b>^O</b>       | Writes out, or saves, the current text in the memory buffer to the file      |
| <b>^R</b>       | Reads in text from a file and pastes the text at the current cursor position |
| <b>^T</b>       | Checks spelling  |
| <b>^U</b>       | Pastes the current line of text  |
| <b>^V</b>       | Scrolls one page down  |
| <b>^W</b>       | Whereis (allows you to search for a string of characters)                    |
| <b>^X</b>       | Exit (option to save file before exiting, if necessary)                      |
| <b>^Y</b>       | Scrolls one page up  |
| <b>^F</b>       | Moves the cursor one character to the right                                  |
| <b>^B</b>       | Moves the cursor one character to the left                                   |
| <b>^P</b>       | Moves the cursor one line up   |
| <b>^N</b>       | Moves the cursor to the next line  |
| <b>^E</b>       | Moves the cursor to the end of the current line                              |

### 3.5.5 cat

The display utility **cat** is able to create, but not edit, a new file using the syntax;

```
cat > filename
```

You will be at a blank line, where you may type in the information that you desire. To terminate the input and close the file, enter **CTRL – D (^D)** on a new line.

```
$ cat > testfile
```

```
This is the text that is entered.
```

```
^D
```

### 3.5.6 touch

There is one additional command for creating a new file but not for editing it. The command

```
$ touch filename
```

The new file is of zero length. If touch is used on an existing file, the file is not overwritten, but its creation data / modification dates are updated.

Additional options for touch include:

- a** Update access time only
- m** Update modification time only

***Note: For the remainder of this text, unless otherwise specified, it is assumed that the user has either nano or pico.***

### 3.6 Displaying a File's Contents

Knowing that a file exists or being able to edit is very important, but sometimes you just want to list it to the screen in order to observe its contents. There are two basic types of files, text and binary.

A Text file is basically a human readable file containing ASCII characters. These types of files may be general text, executable script files, database files, or other files that contain information.

A Binary file is machine code information. Virtually all are binary information, but when printed out to the screen might appear as hexadecimal characters. They are generally unreadable without special tools.

#### 3.6.1 Viewing Text Files

Text files may be printed to the screen with the use of several different tools. The common tools include:

1. cat
2. more
3. less
4. head
5. tail
6. tac
7. hexdump
8. od
9. xxd

Whenever we use the term **display**, we imply one of the above commands.

##### 3.6.1.1 cat

**Cat** stands for ***concatenate***. It provides two different features, display files and to append file-2 to the end of file-1.

To display a file to the screen, we issue the command

```
$ cat filename
```

This will display the total file on the screen, or at least the last 25 lines of the file. The problem is that if the file is greater than 25 lines, it will display the last 25 lines.

If you should desire to look at two files at the same time, you can append one file on to the second and look at both at the same time. As we have seen in a previous lab, this can also be used to produce a new third file that is the combination of the previous two. To concatenate two files, we issue the command

```
$ cat filename1 filename2
```

Options are available when using cat. Several of the more useful are:

- n Display line numbers
- E Display '\$' at the end of each line (represents "new line" character)

–v      Show non-printing characters

### 3.6.1.2      **more**

**more** allows us to view a file in segments starting at the top. To view a file, issue the command:

**\$ more filename**

The file will be displayed on the screen, starting at the top. To scroll down through the file, we either tap the spacebar to move down one line at a time, or the Page-Down key to move one screen at a time.

There is one small problem with this process, once we have scrolled past a given line and it has gone past the top of the screen, we can not go back and observe it again.

To terminate the screen display, hit the 'q' key.

### 3.6.1.3      **less**

**less** allows us to view a file in segments starting at the top. To view the file, issue the command:

**\$ less filename**

The file is displayed on the screen, starting at the top. To scroll down through the file, we either use the Down Arrow key or the Page-Down key to move one screen at a time. The extra benefit is that we can also scroll back towards the top by using the Up Arrow key or the Page Up key.

To terminate the screen display, hit the 'q' key.

Additional options to the less command include:

- ?      Show help menu
- I      Case insensitive search
- n      Display line numbers
- pword   Start viewing at the word "word"

Two different files may also be displayed using the less command, with the first file displayed, and the second displayed after clicking on the spacebar.

***Note: Which is your choice, more or less? You might think of this:***

***more does less than less, or less does more than more***

### 3.6.1.4      **head**

Sometimes we may only desire to observe the very beginning of a file, such as when we need to look at the header information of a script file. This command is **head**. To look at the top 10 lines of a file, we issue the command

**\$ head filename**

The top 10 lines are now displayed on the screen.

Additional options include:

- `-n X` Show the first X lines
- `-v` Verbose mode

### 3.6.1.5 tail

There are other times when we desire to observe the end of a file, maybe to view the last changes to the file. The command for this feature is **tail**. To view the end of the file, issue the command:

**\$ tail filename**

Now the last 10 lines are displayed on the screen.

Additional options include:

- `-X` Show the last X lines
- `-v` Verbose mode

### 3.6.1.6 tac

There are special situations where one might desire to list the end of a file in reverse order. Such a command is **tac**. To do this, issue the command:

**\$ tac filename**

This will display on the screen the first 24 lines of a file in reverse order. Note that “tac” is “cat” spelled backwards.

### 3.6.1.7 hexdump

A special utility for displaying a file is **hexdump**. This utility provides the ability to display a file in binary, octal, hexadecimal, or full character display. One of the most useful options is “-c”, which displays a file as a continuous string with a new line character inserted. The most useful syntax is:

**\$ hexdump -c filename** byte character display

An interesting application of this type of command would be to display all of the meta-characters (hidden) of a MS Word file.

### 3.6.1.8 od

A utility similar to hexdump is **od**. The most common options are:

- \$ od -ta filename** displays each byte as a character
- \$ od -tc filename** displays each byte as a ctype character

### 3.6.1.9 xxd

A utility to display a text file as both a numeric value and character (ASCII default) is **xxd**. The syntax is:

**\$ xxd filename**

One option is:

- `-h` Show help information

### 3.7 Adding Users

Because Unix and Linux are multi-user systems, we need to add users so that they may log on. There are two reasons for adding users.

The first reason is security. First off, as a user of the system, unless you need to perform some specific administrative action, you should not log on as **root**. The administrator has total power to perform any action – including the deletion of the operating system. A normal user is restricted from some actions and locations, hence the first level of security.

The second reason is to allow another user to log into your system under their own user identification (**username**). Being multi-user, it is best to have each user log in under their own username. In order to set up a new user, **two** commands are required: **useradd** and **passwd**.

#### 3.7.1 Adding a User

The command to add a new user is:

```
$ useradd username
```

Some variations of Linux may support the command **adduser**, which Red Hat does – your choice of use.

To add a new user John Doe with the username of **jdoe**, issue the command:

```
$ useradd jdoe
```

Note that it is general operating procedure to always add a username in lower case. If the username is in upper case, problems with other programs, namely sendmail, may exist.

The administrator may create a new user from any location within the directory structure, as the **useradd {username}** is a command in the environmental path. To view the existing users on a system, one must be in the **/etc** directory, looking at the **/etc/passwd** file.

Options include:

|           |   |
|-----------|---|
| <b>-c</b> | Set comment field for a user in the passwd file |
| <b>-e</b> | Set expire date                                 |
| <b>-f</b> | Set inactivity interval before expire           |
| <b>-g</b> | set initial login group                         |
| <b>-G</b> | Assign additional group membership              |
| <b>-k</b> | Specify home directory path                     |
| <b>-s</b> | Set the login shell                             |
| <b>-p</b> | Assign the password with the new user creation  |

#### 3.7.2 Assigning User Password

We have now added the user John Doe to the system, but the user John Doe is still not able to log onto the system – he does not have a password.

The command to add or modify a password for a user is

```
$ passwd {username}
```

To modify the password for **jdoe**, issue the command:

```
$ passwd jdoe
```

You will then be prompted to enter the new password, then to repeat it. Note that if you do not enter the {username}, by default it will modify the password for the current user – which is what you logged in as! Be cautious, you may end up modifying your own password rather than that of another user. Note that some of the options require the existence of the **/etc/shadow** file.

Options for the **passwd** command include:

- d** Delete user password
- f** Expire password and force user to change it at next login
- i X** Lock account after password has expired X days
- n X** Set minimum number of days before password expires
- s user** Display and set password attributes for specified user
- S** Display the account status
- u** Unlock account for specified user
- w X** Warn user X days prior to account expiration
- x X** Set maximum number of days before password expires

As an alternative to using the **passwd** command, an administrator may add a password directly when creating a new user, by using the “-p” option. The format of the command is:

```
$ useradd jdoe -p new-password
```

There are several conditions for a password. First, if the password is four or less characters long, it will not be accepted. With the use of the MD5 feature (enabled during installation), the password can be up to 256 characters. A password may contain a space (very few password crackers check for a space – making it an extremely improved password). Second, if the password is six characters long or less, you will receive a warning message saying that the password is too short – as the administrator you can ignore it if you desire. Third, you may receive a warning message that the password is based on a dictionary word, you may ignore it if you desire.

***Note: Again, if you do not add the username onto the end of the passwd command, the system will assume that you are changing the password for yourself (in this case, root). This is very important to recognize this – you can change the root’s password when you mean to change that of another user!***

***You must issue both the useradd and passwd commands for a new user in order to make the user active.***

After the user logs into the system and enters their password, they may change their password to one that is unique and secure to them. They do this by just entering the command **passwd**. Their username is not required.

When you, as the administrator, create a new user, several actions are performed. First as we just specified, the user was added to the system. Second, we created their own “home” directory.



The home directory for you, the administrator, is **/root**. The home directory for the user **jdoe** is **/home/jdoe**. If you change to the **/home** directory, you will now observe a new directory called **jdoe**. The user's home directory is created from a skeleton directory located in the **/etc** directory, this is the **/etc/skel**.

The administrator may create a new user from any location within the directory structure, as the **adduser {username}** is a command in the **environmental path**. To view the existing users on a system, one must be in the **/etc** directory, looking at the **/etc/passwd** file.

A normal user may modify their own password by issuing the command  
**\$ passwd**

Note that the username is not required when modifying one's own password although they must enter their current password before being allowed to change to a new password. There are some specific restrictions to the password:

1. Dictionary words are not accepted.
2. New password must be greater than 6 characters.
3. Simple passwords are not allowed.

The dictionary that **passwd** uses is quite extensive, so guessing a proper password may be difficult. The above restrictions do not apply to the root administrator.

When the administrator modifies another user's password, the syntax is:

**\$ passwd username**

If you do not specify the username, you will be changing the password for yourself – the root administrator.

### 3.7.3 User Home Directory

When you, as the administrator, create a new user, several actions are performed. First as we just specified, the user was added to the system. Second, we created their own "home" directory.

The home directory for the administrator is **/root**. The home directory for the user **jdoe** is **/home/jdoe**. If you change to the **/home** directory, you will now observe a new directory called **jdoe**. All information with regard to normal operation and configuration is maintained in the home directory, and in addition, user data files are stored in the directory. Other users are not allowed access to another user's home directory.

When displaying the prompt, the home directory may be shown as either username or tilde (~).

### 3.7.4 Deleting a User

A user's account may be deleted from a system by using the command:

**# userdel username**

Normally this does not delete the user's home directory. To also delete the user's home directory, issue the command:

```
# userdel -r username
```

***Note that only the administrator may issue the above command.***

### 3.8 Password File\_

At the top of the list concerning security of the system must be the password assigned to each user and how they are maintained.

We have previously set up a new user and their password. We will do this again, but this time we will provide additional information and monitor the changes.

Passwords are maintained in the **/etc/passwd** file. Although it has the same name as the command, this is a data file. We will look at this file to interpret what it means.

#### 3.8.1 Passwd Data Entry

First, if we add a new user with username `jdoe`, and display (`cat /etc/passwd | grep jdoe`) the `passwd` file, we will observe something like the following in the last line:

```
jdoe:x:502:502::/home/jdoe:/bin/bash
```

Before we proceed, lets issue another command to fill out the `passwd` entry for the comment field, which we commonly set to the user's full name. Issue the command:

```
$ usermod -c John_Doe jdoe
```

Alternatively might have used the command **chfn** that will allow us to change the user's name, workroom, office phone number, and home phone.

Now add a password to `jdoe`, such as **class**. If we now again go back and look at the `passwd` file, we will have something like:

```
jdoe:$1$af7qmRxq$hTtn8NT.cahi9rstnTING.:502:502:John_Doe:/home/jdoe:/bin/bash
```

When a normal user wishes to modify their password, the system will first request their current password before allowing the password to be changed. A user is not allowed to enter in their username as their password, although the administrator may.

```
# passwd jdoe
```

```
Enter current password: abcdefg
```

```
Enter new password: zyxwvutsr
```

```
Reenter new password: zyxwvutsr
```

```
Changing password for user jdoe
```

```
passwd: all authentication tokens updated successfully
```

Note that the password entry is not echoed to the monitor

Now lets look at the information in the `/etc/passwd` file (your display will differ for the password):

```
jdoe
```

user name

|                                       |   |
|---------------------------------------|---|
| \$1\$af7qmRxq\$hTtn8NT.cahi9rstnTING. | user encrypted password                 |
| 502 (1 <sup>st</sup> )                | userid                                  |
| 502 (2 <sup>nd</sup> )                | groupid                                 |
| John_Doe (note underscore)            | User comment field, usual the full name |
| /home/joe                             | user home directory                     |
| /bin/bash                             | user shell                              |

The last item, **/bin/bash**, is the user's default shell. This may be modified with the command:

```
# usermod -s [shell] username
```

For example, if the user dennis is to be changed to the bsh shell, the administrator would issue the command:

```
# usermod -s bsh dennis
```

All of the shells are maintained in the **/bin** directory. If you search the **/bin** directory, you will note that all shells end with the letters "sh".

The above values, particularly with the password, userid and groupid will probably be different. In the case of the Comment Field, if you have spaces in the name, then enclose the name in double quotes.

### 3.8.2 Improving Password Security

The **/etc/passwd** file is open to reading by everyone (list its attributes). That means that someone would be able to read the file and potentially copy the encrypted password, then decrypt it (with extreme difficulty!). Not a good thing.

To overcome this, a new level of security was enabled to delete the encrypted password from the **passwd** file and transfer them to a new file called **/etc/shadow**. This also allows us to add a few more security features. Note that this process is no longer available in newer distributions, as the shadow file is always created.

#### 3.8.2.1 Password File Conversion

To convert a system from just the **passwd** to the shadow file, we need to issue the command:

```
pwconv
```

This modifies the **passwd** file, deleting the encrypted password, and creates the file

```
/etc/shadow,
```

placing the encrypted password there and providing additional option fields.

Presuming that the command works correctly, the response to the command will be a new prompt.

Although the administrator may convert the password back to the **passwd** file and delete the shadow file, this process will not be reviewed.

Note that with the newer versions, the **/etc/shadow** file is automatically created during installation, no option exists for creating the file.

### 3.8.2.2 Shadow File Contents

The shadow file will have the form:

```
jdoe: $1$af7qmRxq$hTtn8NT.cahi9rstnTING.:11692:0:99999:7:::
```

It is an excellent procedure to force a user to change their password on a periodic basis. As an example, say we want to force a new password every 90 days. This can be set up by issuing the command:

```
chage -M 90 jdoe
```

The shadow file will now have the form:

```
jdoe: $1$af7qmRxq$hTtn8NT.cahi9rstnTING.:11754:0:90:7:b:b:b
```

We can observe that the 90 appears as the second to the last field, indicating how often the password must be modified. The format of this file can be read as:

|               |  |
|---------------|--|
| jdoe          | Login name   |
| \$1\$af7 .... | Encrypted password                                       |
| 11754         | Days since 1 January 1970 that password was last changed |
| 0             | Days before password may be changed                      |
| 90            | Days after which password must be changed                |
| 7             | Days before password is to expire that user is warned    |
| (blank)       | Days after password expires that account is disabled     |
| (blank)       | Days since 1 January 1970 that account is disabled       |
| (blank)       | Reserved for future use                                  |

Additional options to the chage command are:

|           |  |
|-----------|--|
| <b>-m</b> | Minimum number of days between password changes.   |
| <b>-M</b> | Maximum number of days between password changes.   |
| <b>-d</b> | Number of days since 1 January 1970 that the password was last changed.                            |
| <b>-I</b> | Number of days of inactivity after a password has expired before account is locked out of service. |
| <b>-E</b> | Date on which account will no longer be accessible and the account is locked out.                  |
| <b>-W</b> | Number of days of warning before a password change is required.                                    |

Although the administrator may remove the shadow file and move the passwords back to the passwd file, it is not recommended.

### 3.8.3 Random Password Generator

Oh well, you just can't think of a good password. There is a solution to that – issue the command:

```
mkpasswd
```

By default, this will create a new password of 9 characters in length. If you should want the password immediately applied to a user, then you need to specify the username following the command. Make sure that you copy the password down correctly, because it is encrypted in the passwd / shadow file. Just issuing the command by itself does not insert the encrypted password into the passwd (shadow) file, but options do allow this.

Say you want the password to be a specific length, contain at least specified number of lower case and upper case alpha characters, and include a specific number of digits. The format is then:

```
mkpasswd -l {N} -c {N} -C {N} -d {N}
```

where:

- l** specifies number of password (maximum 256)
- c** specifies number of lower case alpha characters
- C** specifies number of upper case alpha characters
- d** specifies number of numeric digits
- N** numeric value of the above options

There must be a space between the option and the numeric value.

```
$ mkpasswd -l 10 -c 3 -C 3 -d 2  
Usj3BcbQ5+
```

If the user issues the command:

```
$ mkpasswd jdoe  
Usj3BcbQ5
```

and then observes the **/etc/passwd** or **shadow** file, then one will observe the encrypted format of the “Usj3BcbQ5” password. Be careful doing this – you now have to remember the password. An option exists to automatically write the password to the **/etc/shadow** file, but is not shown here because it is too easy to forget.

### **3.9 Creating Groups**

In addition to individual users, Unix and Linux also support the creation of a group of users. A group may consist of individual usernames and other groups. This information is maintained in the **/etc/group** file.

When a new user is created, that user also has a new group created for the user. They will both have the same name, hence one must be sure of whether they are talking about the user or group. In general, we always talk about the user unless otherwise necessary.

A group name is also identified by a unique groupid. Numbers from 0 to 499 are reserved for system usage, 500 and above are used for new users.

#### **3.9.1 Creating a Group**

The command for creating a group is:

```
# groupadd group_name
```

Presuming that the command works correctly, the response will be a new prompt.

This appends an entry to the **group** file with the new **groupname**, **group\_password**, **groupid**, and **group-members** – which at this time is empty. This file is called **group** and is located in the **/etc** directory.

Although a password is available for each group, we normally do not implement it unless there is a security reason.

We must now modify the file that contains the groups to add the members. The command to modify an existing entry to the **group** file is:

```
# usermod -G group_name user_name
```

where:

|                   |  |
|-------------------|--|
| <b>group_name</b> | The group that a user is to be added to. |
| <b>user_name</b>  | The user's login id.                     |

Usage of this command allows only one user to be added to a group at a time. A more user-friendly method is to use the X GUI mode, where multiple users may be added to a group at one time.

```
# usermod -G jdoe root
# usermod -G jdoe john
# cat /etc/group | grep jdoe
jdoe:x:500:root,john
```

Some vendors of Linux may support the command “**addgroup**” as an alternative to **groupadd**, but Red Hat does not.

### 3.9.2 User Group Membership

To display what groups a specific user is a member of, issue the command:

```
$ groups username
```

For example:

```
$ groups jdoe
```

### 3.10 Piping Utility

It is sometimes to our advantage to have the output of one command inputted to another. This is called **piping**. The symbol that we use to represent this function is:

|

This is the shift key above the back-slash (\) or **whack**.

What happens is that the command response, which is normally outputted to the screen, is redirected to a second command, which requires an input. Our command line will have the form:

```
command-1 | command-2
```

The pipe ( | ) represents an output / input buffer between commands.

We will use this command feature often when we desire to scroll through a long output or search for a desired string, several which will be demonstrated during the exercises.

### **Command\_1 Output ® Inputed to Command\_2 ® Displayed Output**

Examples for using the pipe include:

**\$ cat passwd | grep ftp**

Search the password file for the lines that include “ftp”.

**\$ history | less**

Display the last 1000 command issued one screen at a time.

### **3.11 Grep Utility**

When we view a file, we sometimes need to search for a particular string or word. Listing a complete file, particularly if it is a long file, may be difficult and even subject to missing the text that we wish to find. There are many options to this command, and it is one of the most useful that we can utilize for obtaining specific information from a large amount of information. It may be used by itself or in conjunction with the **pipe (|)**. To do this we use the **Global Regular Express Parser**, otherwise known as **GREP**.

We can search a file for a particular string by using the command

**grep ‘search string’ filename**

The grep may be used as a stand-alone command or in conjunction with another command whose output is **piped** through to the grep command. This includes listing a file or command.

The command syntax for piping to grep is of the form:

**command | grep ‘search string’**

If we desire to use options, the format of the command is:

**grep [options] ‘search string’ filename**

Where a few of the options include (but not all of them):

- i** Ignore the case of the searched text
- n** Provide line numbers with the output to assist with reading of the text
- r** Recursive, search all files in the specified directory
- v** Invert-match, match to the inverse of that specified
- q** Stop after first match
- X** Show X lines preceding and trailing line of interest

As a quick example of some uses, assume that we have the file that contains the following (‘file’):

Now is the Time  
For all good men

To come to the aid  
Of their party.

Now:

```
grep time file           null, no matches
grep -i time file        Now is the Time
grep -n come file        3:To come to the aid
grep -r lab /lab          /lab/dirlist:-rw-rw-r-- 1 owner group date
                           time
                           labspell
grep -v aid file          Now is the Time
                           For all good men
                           Of their party.
```

A command that will be reviewed later is **history**, which lists out the last 1000 commands that have been issued. For now let's just use it in an example.

**history | grep ls**

Here each line of the output is scanned to see if it contains the phrase “ls”, if it does, then that line is displayed on the monitor.

If you do a listing of the active processes, `ps aux`, you will obtain a long listing of all processes that are presently in use. If you wish to search for just one, we can issue the command:

**ps aux | grep ftp**

and we obtain:

**root 20238 0.0 0.4 1728 592 pts/0 S 17:01 0:00 grep ftp**

There is another version of `grep`, called **egrep**. If doing a search of the manual (`man`), you will get the same page for either `grep` or `egrep`. Although `egrep` is an enhanced version, today, `grep` in fact implements the `egrep` command.

### **3.12 Director Utility**

The normal output of a command is to the “standard io”, or standard input / output. The standard output is the monitor and the standard input is the keyboard. There are very specific situations where we desire to have the output directed to another device, most commonly to a file.

The format of the command is:

**command > filename**

When this happens, a file is either created or written over.

Sometimes we desire to append some information to the file. We then use the command:

**command >> filename**



Now the new information is appended on to an existing file, or a new file is created.

We have previously used the director when we created a new file with the concatenate command.

If we issue the command:

|                          |  |
|--------------------------|--|
| <b>\$ cat &gt; notes</b> | Create a new file for stdin (keyboard) |
| input                    |  |
| <b>Now is the time</b>   |  |
| <b>^D</b>                |  |
| <b>\$ ls</b>             | List the contents of the directory     |
| <b>notes</b>             |  |
| <b>\$ cat notes</b>      | Display the contents of the notes file |
| <b>Now is the time</b>   |  |
| <b>\$</b>                |  |

Now lets add an additional line to our file notes.

```
$ cat >> notes
This is another line
^D
$ cat notes
Now is the time
This is another line
$
```

### **3.13 Copying Files**

Quite often it is necessary to copy a file, most commonly for backup purposes. This is generally a good practice when we advance to the level of configuring our system for server functions.

The format of the command is:

```
$ cp filename1 filename2
```

This copies the file filename1 in the present directory to a new file with the name of filename2 to the present directory. If you need, you may also specify the path of the filename if the original file and / or terminating file are located in another directory.

If you need to copy a file between directories, use the command specifying the full path:

```
$ cp /path/filename1 /path/filename2
```

Options include:

|           |                                   |
|-----------|-----------------------------------|
| <b>-a</b> | Preserve original file attributes |
| <b>-r</b> | Copy directories recursively      |
| <b>-v</b> | Verbose mode                      |

### **3.14 Moving and Renaming Files**

There are two situations where we either have a file in the wrong location, or we need to rename it. The command for each of these situations is the same.

The format for this function is:

**\$ mv /path1/filename1 /path/filename2**

Here the file filename1 located in the directory /path1 is moved and renamed to filename2 in directory /path2. The path is not required if you are in the file's directory.

Conditions of the command are:

Path1 and Path2 may be the same, but filename2 must be different from filename1.

**or**

Filename1 and Filename2 may be the same, but path2 must be different from path1.

Options include:

|           |              |
|-----------|--------------|
| <b>-b</b> | Make backups |
| <b>-f</b> | Force move   |
| <b>-v</b> | Verbose mode |

### **3.15 Creating and Deleting Directories**

It is often necessary for us to create and / or delete a directory.

#### **3.15.1 Creating a Directory**

The command to create a directory is:

**\$ mkdir /path/directoryname**

If the path is not specified, then the new directory is created immediately below our present location. If the absolute path is specified, it must be the full path name (there are exceptions to this, when you become more proficient).

Options include:

|           |  |
|-----------|--|
| <b>-p</b> | Create parent directory if it does not exist |
|-----------|--|

#### **3.15.2 Deleting a Directory**

The command to remove a directory is:

**\$ rmdir /path/directoryname**

Again, the path does not have to be specified, but the directory must be immediately below our present location. If you desire, you can remove a directory in another location by specifying the full path.

In order to remove a directory, the directory must be empty. If there is another file or directory within the directory that is to be removed, the command will fail. Consider this a protection to make sure you do not delete a whole directory that should not be deleted.

To remove the an empty parent directory, use the option:

**-p** Removes an empty parent directory

In order to remove a directory that has file or subdirectory contents, we need to use the command:

**\$ rm -r /path/directoryname**

This removes everything recursively. If there are contents in the directory, then the system will query you prior to removing the files.

### **3.16 Deleting Files**

No matter what the situation, we eventually get to the point that we need to delete files.

#### **3.16.1 rm**

The command to remove a file is:

**\$ rm filename**

The file is deleted! In the MS Windows system, if immediate action is taken, a deleted file may be recovered (MS changes the first character of the file name to a "\$"), but in the Unix / Linux system, the inode of the file is deleted and we therefore no longer have a reference to it. It can not be recovered.

As a small protection to the remove directory command, in older systems we can modify the command to

**\$ rm -i filename**

This will provide a confirmation to insure that you really want to remove the file. With the Linux kernel 2.2 and above, this is generally been modified to always ask for a confirmation.

Options for the removal of a file include:

**-f** Force removal with no warning  
**-v** Verbose mode

#### **3.16.2 shred**

Every once in a while, for security reasons, we need to delete a file and make sure the data is totally erased. To do this, we issue the command:

**\$ shred [options] filename**

Using the **rm** command to delete a file only deletes the file's inode value, not the actual file contents. If we use the **shred** command, then the file is written over with a random pattern to hide the contents. Some of the more useful options include:

**-f** Changes the permissions of a file in order to force the overwriting of the file  
**-n** Overwrite the file n times, default is 25  
**-v** Provide a verbose output display

- z** On the final write, write over the data with all zeros to provide a fully deleted file on the disk space
- help** Display a help screen and exit

Shred does not delete the file, just overwrite it with lots of garbage. After the file is shredded, issue the `rm` command to delete it.

### **3.17 Links**

Many applications need to utilize another application that is not in the same directory or in the common environmental `$PATH`. There is a method where we can reference another file or application with the technique called a **link**. Sometimes we desire just to use a more intuitive file name, for example there is the command to add another user `useradd`, but someone thought it was more obvious if the command were `adduser`. They therefore created a link from `adduser` to `useradd`.

A **link** is a pointer to another file. It might also be thought of as an alias name for another file. There are two types of links, hard and soft.

A **hard link** points to another file that is on the same hard drive partition.

A **soft link**, also known as a **symbolic link**, may point to another file that is located anywhere on your system. The file that is pointed to may be on the same partition, hard drive, different partition, or on a NFS partition (NFS will be explained later in another lab).

To create a hard link, we use the command

```
$ ln /path/real-filename new-filename
```

If the real file and new file are in the same directory, then the `/path` portion of the `real-filename` is not required. Remember, the `new-filename` must be on the same partition. As an example, say we have a file called `forecast`, and we desire to create a new file name that lists its contents. We would then issue the command

```
$ ln forecast weather
```

If we then do the attribute directory listing (`ls -l forecast weather`) we would have something like

```
-rwxr--r-- 2 root root 100 Jan 01 10:00 forecast
-rwxr--r-- 2 root root 100 Jan 01 10:00 weather
```

At first appearance, this appears to be two separate files, but there is another way to really see if they are the same. There is an option for listing the directory contents that lists a file's, called the **inode**. The inode number is a unique identifier used to identify the specific location on the hard drive that the file may be found. Thus if we issue the command `ls -li forecast weather`, we will obtain

```
5678 forecast 5678 weather
```

So we are able to observe that both files have the same inode, hence they are the same file. Both file names point to the same physical location on the hard drive.

If we should delete one filename (e.g. forecast), the file weather will still point to the existing file. In order to totally delete the file from the system, we must delete both file names.

To overcome the hard link problem of spanning partitions, a soft link, or **symbolic link** was created. The creation of the symbolic link is quite similar to the hard link with the addition of an option:

```
$ ln -s /path/real-filename new-filename
```

This forces the symbolic link. If we were to create the links as before with

```
$ ln -s forecast weather
```

Now doing an attribute listing, we obtain something like

```
-rwxr--r-- 1 root root 100 Jan 01 10:00 forecast
lrwxr--r-- 1 root root 5 Jan 01 10:00 weather -> forecast
```

We now observe that the file type is a link for the file weather. For a soft link, we do not need to check for the inode value.

As a normal operational issue, we more commonly create symbolic links since they can cross partitions.

Additional options for the creation of links include:

```
-b      Make backup
-d      Make hard links to directories (may only be implemented by
          superuser)
-v      Verbose mode
```

### **3.18 Shell Interpreters**

When we are at the Command Line Interface (CLI), we use a Shell that interprets our keystrokes, converting them into appropriate bits that the kernel can execute. Several different shells exist that make this process easier for us.

The CLI is indicated by a prompt. In general when you are using Linux, your bash prompt will appear as

```
[username@hostname present-directory] $
```

This specifies that your username on the host is presently at a specified directory. If it ends in a "#", then you are the root administrator. From this you are able to enter various commands.

The original shell was developed by Bourne for the Unix system back in the early 1970's. Since then others have developed alternatives. This may seem unnecessary, but in another perspective, this is one of the powers of the Unix / Linux concept – you can develop system controls that benefit you and are not dictated to by a single vendor.

Other common shells include the sh "shell", korn shell, tsh shell, csh shell, tsh shell, zsh shell, ash shell, pdksh the public domain korn shell, and the bash shell.

There are pros and cons to each which may be argued by those who use them. A few comments on some include:

**sh and korn** Some of the original shells that provide basic functions.

**csh** A shell that supports the development of C Language program code.

**bash** A shell that has been updated from the original sh and korn shell. By default, this is the shell that Linux and most other systems operate in.

The last entry of a user record in the /etc/passwd file specifies the shell that a user is logged on with. This may be changed with the **usermod** or **chsh** utility.

Although the other shells have been developed, it is left to the user to investigate them and evaluate if they desire to utilize a shell other than bash.

Bash will provide you better than 99 % of all of your requirements.

Common features of Bash include:

- Multi commands per line, separated by a “;”
- Command Execution
- **[ ]** matching of possible characters in a filename
- **|** Support of the pipe command
- **&** Execute command in the background
- **\*** Wildcard, match on any set of characters in filename
- **?** Wildcard, match on any single character in filename
- **>** Redirect standard output to a file, overwriting file or creating a new file
- **>!** Forces the overwriting of the file
- **<** Redirect standard input from a file or device to a program
- **>>** Redirect standard output and append to specified file
- **2>** Redirect standard error to a file
- **2>>** Redirect and append stand error codes to a file
- **2>&1** Redirect the standard error code to the standard output
- **>&** Redirect the standard error to a file or device
- **|&** Pipe the standard error as an input to another command

Bash also supports the creation and execution of scripts. Using the above symbols and various commands, one may write programs to perform various executions in a batch format. Creating script files will be covered in a later lab.

To change to another shell, you need only key in the shell name, and you will be presented with the shell’s prompt. Note that different shells have different prompt formats. To return to your original shell, key in “exit”.

### **3.19 System Help**

Quite often we need help in what we are attempting to do. To support this, Unix / Linux support several on line help systems. These include the online manual (**man**), information (**info**), and alternative commands that may be used (**apropos**).

### 3.19.1 Man Pages

The number one help agent is **man**. This provides the command syntax for the many commands and a basic explanation of how the command works. In nearly all cases this will answer your questions of what options are available and how the command should be formatted.

The format of command is:

**\$ man keyword**

The SYNOPSIS gives the syntax of the command, that is, how the command is formatted.

As an example, if we key in **\$ man cat**, we obtain:

```

CAT(1)           FSF           CAT(1)
NAME
    cat - concatenate files and print on the standard output
SYNOPSIS
    cat [OPTION] [FILE]...
DESCRIPTION
    Concatenate FILE(s), or standard input, to standard output.
A,    --show-all
        equivalent to -vET
-b,    --number-nonblank
        number nonblank output lines
-e     equivalent to -vE
-E,    --show-ends
        display $ at end of each line
-n,    --number
        number all output lines
-s,    --squeeze-blank
        never more than one single blank line
-t     equivalent to -vT
-T,    --show-tabs
        display TAB characters as ^I
-u     (ignored)
-v,    --show-nonprinting
        use ^ and M- notation, except for LFD and TAB
--help display this help and exit
--version
        output version information and exit

    With no FILE, or when FILE is -, read standard input.
AUTHOR
    Written by Torbjorn Granlund and Richard M. Stallman.
REPORTING BUGS
    Report bugs to <bug-textutils@gnu.org>.

COPYRIGHT
    Copyright © 2001 Free Software Foundation, Inc.
```

***This is free software; see the source for copying conditions.  
There is NO warranty; not even for MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE.***

### **SEE ALSO**

***The full documentation for cat is maintained as a Texinfo manual. If the info and cat programs are properly installed at your site, the command  
info cat  
should give you access to the complete manual.***

**cat (textutils) 2.0.14    March 2001**

**CAT(1)**

There are eight sections to the manual pages. Each is specialized to give more specific information, and in some cases an entry may exist in more than one section. The sections are:

1. Executable programs or Shell commands
2. System calls
3. Library calls
4. Special files
5. File formats
6. Games
7. Miscellaneous information
8. Maintenance commands

If you enter the direct keyword, the default section will be displayed. This is commonly section 1. You can force the reading of another section by using the format:

**\$ man -k keyword**

Where k represents the section of interest.

Additional options include:

- a            Show all matching man pages
- h            Show help
- k *string*   Search for 'string' in the short description
- w            Show path of man page

### **3.19.2      Info Pages**

In an abbreviated format, the **info** pages provide a basic list of options and syntax. It is typically much more detailed than the man pages when warranted; otherwise it is the same. Info is an updated version of man and is supported by GNU. Often, the info pages contain the same information as the man pages, but in newer info updates, the content may be considerably more comprehensive. Additionally, the info pages are formatted using the EMACS style editor, so one needs to know the basics of EMACS in order to navigate through the pages.

The format of the command is

**info keyword**



### 3.19.2.1 Navigating Through Info

Unfortunately, obtaining information from **info** may be difficult, as it features the ability to create a menu system. This allows one to catalog information. The problem is that one has to know how to navigate the menu system in addition to the normal navigation.

#### Basic Navigation Commands

| Key Stroke | Function                | Explanation   |
|------------|-------------------------|---|
| q          | Quit                    | Terminate the Info Program                              |
| h          | Tutorial Help           | Provides a tutorial of operation                        |
| ?          | Basic Command Reference | Basic Command Reference, which is not always intuitive. |

#### Menu Navigation Commands

| Key Stroke | Function                                   | Explanation  |
|------------|--|--|
| n          | Next Menu Node                             | Move the user to the next menu node. The next node is defined as the current node's next sibling in a hierarchy.   |
| p          | Previous Menu Node                         | Move the user to the previous menu node. The previous node is defined as the current node's parent node in the hierarchy.  |
| u          | Move to Parent Node                        | Go up to the parent node in the hierarchy.   |
| l          | Last Node Visited                          | Go to the last node visited. This is like the "back button" on a browser.  |
| b          | Beginning of Node                          | Go to the top of current mode. Why isn't this called t? Because t takes you to the top node in the file.   |
| t          | Top of Node                                | Go to the very top of the hierarchy.   |
| ENTER      | Follow the reference that the cursor is on | When you press Enter while the cursor is on a line that starts with an asterisk, if that line is later in the node than one that starts with an asterisk and is followed by a space and the word "menu", you will be transported to the referenced node. To come back from a referenced node, use the l (last node visited) command. |

**Scrolling Commands**

| Key Stroke    | Function              | Explanation   |
|---------------|-----------------------|---|
| Down Key      | Move down one line    |   |
| Up Key        | Move up one line      |   |
| Backspace Key | Move up several lines | Walks up and down the hierarchy in the reverse direction of the Space command |
| Space Bar     |                       |   |

**3.19.3 Apropos Pages**

The apropos pages provide a listing of other like commands. By issuing an **apropos** command, you can learn what other commands might also be helpful. If you are interested in additional commands to investigate, initiate a command search using **apropos**. The display will specify additional commands, and the section of the man page that needs to be investigated.

The format of the command is

**\$ apropos keyword**

Options include:

**-h** Show help messages

**3.20 Removable Devices and Mount Points**

Today we have high capacity removable drives available to us. These are commonly known as CDROM, ZIP Drives, and now the DVD drive. USB and Firewire drives are also new entrants to the high capacity drive options. The floppy drive is not normally included in this group if the floppy is formatted in the MSDOS format. It is included if formatted as Linux or Unix. These storage devices are considered temporary and are referred to as removable devices.

The contents of the drive are not accessible immediately as they are in MS Windows. In order to observe the contents of these drives, they must be “mounted” every time it is installed.

Access to these devices is normally in the **/mnt** or **/media** directory, with each device having its own previously created subdirectory. Each device may be mounted or unmounted from any other directory, with the one exception being the device’s own subdirectory. There are two ways of mounting a removable device – the hard way and the easy way.

**3.20.1 Mount Command**

The mount command takes on two modes, query and action. In the first instance is when we issue the command:

**\$ mount**

**/dev/hda6 on / type ext3 (rw)**

**none on /proc type proc (rw)**

```

usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/hda2 on /apps type ext3 (rw)
/dev/hda3 on /backup type ext3 (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)

```

### 3.20.2 Mount or Media Directory

Removable drives in Red Hat 9 and below are typically located in the directory called **/mnt**, located immediately below the root. For Fedora Core 3, the removable drives are located in the **/media** directory. Below this directory are the additional directories of **floppy**, **cdrom**, and **zip** (if installed). In the case of a zip drive, the actual name may be **zip100.0** or **zip250.0**. To mount either the cdrom or zip, we generally change to the **/mnt** directory, but may mount the drive from any location by using a path command.

To mount a device the hard way, the user needs to specify:

1. The device type from the **/dev** directory
2. The type of filesystem
3. Desired options
4. Mount point

As an example, if we mount the cdrom, we might issue the command:

```

mount -t iso9660 -r /dev/cdrom /mnt/cdrom      or
mount -t iso9660 -r /dev/cdrom /media/cdrom

```

If you should have a parallel port zip drive, it can be mounted by issuing the command:

```

mount -t vfat /dev/sda4 /mnt/zip                or
mount -t vfat /dev/sda4 /media/zip

```

To mount a device the easy way, given that the device driver has been previously designed into the installed Linux system, one only need to issue the command and device. This is common for floppy drives, cdroms, and zip drives if detected during installation. We need to issue the following commands:

```

cd /mnt      or      cd /media
mount cdrom  or      zip100.0  or      floppy (if Linux formatted)
cd cdrom     or      zip100.0  or      floppy (if Linux formatted)
ls

```

We may then observe the device's contents with the list command, change to the desired directory to copy files or install programs.

If the drive has an electronic eject button, it is disabled while the disk is mounted. A drive with a mechanical eject button will still work. Linux and Unix also support a command to eject a disk. The present version of Red Hat Linux supports the ejection of a drive while it is mounted (**eject cdrom**), by automatically un-mounting the drive and then ejecting it. Previous versions did

not. If the drive supports it, there is also the **eject -t cdrom**, which will close the cdrom caddy; note that all drives do not support this feature (specifically laptops).

Options include:

- r** Mount device in read-only mode
- v** Verbose mode

To manually remove a disk, we need to un-mount it. From the **/mnt** or **/media** directory, we issue the command

**umount cdrom** or **zip100.0** or **floppy** (if Linux)

Note the spelling of the un-mounting command – **umount**!

If you presently reside in a directory of the removable device, you are not able to mount or unmount it.

### **3.21 Background Processes**

From the CLI we often desire to perform multiple process at the same time, where one process can operate in the background while we work on the other in the foreground.

To start an application in the background, we enter the command at the CLI prompt as normal, but add the “&” character to the end.

We would typically use this to run an application that we do not need to access, or one that does not require user input. If we are using X Windows, this will allow us to run multiple applications from the same terminal window.

In order to bring a process back from the background, we must first learn its process id (pid). We issue the command **ps aux** to learn what the pid is for the desired process. We can then issue the command:

**\$ fg pid**

This then brings the process back to the front screen. If we should desire to transfer the process back to the background, the procedure is:

**\$ bg pid**

If you are currently in the middle of using a process, you can send it to the background by clicking the keys CTRL-Z (^Z).

Lets say we want to compile a long program that we have written. If we do this by issuing the command:

**\$ gcc c-program**

Our system now does its thing – and we are left sitting there waiting for it to finish. OK, so instead we issue the command to put it in the background by issuing the command:

**\$ gcc c-program &**

While the system is compiling the program, we can now continue doing other work.

### **3.22 Alternate Terminals**

You have been told that Linux (Unix) is a multi-user system. This means that more than one user may utilize the system at the same time. It also means that you may have multiple sessions simultaneously from the Command Line. The system is set up to support up to six sessions at one time. You can log in six times as the same user or use six different username's.

With each session, you must log in. To open another session, strike the keys **ALT-F{X}**. For example, to switch to the second terminal, strike ALT-F2. This works for terminal sessions F1 through F6.

After you have opened another session (such as F2) and logged in, you can switch back to the original session by again striking ALT-F1 – you will be back at your original location.

By using this technique, you can have six different process running at the same time; switching between them as you need.

To illustrate, you have logged in under your normal username, and you have your prompt [jdoe@system jdoe]\$. You now click the keys **ALT-F2** and you have:

**login:**

So the question arises – which terminal am I logged in under? To learn which terminal you are logged into, issue the command:

```
$ tty
tty0
```

The returned value is the virtual terminal that you are logged on to.

### **3.23 Installing Applications**

We have installed the full configuration that provides us with a long list of applications. But as you research the network, you will find one that you want, or there is a newer version. Now you want to install it. There are two options for installing applications, RPM, APT-GET or YUM, and installing from source code, also known as installing from tarballs.

#### **3.23.1 Red Hat Package Manager**

Here we focus on the installation of RPM packages, applications that have been compiled and prepared to specifications by Red Hat. To improve the installation, Red Hat created a process called **RPM**, or **Red Hat Package Manager**. When an application has been set up for RPM, installation is generally very straightforward. Caldera's Open Linux, Mandrake, and Suse also accept the RPM installation procedure.

There are many options available to using RPM, of which we will cover a subset. The rest you will need to refer to the man pages to obtain.

The standard command line for installing an application with RPM is:

```
rpm options rpm-package-name
```

The **options** are:

- U** Update the presently installed package with a newer version – or install package
- i** Install package
- e** Remove previously installed package
- qi** Display information for a previously installed package
- ql** Display a file list for an installed application
- qpi** Display information from an uninstalled RPM package
- qpl** Display a file list from an uninstalled RPM package
- h** Displays hash marks (#) to indicate installation progress
- v** Verbose output
- force**  
This is a special command that forces an installation even if it is already installed. It is sometimes required when you want to install over an existing installation, but the system will not allow you. It is used in conjunction with the **-i** option.

The **rpm-package-name** will generally have the form:  
**app\_name-version\_number-HW\_version.rpm**

This has a nice benefit – we can look at the application and see an obvious reference to the version number. You can now compare your installed version to a possible newer version off of the Internet.

You can also see if there are limitation to the hardware. This is noted in the Architecture\_version as:

- .i386** Will operate on a basic 386 or better computer
- .i586** Requires a Pentium computer and support from a kernal that is compiled for the Pentium processor – namely the Mandrake version

When installing an upgrade, the following **rpm** format is strongly recommend:  
**\$ rpm -Uvh package-name**

If you are interested in what is available in rpm format, go to RedHat's web page ([www.redhat.com](http://www.redhat.com)) and check out what is available. There are also a wide variety of applications available on the site [www.rpmfind.net](http://www.rpmfind.net).

### 3.23.2 Red Hat Updates

There are several means by which a later version of Red Hat (RH8 or later) may be updated.

Red Hat offers its own service, called **up2date**. This is fine after an initial installation for 60 days, after which there is an annual fee.

Ximian also offers an update service called **Red Carpet**. This too is free for approximately 60 days, after which there is again an annual fee.

A problem exists with rpm installations sometimes due to library dependencies. These are typically C libraries or other applications that are required for a program to operate. If the dependency is not available, then the

rpm installation will fail. To say the least, this is a very frustrating situation – one that apt-get and yum are able to overcome if the desired application has been integrated into the apt-get / yum process. There are several sites that support apt-get and yum, which one should research from the Linux Magazine article or on the web.

### 3.23.3 Update Managers

There are two alternatives to updating a system for standardized applications, Yum and Apt-Get. Both of these utilities provide the same function, although slightly differently. An RPM contains within its header, a list of all required dependencies. When using the rpm utility, the list is read, system checked, and if the dependency is not available, the installation fails. Using either Yum or Apt-Get, the process will download the required dependency, install it, then proceed with the package installation. The problem is that not all applications not included in the basic installation have been ported over to Yum or Apt-get.

#### 3.23.3.1 YUM

Yum, Yellow Dog Update Manager, was ported from the Yellow Dog distribution. The Fedora Core distribution has made Yum as the default update manager. The use of Yum is very easy. Starting with Fedora Core 3, a checksum of process is required. This must be set up the the very first time it is run.

To set up the initial checksum process, following process must be performed.

```
# cd /usr/share/rhn
# ls
RPM-GPG-KEY
RPM-GPG-KEY-fedora
RPM-GPG-KEY-fedora-test
(other files not shown)
# rpm --import /usr/share/rhn/RPM-GPG-KEY
# rpm --import /usr/share/rhn/RPM-GPG-KEY-fedora
```

This installs the required keys.

Yum has a default set of sites that are checked, but this may be enhanced with additional repository sites. Change to the **/etc/yum.conf** file to update specific requirements if necessary. Newer versions of Fedora Core have moved the updating files to the **/etc/yum.repo.d** directory. Here multiple files specify the sites that are to be accessed, and their directories. As a normal process, no action is required. Read the man yum.conf page for additional detail.

Commands that are available under Yum are:

|                        |  |
|------------------------|--|
| <b>update</b>          | This updates all packages previously installed on the system.          |
| <b>install package</b> | Installs specified package.  |
| <b>check-update</b>    | Checks to see if updates are required, but does not update the system. |
| <b>upgrade</b>         | Performs the same as update.   |
| <b>remove / erase</b>  | Removes the specified package.   |

|             |   |
|-------------|---|
| <b>list</b> | Displays information about a package.                           |
| <b>Info</b> | Displays a description and summary information about a package. |

As an example of doing an update, which is the most common action, issue the command:

```
# yum update
Gathering header information file(s) from server(s)
...

Is this ok [y/N]: y           You need to answer this query.
Getting samba-client-version-....
...
```

It takes a some time to gather information and perform the update. A lot information will flow past the screen.

### 3.23.3.2 Apt-Get

Apt-Get, Advanced Package Tool, is an alternative to yum, and must be downloaded and installed in order to utilize it. It functions nearly identically to yum, although it does not have the checksum feature.

The Linux distribution called Debian provides an excellent means of downloading updates and applications, called **apt-get**. This has been ported over to Red Hat for system updates, supported by the URL site **www.freshrpms.net**.<sup>2</sup> For a limited number of applications, they may also be installed using the apt-get utility.

From the **http://apt4rpm.sourceforge.net/** site, download the latest version of apt. Make sure you select the proper version of apt-get for the Red Hat / Fedora version that is operating. Install the apt utility:

```
# rpm -Uvh apt-*.rpm
```

Various repositories are available for downloading files from. These are maintained in the **/etc/source.list** file. Entries in the sources.list file appear like the following:

```
rpm http://apt.freshrpms.net redhat/{version}/en/i386 os updates freshrpms
```

By default, the file contains a number of repositories, but additional ones may be added.

You can now run the update for the system.

```
# apt-get update
```

To install a new package, one can use the following command:

```
# apt-get install package-name
```

Note that the version number does not need to be specified, the latest will be automatically installed.

<sup>2</sup> A Very Apropos apt, Linux Magazine, October 2003; by R. Scott Granneman; pp 24-28



### 3.23.4 Tarballs

Another way to install a program is to obtain the source code and then compile it. Typically the source code will come with several other files for the configuration and documentation. It would be inconvenient to download the individual files and to install them in the correct location. To this end, in a similar manner as the commonly used “zip” process, files may be collected together and then unpacked – with each being placed in the correct location. The process of collecting multiple files together is “tarring”, and they are typically compressed using one of several compression utilities – typically “gzip”. Hence this type of file is called a “**tarball**”.

The installation of a tarball requires a process to fully install. The typical procedure is to download and then compile the application. When downloading the tarball, save it to a directory that you wish to store files in, then it must be extracted, and finally compiled. Issue the following set of commands:

```
$ cd {where tarball file is stored}
$ tar -xvzf {tarball filename}
$ cd {new tarball file directory}
$ ./configure
$ make
$ su root
# make install
```

Notice that in the above example, the basic commands have been issued as a regular user, but the last command, “**make install**”, must be issued as the administrator. (The command to change, “su”, is explained in section 3.25.) After the command is complete, the application will be available for operation.

There is one alternative packing of a tarball, what was previously demonstrated a tarball that was packed using gzip; alternatively, an application might be packed using bzip2. In this case, the command to unpack the tarball is:

```
tar -xvjf {tarball filename}
```

## 3.24 File Location Utilities

No matter how well we remember, we will not be able to recall the location of the file that we created just last week. In MS Windows from the find GUI, we have the ability to locate a file. Within Unix and Linux at the Command Line, we have four commands to locate a file – **find**, **locate**, **whereis**, and **which**. Each utility provides a different type of information.

### 3.24.1 find

The first command is called **find**. It has several options that may at first be intimidating. The syntax of the utility is:

```
$ find -{option} filename option-criteria
```

The typical usage is:

```
$ find -name filename
```

The options are:

|                       |   |
|-----------------------|---|
| <b>-name pattern</b>  | Directory name or file name   |
| <b>-group name</b>    | Searches for files that have the specified group name   |
| <b>-size numeric</b>  | Searches for file with specified number of blocks, or if numeric is added after the number, for the size in characters  |
| <b>-mtime number</b>  | Searches for files modifies <u>number</u> of days ago   |
| <b>-newer pattern</b> | Searches for files AFTER the one matched by pattern.  |
| <b>-print</b>         | Outputs the results to the standard output (monitor). Normally not required with Red Hat. It is only necessary when the command is used in a script and the output must be directed to the output device. |
| <b>-type filetype</b> |   |
| b                     | Block device file   |
| c                     | Character device file   |
| d                     | Directory file  |
| f                     | Named pipes (fifo)  |
| l                     | Symbolic Links  |

Most commonly we desire to search for a file, so we use the “-name filename” option. The print option in Red Hat is not necessary, although some versions of Linux or Unix may require it.

The search will start at the present directory and search all sub directories. If you start at the root directory, this may take some time to complete the search. You may specify the starting directory by using the command format **find /path -name filename**.

### 3.24.2 locate

The **locate** utility is now a link to the newer **slocate** utility, which provides enhanced security from the former. It is an alternative to the **find** utility. It works on the concept of searching a database rather than the directory structure for the desired file location, and is therefore much faster. The paths searched are specified by the user's \$PATH. Commands that are not located in the \$PATH statement are not found. Note that the user's \$PATH may differ between users. Additionally, the database stores file attributes, so that it will not display files that a user normally does not have access to. The syntax of the locate utility is:

```
$ locate filename
```

In order to find the desired file, you may be required to update the database, using the syntax:

```
$ locate -u
```

Be prepared to wait while it updates the database. It may be wise to run the program in the background (locate -u &). Many other options are available, the user is referred to the manual pages for additional detail.

### 3.24.3 which

The **which** utility is designed to specify only the path to a (shell) utility. The syntax is:

```
$ which command-utility
```

An example would be:

```
$ which passwd  
/usr/bin/passwd
```

### 3.24.4 whereis

The **whereis** utility finishes up our list of file location utilities. It is very fast, using a technique of stripping part of the filename in order to broaden the search base, for files located in the list of standard Linux places. This means that if one installs or moves a file to a non-standard location, then the file will not be found.

One of the benefits of the **whereis** command is the ability to search for different types of files. The normal search is for files that have entries in a directory, the source is also available, and there is an entry in the manual pages. One can limit the search by excluding one or more of the above criteria.

You can search only for executables (binaries) [-b], manual pages [-m], or source code [-s]. Normally we want to find files that have all entries, but sometimes we may know that some are missing.

Usage of the command is:

```
whereis [-option] filename
```

As an example, we issue the command:

```
$ whereis passwd  
passwd:  usr/bin/passwd  /etc/passwd  
          /etc/passwd.OLD  /usr/share/man/man1/passwd.1.gz  
          /usr/share/man/man5/passwd.5.gz  
whereis -b passwd  
passwd:  /usr/bin/passwd  /etc/passwd  /etc/passwd.OLD
```

Additional options for whereis include:

|           |                                   |
|-----------|-----------------------------------|
| <b>-m</b> | Search for man page sections only |
| <b>-s</b> | Search for sources only           |
| <b>-u</b> | Search for unusual directories    |

### 3.24.6 glimpse

The command “**glimpse**” is not normally provided with the default installation of Linux, but is an excellent alternative to the find command. Glimpse stands for Global IMPLICIT SEArch. It supports a faster algorithm than that used by find, but the system needs to be periodically indexed.

Glimpseindex will index all text files for a specified directory. Several different indexes may be created to speed the search process.

Glimpse must be downloaded from the Internet, and may be found at [www.rpmfind.net](http://www.rpmfind.net). It is available in a rpm format for easy installation to Red Hat.

### **3.25 Switch User**

During most of the previous labs, we have always logged onto our system as the administrator. We did this because we needed to perform various duties as the administrator. Obviously, this a general security risk. What would be better would be for us to log on as our normal username, and then change to the administrator as necessary to perform the necessary, then to exit back to our username login. Here we address a process to allow us to log in under our normal username and then switch to another username (the next section will provide an even better method).

The ability to “switch user” is very simple. The command:

```
$ su root
```

will change the logged in user to the administrator root. In actuality, you do not need to type in the word “root” – it is the default user for the command. So to upgrade yourself from your normal username to root, just type in the command:

```
$ su
```

Likewise, you are able to change to any other username you want by issuing the command:

```
$ su username
```

```
[root@host root]# su jdoe  
[jdoe@host root]$
```

Of course, in all cases, after you have switched to another username, you will have to provide the appropriate password of the desired user – either for the root user or of the other user. The one exception to this is when you log in as the administrator, and then switch to another user. You are already at the highest authority.

To exit back to your normal username, type in ‘exit’ at the command prompt.

When one switches to another user, one does not inherit the other user’s environment – rather they maintain their original environment. If one does wish to inherit the new user’s environment (typically that of root), then we issue the command:

```
$ su -
```

```
# su - jdoe  
[jdoe@host jdoe]$
```

In this case, one can then operate with the full privileges of root as if we had logged in as the administrator. Additionally, if we use only the **su** command, then you will remain in the same directory from which you issued the command, whereas if you use the **su -** command, then you will be changed to the home directory of the administrator (/root).

Options include:

**-c *cmd*** Issue simple command only and return to exiting user

### **3.26 Switch User by Command**

As a normal procedure, one should not log on as the "root" administrator as was noted previously for the **su** command. At the same time, we need to perform the functions of the administrator. We have already investigated the **su** (switch user) command, that allows the user to switch to another user without having to log off, but this has a danger of remaining in that mode continuously. If you should walk away from your system, another person could walk up to your system and issue inappropriate commands.

The **sudo** allows one to log on to a system as a normal user, doing the regular tasks. Then when another user needs assistance, such as updating their forgotten password, you can **sudo** to the root administrator, do the command, and when complete you will automatically be back to your normal user status.

The discussion presented here is at the very basic level, and will be presented again in much more detail in the configuration of the **sudoers** file in Chapter 15.

An alternative to the "su" command is to provide the permitted user the ability to issue commands as another user on a one time basis. This adds a little more to the command structure, but significantly improves security. The command that provides this process is called **sudo**. In order to make it effective, the file **/etc/sudoers** must be edited to specify who is allowed to utilize the command.

**Sudo** is issued in front of each command that you wish to have take place. Then, according to specifications specified in the **/etc/sudoers** file, commands are automatically performed as the specified user level. If you are setup correctly as username, then when you issue the **sudo** command, you would perform the action as if you were the root user. The first time you issue the command, you will be required to authenticate yourself with your personal password (not the user you want to act as), but after that you may continuously issue commands without re-entering your password - within a limited time limit (default time of 5 minutes).

The format of the command is:

**\$ sudo [- option] command**

The options list is quite long, and is typically not needed. If required, one can view the options on either the **man** or **info** page. At this time we will not provide a detailed listing.

As an example, say you are logged in under your normal username, and you want to display the **/etc/shadow** file; assuming you have been given the rights of the root administrator in the **/etc/sudoers** file, you may then issue the command:

```
$ sudo less /etc/shadow
```

The shadow file will be displayed after you enter your logged in username password. Note that you only need to do this once (for a given period of time – after which you will have to re-enter your username password again).

The sudo function does not allow the user to do everything that say the root user could do. It does check your personal rights and will restrict the user from performing some acts. In these cases, one would either have to log on as the root administrator or use the **su** command.

Prior to being allowed to perform the above command, the root administrator must assign you the user rights in the **/etc/sudoers** file. This file provides many options to allow the administrator to specify users, groups, hosts, or networks various rights to perform a specified level of commands. We will investigate just a few here, but for further detail the user needs to refer to either the man or info page for the sudoers file.

The default setting for the file is:

```
root ALL=(ALL) ALL
```

It specifies that the root user on this system has the right to issue all commands, but no one else may do anything! We need to add users and privileges to our options. To edit the **/etc/sudoers** file, we must use the special editor command visudo as the root administrator. Do not edit the file with another editor because visudo has been designed to check for proper syntax of the lines prior to exiting the file. Note that visudo is a vi editor - you must know the basic commands of vi in order to properly edit the file.

The sudoers file is composed two types of entries: aliases (basically variables) and user specifications (which specify who may run what). Aliases provide the ability to group users into various classes for both access as individuals and hosts. The simplest method is to create a group called "wheels" (which should exist from the installation) and then assign the user to that group. So below the line above line, we would add the following:

```
%wheel ALL = (ALL) ALL (should already exist)
```

The "%" means that wheel is a group.

In Red Hat version 7.2, the above line exists as shown in the sudoers file. In later versions, this line is commented out and one must remove the "**#**" in front of the line in order to permit the wheel group to be active. Remember, that in order to be part of the wheel group, the administrator must issue the command:

```
# usermod -G wheel username
```

As an example, we will add you as your normal username to the sudoers file to be able to perform all commands as the root user. First we create the group "wheels" (should already exist in the **/etc/group** file) and then add our username to that group. The proper procedure for editing the sudoers file is to use the **visudo** command, we enter the above line directly below the "root" line. The

**visudo** command evokes the vi editor, recall that the commands needed to function in vi are:

- i** Change to the INSERT mode to enter text
- ESC** Change to the COMMAND mode to save and exit
- :** Specifies at the COMMAND line that a command will follow
- w** Write-Out, or SAVE the changes from the buffer to the hard drive file
- q** QUIT the vi editor
- q!** QUIT the vi editor without saving the buffer contents to the hard drive file
- wq** Write the buffer to the hard drive file and exit immediately

The first time you use sudo after logging on, you will be required to enter the password for the logged in user. After that, for that login session, you not have to enter the password again.

This provides the basic configuration for simple access on one system. You could set up a server with sudo features, then **ssh** (covered in Chapter 15) into it as another user, and finally **sudo** a command as the root user. This would provide excellent security for administering a server from a remote location while encrypting the information across the network.

We will cover the user permissions in more detail in Chapter 15, where the configuration of the sudoer file will be reviewed.

### **3.27 Starting X Windows**

All of this time we have been using the Command Line Interface – much to our other desires. The X Windows GUI interface is so much more interesting. You have probably already been using the X Windows system (in spite of what you have been directed to do), and that is OK. For those who have not yet investigated, we will add some points so that you can utilize the X Windows system as if you were at the Command Line.

Official, we need to say “X Windows”, but we will often error and just say Windows. The term “Windows” is not copywrited by another company, but it is commonly implied. It might be better to say “MS-Windows” when you wish to specify that particular system.

Many users desire to set up their system to automatically boot into the X Windows system, and for a desktop system, that is fine. But realize that using the X Windows mode requires a lot of processor power. Hence if we can avoid it, we can utilize that power for other processes. This is especially important if we want to set up a server that will be serving the Internet and have a lot of queries.

From the Command Line Interface, issue the command:

**\$ startx**

The X Windows will then start up. Because there is a lot of things being set up, it will take some time to get started – depending upon your processor speed. The two most common configurations are GNOME and KDE, but many others

are available. Linux allows you to switch between the more common Desktops. For those who are interested, they may also utilize less common Desktops. For now, we will limit our discussion to running the terminal mode of the Desktop so you can obtain the same results as if you were at the CLI.

At the bottom of the screen in on the Desktop, called the Taskbar, you will observe a small screen or monitor. This starts an “XTERM” session. Each time you click on the monitor, a new XTERM session will be started, so you may have multiple sessions running at the same time. By placing the cursor inside one of the XTERM sessions, that one will be active for you to issue commands into.

The first thing you should notice when you open the terminal is the prompt. It is exactly the same as when you were at the CLI prompt. At this time you may issue any of the commands that you have been so far, and you will obtain the same results.

When you are finished with the session, you may close the terminal by clicking on the small X in the upper right-hand corner.

To terminate the X Windows process, right-click on the Desktop background, bringing up a command window. At the bottom you will observe the “Logout” line. Clicking on this will close the X Windows and return you to the CLI.

There is one time when you may observe problems in running the X Windows session. This will normally occur when you are installing Linux or when you change to another terminal. A terminal (and the video card) are set up to support certain resolutions and to be able to operate at specific frequencies. If a monitor is unable to support these resolutions, some problems may develop.

If you have an old terminal, it may not be able to support the higher resolution – and the operation may burn up the monitor. (This can be verified by personal experience!) Newer monitors are “smarter” and will automatically shut down before being destroyed.

If you observe a monitor that is not working properly, you can terminate the X Windows session by entering the keystrokes “CTRL-ALT-BACKSPACE”. You will be returned to the CLI.

### **3.28 Printer Configuration<sup>3,4</sup>**

Although normally a Network Administration level, we will at this time provide a basic setup so we may utilize the printer in future labs. Setting up of a system as a print server is covered in Chapter 10. There are three basic methods of configuring the printer that are functionally identical. The first is while in the CLI mode, and the second is in the GUI mode. Other than access method, both the first and second modes use the same procedure to configure the printer setup and provide similar screens, other than one is text mode and the other is GUI. The third method, using the CUPS utility, is left to a later time.

The CLI mode is enacted by issuing the command:

**# printconf**

This brings up a text mode graphic, where one TABs through the various options to configure a printer setup.

<sup>3</sup> Guide to Linux Installation and Administration 2<sup>nd</sup> Ed., Nicholas Wells, Thompson

<sup>4</sup> The Complete Reference – Red Hat Linux 8, Richard Peterson, McGraw Hill / Osborne



The GUI mode is enacted by selecting:

**MENU (K or Big-Foot) – System – Print Configurator**

This will bring up a “nicer” graphic, but is in fact basically identical to the text mode. The benefit is that one can either use the TAB key or the mouse to select the desired function.

When either printconf or the GUI Print Configurator is finished, they produce a set of files that define the print configuration.

The first file is **/etc/printcap**. Do not edit this file because all modifications will be lost the next time the daemon is restarted. This file is regenerated every time the system is booted up. If special modifications are required, then the administrator needs to implement the changes through the configurator. Another file is also available to be used and is manually edited by the administrator for configuration of the desired printer – this is the **/etc/printcap.local** file. Initially, the printcap.local file is empty.

Assuming that the printer has been named “HP244”, the printcap file will appear something like the following:

```
# /etc/printcap
#
# DO NOT EDIT! MANUAL CHANGES WILL BE LOST!
# This file is autogenerated by printconf-backend during lpd init.
#
# Hand edited changes can be put in /etc/printcap.local, and will
# be included.

HP244:\
:sh:\
:ml=0:\
:mx=0:\
:sd=/var/spool/lpd/HP244:\
:af=/var/spool/lpd/HP244/HP244.acct:\
:lp=|/usr/share/printconf/util/jetdirectprint:\
:lpd_bounce=true:\
:if=/usr/share/printconf/util/mf_wrapper:

#####
## Everything below here is included verbatim from /etc/printcap.local ##
#####
# printcap.local
#
# This file is included by printconf's generated printcap,
# and can be used to specify custom hand edited printers.
```

The above file specifies basic system configuration and the location of other files. Two additional files provide configuration information for the specific printer.

The **/var/spool/lpd/HP244/mf.cfg** provides specific default configurations for the designated printer:

```
# foomatic/magicfilter configuration
# Make: HP
# Model: LaserJet 4050
# Printer Id: 62304
# Driver: Postscript
# TERMINATION=
# FILTER_LOCALE=C
```

```
define(MAKE, `HP')dnl
define(MODEL, `LaserJet 4050')dnl
define(COLOR, `false')dnl
define(PAGESize, `Letter')dnl
define(TEXTfilter, pipe/postscript/ /usr/bin/mpage -b ifdef(`PAGESize',
PAGESize, Letter) -l -o -P- -)dnl
define(PSfilter, `filter /usr/share/printconf/util/mf_postscript_wrapper
--mfomatic -d Postscript-62304.foo')dnl
define(PCLfilter, `cat')dnl
define(PJLfilter, `cat')dnl
define(DEFAULTfilter, `cat')dnl
```

and /var/spool/lpd/HP244/script.cfg provides the remote printer IP address and port number, if required:

```
printer_ip=192.168.102.11
port=9100
```

In the above files, all lines that start with a “#” are comments, and are therefore ignored. A double colon separates each field ( :: ). The total definition for a printer may be specified on a single line, but this would be harder to read, so the backslash ( \ ) is used to tell the OS to ignore the newline character (Line Feed – Carriage Return in Microsoft). Thus the coding of:

**HP244::sh::ml=0::mx=0 ...**

is the same as:

```
HP244:\
:sh:\
:ml=0:\
mx=0:\ ...
```

In the examples above, the designation of “HP244” is normally replaced by your printer name.

To understand what is happening, one needs to understand the mnemonics that are used. A short list consists of (see the man page for printcap for a full list):

|              |   |
|--------------|---|
| <b>HP244</b> | In this example, the name given o the printer.                                |
| <b>ab</b>    | Always print a banner, default is false.                                      |
| <b>af</b>    | Specifies the printer accounting file and location.                           |
| <b>ff</b>    | Define a remote system character (ff = '\f' ).                                |
| <b>fs</b>    | Issue a Form Feed at the beginning of a document.                             |
| <b>if</b>    | Print filter to be utilized when printing.                                    |
| <b>lp</b>    | Specifies the device interface.   |
| <b>ml</b>    | Minimum printable characters for printable check.                             |
| <b>mx</b>    | Specifies maximum file size (0=unlimited).                                    |
| <b>pc</b>    | Price per page for accounting purposes.                                       |
| <b>pl</b>    | Page length in lines, default is 66.  |
| <b>pw</b>    | Page width in characters, default is 80.                                      |
| <b>rg</b>    | Restricted group, only designated users may submit documents to this printer. |

|           |  |
|-----------|--|
| <b>rm</b> | Remote printer server / system hostname. Used when printers not directly attached. |
| <b>rp</b> | Remote printer name.   |
| <b>rs</b> | Restrict remote users from printing to this printer.                               |
| <b>sd</b> | Specifies the spool directory.   |
| <b>sh</b> | Suppress printing of a banner page containing the username.                        |
| <b>sb</b> | Output a short one line banner page, default is false.                             |
| <b>sc</b> | Suppress multiple copies of the same print job.                                    |

Note that the above discussion applies to a system that has not been configured using CUPS. Using CUPS implements a different method of configuring the printer, although the end results are the same – see below.

There are basically five configurations for setting up connectivity to a printer – local, Unix, Windows, Novell, and IP Network printer.

|                  |  |
|------------------|--|
| <b>Local</b>     | Printer attached to the parallel port (printer) of the PC. |
| <b>Unix</b>      | Printer attached to a remote Unix / Linux PC.              |
| <b>Windows</b>   | Printer attached to a remote MS Windows PC.                |
| <b>Novell</b>    | Printer attached to a Novell network.                      |
| <b>JetDirect</b> | Printer with an Ethernet NIC and assigned an IP address.   |

### 3.28.1 Local Printer

**Local Printer** — a printer attached directly to your computer through a parallel or USB port. Using either the CLI or GUI mode, use the following procedure:

1. Select **New**
2. Select **Local Printer**
3. Type in a printer name, which must start with either [a..z] or [A..Z]. The name should be descriptive to the user, but need not be exact. The name of the printer is not case sensitive. TAB to the **Next** button and click ENTER.
4. Select a printer vendor from the list by TABBING to **Custom** and selecting the desired printer. For instance, select key in “HP”, and a screen of vendors will appear. Select the vendor HP (in this example), then click ENTER to expand the list. Select the specific model that is attached. Again click ENTER to expand the print-type. If the printer model are not shown, the administrator might visit the site **www.linuxprinting.org** to see if a driver is available.
5. You typically wish to have a PostScript type print. Assuming this is available, select this option and then TAB to Next.
6. The final screen is presented that reviews all of the selected data. If correct, TAB to Finish and click ENTER.
7. You are now returned to the front screen, and you should see your printer listed across the top. TAB over to Test and click on ENTER.
8. The new page specifies that you are able to create a new test page. Clicking on YES will print out a test page.

You have now configured the printer.

### 3.28.2 Remote Unix Printer

**Unix Printer (lpd Spool)** — a printer attached to a different UNIX system that can be accessed over a TCP/IP network (or example, a printer attached to another Red Hat Linux system on your network).

1. Select **New**
2. Select **Unix Print Queue**
3. Type in a printer location – specifying a fully qualified pathname, which should be specified in the /etc/hosts file to designate the IP Address. The Queue must also be specified. The Queue is maintained on your local system, and might have the name such as {printer-name\_lpd}. TAB to the **Next** button and click ENTER.
4. Select a printer vendor from the list by and selecting the desired printer. For instance, select key in “HP”, and a screen of vendors will appear. Select the vendor HP (in this example), then click ENTER to expand the list. Select the specific model that is attached. Again click ENTER to expand the print-type.
5. You typically wish to have a PostScript type print. Assuming this is available, select this option and then TAB to Next.
6. The final screen is presented that reviews all of the selected data. If correct, TAB to Finish and click ENTER.
7. You are now returned to the front screen, and you should see your printer listed across the top. TAB over to Test and click on ENTER.
8. The new page specifies that you are able to create a new test page. Clicking on YES will print out a test page.

You have now configured the printer.

### 3.28.3 Remote Windows Printer

**Windows Printer (SMB Share)** — a printer attached to a different system that is sharing a printer over a SMB network (for example, a printer attached to a Microsoft Windows machine).

1. Select **New**
2. Select **Windows Print Queue**
3. Type in the following information:
  - a. Share Name
  - b. Host IP
  - c. Workgroup Name
  - d. User Name
  - e. Password of UserTAB to the **Next** button and click ENTER.
4. Select a printer vendor from the list by and selecting the desired printer. For instance, select key in “HP”, and a screen of vendors will appear. Select the vendor HP (in this example), then click ENTER to expand the list. Select the specific model that is attached. Again click ENTER to expand the print-type.

5. You typically wish to have a PostScript type print. Assuming this is available, select this option and then TAB to Next.
6. The final screen is presented that reviews all of the selected data. If correct, TAB to Finish and click ENTER.
7. You are now returned to the front screen, and you should see your printer listed across the top. TAB over to Test and click on ENTER.
8. The new page specifies that you are able to create a new test page. Clicking on YES will print out a test page.

You have now configured the printer.

#### 3.28.4 Remote Novel Printer

**Novell Printer (NCP Queue)** — a printer attached to a different system, which uses Novell's NetWare network technology.

1. Select **New**
2. Select **Novell Print Queue**
3. Type in the following information:
  - a. Share Name
  - b. Queue Name
  - c. User Name
  - d. Password of UserTAB to the **Next** button and click ENTER.
4. Select a printer vendor from the list by and selecting the desired printer. For instance, select key in "HP", and a screen of vendors will appear. Select the vendor HP (in this example), then click ENTER to expand the list. Select the specific model that is attached. Again click ENTER to expand the print-type.
5. You typically wish to have a PostScript type print. Assuming this is available, select this option and then TAB to Next.
6. The final screen is presented that reviews all of the selected data. If correct, TAB to Finish and click ENTER.
7. You are now returned to the front screen, and you should see your printer listed across the top. TAB over to Test and click on ENTER.
8. The new page specifies that you are able to create a new test page. Clicking on YES will print out a test page.

#### 3.28.5 Jet Direct

**JetDirect Printer** — a printer connected directly to the network instead of to a computer. If the printer is set up as a Jet Direct printer, you need to obtain the with an IP address from the network administrator. Configuration of the printer is from an X Windows Printer Configuration utility, which is accessible from the X Windows Start menu. This brings up a GUI configuration window.

1. Select **New**
2. Select **JetDirect Printer**
3. Type in the following information:
  - a. Remote Printer IP Address
  - b. Port Number – should be 9100 and not be modified

4. TAB to the **Next** button and click ENTER.
5. Select a printer vendor from the list by and selecting the desired printer. For instance, select key in “HP”, and a screen of vendors will appear. Select the vendor HP (in this example), then click ENTER to expand the list. Select the specific model that is attached. Again click ENTER to expand the print-type.
6. You typically wish to have a PostScript type print. Assuming this is available, select this option and then TAB to Next.
7. The final screen is presented that reviews all of the selected data. If correct, TAB to Finish and click ENTER.
8. You are now returned to the front screen, and you should see your printer listed across the top. TAB over to Test and click on ENTER.
9. The new page specifies that you are able to create a new test page. Clicking on YES will print out a test page.

### 3.28.6 Configuration Utilizing CUPS<sup>5</sup>

The third, and most recent, method of configuring a system is CUPS, or Common Unix Printing System. CUPS is pre-installed, and configured as the primary print configurator in Red Hat 8 and later, it requires installation on Red Hat 7.2.

To confirm that CUPS has been installed on the system, issue the command:

```
# rpm -qa | grep cups
```

#### 3.28.6.1 Switching to CUPS

If your system was previously using the LPRng configurator, then as administrator you must switch to the CUPS configurator. To determine if CUPS is selected, issue the command:

```
service lpd status
```

If the LPRng is operational then it needs to be terminated with the command:

```
service lpd stop
```

At this time we need to activate CUPS. This is implemented by the following mouse clicks from the desktop.

```
Menu
Extras
System Settings
Printer System Switcher
CUPS
and finally OK
```

After the above sequence, you see a message indicating that it has been successfully switched. Before the CUPS service has been fully operational, it must be reactivated. From the CLI (XTerm) you need to issue the command:

```
service cups restart
```

---

<sup>5</sup> Hello Linux, by Clyde Boom, Lancom Technologies

### 3.28.6.2 CUPS System Files for Printer

Now that the CUPS daemon is active, we can proceed to configure a printer. The configuration files are located in the `/etc/cups` directories.

Files that are utilized are:

|                     |  |
|---------------------|--|
| <b>cupsd.conf</b>   | Configures the cupsd daemon.   |
| <b>client.conf</b>  | Sets up the system as a server with respect to the client. It is heavily commented to assist with its configuration.   |
| <b>printer.conf</b> | Contains settings for printers that have been created, such as:<br>Printer Name<br>Default Printer<br>Location<br>Device Interface   |
| <b>ppd</b>          | Postscript Printer Definition file maintains settings that configure a printer for settings such as:<br>Resolution<br>Printer Tray<br>Paper Size<br>This file is located in the <code>/etc/cups/ppd</code> directory, select Configure Printer (not Modify Printer) from the Print menu. |

To maintain backward compatibility with LPRng, the `/etc/printcap` file is still maintained, although it no longer maintains the printer configuration, just the printer names.

An improvement from LPRng has been set up in the `/var/log/cups` directory, where errors are logged into the `error_log` file.

All files going to the printer are initially queued in the `/var/spool/cups` directory.

### 3.28.6.3 Configuring a Printer using CUPS

A printer is configured by either the administrator or a client that has been appropriate permissions. Additional administrative tasks are also able to be performed.

Administration of CUPS is accomplished via a web interface. To access the utility from the printer system, enter the URL of:

**localhost : 631**

The administrator may also access the utility via the desktop by clicking:

**Menu**  
**Extras**  
**System Tools**  
**CUPS Printer Configuration**

Using a browser, the CUPS configuration utility may also be accessed from a remote location by specifying the IP-Address : 631. The screen will display all configured printers and pending print jobs.

### 3.28.6.3.1 Adding a Printer

From the front screen, click on the Manage Printers tag. Directly below the list of printers is an **Add Printer** button. Click on this button in order to create a new printer definition. Perform the following process:

Manage Printers

Add Printer

Log in as the System Administrator, with password, if necessary

Complete the fields for:

Printer Name

Printer Location

Printer Description

From Device, select the printer device interface from the pull down menu (USB not supported)

From Make, select the Printer model from the pull down menu

Click on the newly created printer name, the printer specification appears

Click the **Print Test Page** button to generate a test page.

You now have a configured printer attached to the system.

## 3.29 Commands Used in this Chapter

|            |  |
|------------|--|
| ALT-F{X}   | Changes to an alternative terminal   |
| &          | Forces a process to operate in the background                                    |
| CTRL-Z     | Forces an active process to operate in the background                            |
| > director | Redirects the output of a command to a file                                      |
| pipe       | Directs the output of a command into a second command                            |
| adduser    | Creates a new user on a system   |
| apropos    | Searches for alternative words that should be investigated relative to a command |
| bg         | Lists processes that are in the background                                       |
| cat        | Displays a file or creates a new file from direct keyboard input                 |
| cd         | Change Directory   |
| chage      | Modifies password aging attributes   |
| cp         | Copies a file  |
| emacs      | A text editor  |
| find       | Searches the file system for a file  |
| fg         | Moves a file from the background to the foreground                               |
| glimpse    | Searches for command files   |
| grep       | Searches text for a specified text string  |
| groups     | Displays groups that a user is a member of                                       |
| groupadd   | Adds a new group   |
| head       | Displays the top 10 lines of a file  |
| hexdump    | Displays a file in binary, octal, hexadecimal, or ASCII format                   |



|          |   |
|----------|---|
| info     | Displays an abbreviated format of the man page  |
| less     | Displays a file, allowing one to scroll both up and down through the file                                     |
| ln       | Create a link to another file   |
| locate   | Searches a database for commands  |
| ls       | List a directory's contents   |
| man      | Displays a commands on-line manual  |
| mkdir    | Creates a new directory   |
| mkpasswd | Creates a proposed  |
| more     | Displays a file a page at a time, does not allow one to page back   |
| mount    | Attaches or temporarily installs a removable drive, such as a CDROM, ZIP Drive, or ext2 formatted Floppy      |
| mv       | Renames a file or relocates a file to another location  |
| od       | Displays each byte in a file as a character   |
| passwd   | Creates or modifies the password of a user  |
| pico     | A text editor   |
| pinfo    | A graphic format for displaying an abbreviated format of the man page   |
| ps       | Displays various process that are active  |
| pwconv   | Converts a system that is using the passwd file for maintaining the password to one that uses the shadow file |
| pwd      | Displays the Path to the Working Directory  |
| rm       | Deletes a file or directory   |
| rmdir    | Deletes an empty directory  |
| rpm      | Installs applications that have been pre-formatted in the rpm manner  |
| service  | Enables, disables, or checks status of a server daemon  |
| shred    | Deletes a file and over-writes file space to fully destroy the file's contents                                |
| startx   | Used to initiate the X Windows GUI mode   |
| su       | Used to switch from one user to another   |
| sudo     | Used to temporarily switch from one user to another on a per command basis                                    |
| tac      | Displays a file in reverse order  |
| tail     | Displays the last 10 lines of a file  |
| tar      | Compresses or de-compresses a tarball   |
| touch    | Creates a new file with zero contents   |
| tty      | Displays which terminal the user is logged into   |
| umount   | Un-mounts a previously mounted CDROM, Zip or ext2 Floppy  |
| useradd  | Adds a new user to the system   |
| userdel  | Deletes a user  |
| usermod  | Modifies various attributes of a user   |
| vi       | A text editor   |
| whereis  | Searches for a command  |
| which    | Specifies the path to a shell utility   |

xxd                Displays a file in both numeric value and as an ASCII character

### **3.30                Chapter Review Questions**

1.    What is the username of the administrator?
  - a.    admin
  - b.    root
  - c.    login
  - d.    prof
2.    What are the two options for logging off of a system?
  - a.    quit & logout
  - b.    logoff & exit
  - c.    logout & exit
  - d.    quit & exit
3.    What is the full command for powering down a host?
  - a.    halt
  - b.    quit
  - c.    shutdown –h now
  - d.    poweroff
4.    What is the function of the SWAP partition?
  - a.    User information
  - b.    Temporary RAM storage
  - c.    File storage
  - d.    System configuration
5.    What directory contains the Linux Kernel for starting the system?
  - a.    /proc
  - b.    /tftpboot
  - c.    /boot
  - d.    /etc
6.    What command is used to display the path for another command of a shell utility?
  - a.    find
  - b.    which
  - c.    whereis
  - d.    who
7.    Removable drives, such as the CDROM, are located in which directory?
  - a.    /proc
  - b.    /opt
  - c.    /dev
  - d.    /mnt

8. A file needs to be relocated because we created it in the wrong directory, what command is used?
  - a. ren
  - b. relocate
  - c. mv
  - d. move
9. We need the file “/root/test” in the /lab directory, and we are presently in the /var directory. What command is required?
  - a. Move test var
  - b. mv /root/test var
  - c. mv /root/test /var
  - d. mv /root/test /lab
10. You need to remove the password from the passwd file and place it in a more secure location. What command is used and where is the password stored?
  - a. pwconv
  - b. pwconf
  - c. rmpw
  - d. convpw
11. Which command can be used to create a file, taking input directly from the keyboard, and what is the command format?
  - a. cat | filename
  - b. less > filename
  - c. cat > filename
  - d. ess | filename
12. All commands issued on a Unix / Linux system are \_\_\_\_\_ sensitive.
  - a. case
  - b. character
  - c. name
  - d. file
13. Which text editor provides the commands at the bottom of the screen?
  - a. vi
  - b. ed
  - c. pico
  - d. emacs
14. You desire to add a comment to a user, including their full name. What command is used?
  - a. usermod “user-name”
  - b. usermod -c “user-name”
  - c. usermod -c “full-name” username
  - d. usermod -c username “full-name”

15. Before modifying a configuration file, you need to back it up. What would you do?
  - a. `copy oldfile newfile`
  - b. `rn oldfile newfile`
  - c. `mv oldfile newfile`
  - d. `cp oldfile newfile`
16. What commands are available to search for on-line information about other commands?
  - a. `man`
  - b. `info`
  - c. `apropos`
  - d. all of the above
17. What command is used to locate executable and manual pages for other command utilities?
  - a. `man`
  - b. `info`
  - c. `whereis`
  - d. `get`
18. You need to just change the password for the user jdoe, but are logged in under your normal username. What command would you use?
  - a. `sudo passwd jdoe`
  - b. `su passwd jdoe`
  - c. `passwd jdoe`
  - d. `passwd`
19. You need to view the contents of a CDROM, what command is used to make the CDROM readable?
  - a. `mnt cdrom`
  - b. `mount cdrom`
  - c. `mnt /cdrom`
  - d. `mount /mnt/cdrom`
20. While in your home directory, you need to create the directory /date in the /lab directory. What single command is used?
  - a. `mkdir /date`
  - b. `mkdir /lab`
  - c. `mkdir /lab/date`
  - d. `mkdir /lab-date`
21. You desire to create a totally random password, what command could be used to create one?
  - a. `makepasswd`
  - b. `crtpasswd`
  - c. `passwd`
  - d. `mkpasswd`

22. You need to create a file that a program will later access and write to. What command is used to create the file?
  - a. touch filename
  - b. mkfile filename
  - c. create filename
  - d. make filename
23. You need to display the full path to your working directory. What command is used?
  - a. display
  - b. path
  - c. pwd
  - d. list
24. If a user wishes to modify their password. What command and information must be used / supplied?
  - a. password username
  - b. passwd username
  - c. passwd / present password
  - d. present password
25. You need to direct the output of a command so that it is appended to another file. What would you do?
  - a. command > filename
  - b. command >> filename
  - c. command < filename
  - d. command <<filename
26. You wish to protect a file from someone mulishly removing it. How might the administrator provide a remote backup to this changing file?
  - a. cp file
  - b. ren file
  - c. mv file
  - d. ln file
27. You wish to perform a full file structure search for a specific file. What is the command required?
  - a. search filename
  - b. find -name filename
  - c. find filename
  - d. look filename
28. You are logged in under your normal username, but need to act as the administrator for a period of time. What command would you use to change you to the administrator status?
  - a. su
  - b. sudo
  - c. login
  - d. exit and login

29. A user has finished viewing the contents of a CDROM, what command is issued before the CD may be ejected?
  - a. `umount cdrom`
  - b. `umount cdrom`
  - c. `demount cdrom`
  - d. `quit cdrom`
30. In order to remove a directory using the `rmdir` command, what condition must exist?
  - a. Directory must be empty
  - b. Directory contents must be backed up
  - c. Directory may be directly deleted
  - d. Directory may not be deleted
31. You need to create a special group (`itadmin`) for users, which command would be used?
  - a. `grpmod itadmin`
  - b. `grpadd itadmin`
  - c. `groupadd itadmin`
  - d. `usermod itadmin`
32. You need to view the last 10 lines of a file. What command would be used?
  - a. `cat filename`
  - b. `tail filename`
  - c. `look filename`
  - d. `view filename`
33. You are presently in your home directory, and you need to list the contents of the `/var/log` directory without changing directories. What command is used?
  - a. `ls`
  - b. `ls var/log`
  - c. `ls log`
  - d. `ls /var/log`
34. When using `vi`, what keystrokes are required to return to the command mode, save and exit, all in one command?
  - a. `ESC – : q`
  - b. `ESC – : w q`
  - c. `ESC – w q`
  - d. `ESC - : q w`
35. The user `jsmith` has forgotten their password for a server. What command would you issue to give them a new password?
  - a. `password`
  - b. `password jsmith`
  - c. `passwd`
  - d. `passwd jsmith`

36. You need to search the file “text” for the string “Linux OS”. What command is issued?
- a. `find text “Linux OS”`
  - b. `find “Linux OS” text`
  - c. `grep text “Linux OS”`
  - d. `grep “Linux OS” text`
37. You need to delete the file “test.bk”. What command is used?
- a. `rm test.bk`
  - b. `rv test.bk`
  - c. `del test.bk`
  - d. `mv test.bk`
38. You need to install the application newapp-1.2.6-i386.rpm . What command would be required, providing a verbose output and install status?
- a. `install newapp-1.2.6`
  - b. `install newapp-1.2.6.i386.rpm`
  - c. `rpm newapp-1.2.6.i386.rpm`
  - d. `rpm -ivh newapp-1.2.6.i386.rpm`
39. If a directory is to be deleted and it has files in it that are no longer needed, what command is used?
- a. `rm -r ./directory`
  - b. `rm ./directory`
  - c. `del -r ./directory`
  - d. `del ./directory`
40. After you have created a group, you need to add the user jdoe to it. What command is used?
- a. `adduser -g user group`
  - b. `adduser -G group user`
  - c. `usermod -g user group`
  - d. `usermod -G group user`
41. You need to scroll through a file, being able to move back and forth, what command would be used?
- a. `cat filename`
  - b. `more filename`
  - c. `less filename`
  - d. `head filename`
42. Which command is used to display a file’s size?
- a. `dir`
  - b. `ls -i`
  - c. `ls -l`
  - d. `ls -a`

- 43. A new member of the IT team (jdoe) needs to administer a server, what must be done on that server?
  - a. `useradd jdoe`
  - b. `passwd jdoe`
  - c. `useradd, passwd`
  - d. `useradd jdoe, passwd jdoe`
- 44. What utility directs the output of one command to be directed into another command?
  - a. `>`
  - b. `>>`
  - c. `&`
  - d. `|`
- 45. A file has the following permissions, `-rw-r-----`. What is the octal representation of these permissions?
  - a. `320`
  - b. `540`
  - c. `640`
  - d. `750`
- 46. You need to compile a program that you have written, but wish to continue to work on other tasks on the same terminal. What process would you utilize?
  - a. Run compiler with `&`
  - b. Run compiler with `-B`
  - c. Run compiler
  - d. Not able to be done
- 47. Displaying the first 10 lines of a file requires what command?
  - a. `cat filename`
  - b. `head filename`
  - c. `tail filename`
  - d. `more filename`
- 48. Which text editor requires the user to enter into the insert / append mode prior to entering text?
  - a. `pico`
  - b. `emacs`
  - c. `vi`
  - d. `cat`
- 49. You desire to run multiple applications, allowing you to switch between them. How would you set up this environment?
  - a. Open different CLI window
  - b. Open alternative terminal
  - c. Log on as administrator
  - d. Linux supports only one user



50. What command would be required to list a file's owner, inode, and files that are hidden?
- a. ls
  - b. ls -i
  - c. ls -lia
  - d. ls -ia
51. Explain what the parent directory is.
- a. A directory immediately below the present location
  - b. A directory immediately above the present location
  - c. The top of the directory structure
  - d. A user's home directory
52. Explain what a child directory is.
- a. A directory immediately below the present location
  - b. A directory immediately above the present location
  - c. The top of the directory structure
  - d. A user's home directory
53. What is the command to change to the parent directory?
- a. cd
  - b. cd .
  - c. cd ..
  - d. cd -
54. What is the command format to change to a child directory?
- a. cd /directory
  - b. cd directory
  - c. mv /directory
  - d. mv directory
55. What is the command to change to one's home directory without specifying the full path?
- a. cd
  - b. cd .
  - c. cd ..
  - d. cd -
56. What command is used to delete a file and write over the disk space in order to insure that no one will be able to retrieve the contents?
- a. rm filename
  - b. rm -s filename
  - c. shred filename
  - d. del filename
57. What editor is used to modify the sudoer file?
- a. vi
  - b. pico
  - c. visudo
  - d. visu

- 58. What viewing editor is used to scan through the info manual pages?
  - a. vi
  - b. pico
  - c. emacs
  - d. joe
- 59. What password must be entered when the sudo command is used?
  - a. User's password
  - b. Administrator's password
  - c. Owner's password
  - d. Administrator's login name
- 60. After the creation of a new user, what command must be issued?
  - a. adduser username
  - b. usermod username
  - c. passwd username
  - d. shadow username

## Chapter Index

|                                   |      |                      |        |
|-----------------------------------|------|----------------------|--------|
| <b>A</b>                          |      | <b>CUPS</b>          | 70     |
| Adding a User                     | 31   | Activating           | 70     |
| Administrator                     | 5    | Adding a Printer     | 72     |
| Administrator Login               | 5    | Configuring Printer  | 71     |
| Administrator's Password          | 5    | Installed            | 70     |
| ALT-FX                            | 53   | Switching to         | 70     |
| Alternate Terminals               | 53   | System Files         | 71     |
| Alternative Directory \"Names\"   | 15   | <b>D</b>             |        |
| Apropos Pages                     | 50   | Deleting a Directory | 42     |
| Apt-get Installation Utility      | 56   | Deleting a User      | 33     |
| Assigning User Password           | 31   | Deleting Files       | 43     |
| <b>B</b>                          |      | Director Utility     | 40     |
| Background Processes              | 52   | Directory            |        |
| Block Device                      | 13   | Present              | 15     |
| <b>C</b>                          |      | / 9                  |        |
| Case Sensitivity                  | 16   | /bin                 | 9, 35  |
| cat                               | 27p. | /boot                | 9      |
| change directory                  |      | /dev                 | 10     |
| cd                                | 15   | /etc                 | 10     |
| cd -                              | 15   | /etc/cups/ppd        | 71     |
| cd .                              | 15   | /etc/rc.d            | 10     |
| cd ..                             | 15   | /etc/skel            | 33     |
| cd /path/dirname                  | 15   | /etc/X11             | 10     |
| Change Directory                  |      | /etc/yum.repo.d      | 55     |
| cd dirname                        | 15   | /home                | 10, 33 |
| Child Directory                   | 15   | /home/user           | 33     |
| CLI Prompt                        | 16   | /lib                 | 10     |
| Combined Commands                 | 21   | /lost+found          | 10     |
| Command Line Interface            | 5    | /media               | 10     |
| Copying Files                     | 41   | /media/cdrom         | 10     |
| Create File                       |      | /media/floppy        | 10     |
| cat                               | 27   | /mnt                 | 9, 51  |
| emacs                             | 25   | /mnt/cdrom           | 9      |
| joe                               | 26   | /mnt/floppy          | 10     |
| nano                              | 26   | /opt                 | 11     |
| pico                              | 26   | /proc                | 10     |
| touch                             | 27   | /root                | 10, 33 |
| vi 22                             |      | /sbin                | 10     |
| Creating a Directory              | 42   | /tmp                 | 10     |
| Creating a Group                  | 37   | /usr                 | 10     |
| Creating and Deleting Directories | 42   | /var                 | 11     |
| Creating Files                    | 21   | /var/log/cups        | 71     |
| Creating Groups                   | 37   | Directory Structure  | 7      |
| CTRL-ALT-BACKSPACE                | 64   | Directory Tree       | 11     |

|                                 |            |                              |        |
|---------------------------------|------------|------------------------------|--------|
| Display                         |            | Listing Attributes           | 17     |
| Displaying a File               | 28         | File Group Name              | 19     |
| Display Text                    |            | File Last Modification Date  | 20     |
| cat                             | 28         | File Links                   | 19     |
| head                            | 29         | File Location Utilities      | 57     |
| hexdump                         | 30         | File Name                    | 20     |
| less                            | 29         | File Owner                   | 19     |
| more                            | 29         | File Permission              |        |
| od                              | 30         | r 18                         |        |
| tac                             | 30         | w 18                         |        |
| tail                            | 30         | x 18                         |        |
| xxd                             | 30         | File Permission              |        |
| Displaying a File's Contents    | 28         | Group                        | 18     |
| Drive                           |            | Others                       | 18     |
| Designation                     | 7          | Owner                        | 18     |
| Designator                      | 7          | File Permissions             | 18     |
| Floppy                          | 14         | File Pointer (Link)          | 44     |
| Partition Mounting              | 8          | File Size                    | 19     |
| Partitions                      | 8          | File System Table            | 13     |
|                                 |            | File Type                    | 13, 17 |
|                                 | E          | d Directory                  | 18     |
| Editing Files                   | 21         | l Link                       | 18     |
| Editor                          |            | - Normal File                | 18     |
| emacs                           | 25         | Files                        |        |
| joe                             | 26         | Creating                     | 21     |
| nano                            | 26         | Editing                      | 21     |
| pico                            | 26         | find                         | 57     |
| vi 22                           |            | Floppy Drive                 |        |
| emacs                           | 25         | fd0                          | 14     |
| Environmental Path              | 33         | fstab file                   | 12     |
| exit                            | 6          | Block Device                 | 13     |
|                                 | F          | File Type                    | 13     |
| File                            |            | Mount Point                  | 13     |
| attributes                      | 17         | Fully Qualified Pathname     | 15     |
| permissions                     | 17         |                              | G      |
| Permissions                     | 18         | Getting Started - Logging On | 5      |
| /bin/bash                       | 35         | glimpse                      | 59     |
| /etc/fstab                      | 12p.       | Grep Utility                 | 39     |
| /etc/mtab                       | 12, 14     |                              | H      |
| /etc/passwd                     | 31, 34, 37 | Halt                         | 6      |
| /etc/printcap                   | 65, 71     | Hard Drive                   |        |
| /etc/shadow                     | 32, 35     | hda                          | 12     |
| /etc/source.list                | 56         | hdc                          | 12     |
| /etc/sudoers                    | 62         | Partition Designator         | 12     |
| /etc/yum.conf                   | 55         | Primary                      | 12     |
| /var/spool/lpd/HP244/mf.cfg     | 65         | Hard Drive Designation       | 12     |
| /var/spool/lpd/HP244/script.cfg | 66         | Hard Link                    | 44     |
| File Attributes                 |            | head                         | 29     |
| File Permissions                | 18         |                              |        |

|                                    |      |                                    |    |
|------------------------------------|------|------------------------------------|----|
| hexdump                            | 30   | Partitions                         | 11 |
| Improving Password Security        | 35   | passwd                             | 33 |
| Info Pages                         | 48   | passwd username                    | 33 |
| Installing RPM Applications        | 53   | Passwd Data Entry                  | 34 |
| Jet Direct                         | 69   | Password                           |    |
| joe                                | 26   | Echoing                            | 5  |
|                                    |      | username                           | 35 |
|                                    |      | Password File                      | 34 |
|                                    |      | Password File Conversion           | 35 |
| less                               | 29   | Path to Working Directory          | 16 |
| Link                               |      | Permissions                        | 18 |
| Hard                               | 44   | pico                               | 26 |
| Links                              | 44   | Piping Utility                     | 38 |
| Linux Directories                  | 9    | Poweroff                           | 6  |
| list                               | 17   | Printer Configuration              | 64 |
| Listing Directories Only           | 20   | Printer Setup                      |    |
| Listing Directory Contents         | 16   | Print Configurator                 | 65 |
| Listing File Attributes            | 17   | printconf                          | 64 |
| Listing File Inode                 | 20   |                                    |    |
| Listing Files                      | 16   | Random Password Generator          | 36 |
| Listing Hidden Files               | 20   | Reboot - CTL-ALT-DEL               | 6  |
| Local Printer                      | 67   | Rebooting                          | 6  |
| locate                             | 58   | Red Hat Package Manager            | 53 |
| Logical Partition                  | 7    | Red Hat Up2Date                    | 54 |
| Login and Password                 | 5    | Red Hat Updates                    | 54 |
| Logout                             | 6    | Remote Novel Printer               | 69 |
| ls                                 |      | Remote Unix Printer                | 68 |
| Additional Options                 | 21   | Remote Windows Printer             | 68 |
|                                    |      | Removable Devices and Mount Points |    |
| Man Pages                          | 47   |                                    | 50 |
| more                               | 29   | rm                                 | 43 |
| Mount Partition                    | 14   | root - Administrator Login         | 5  |
| Mount Point                        | 13p. | Root User                          | 5  |
| Mount Type                         | 14   |                                    |    |
| Moving and Renaming Files          | 42   | SCSI Drive                         | 12 |
| mtab file                          | 12   | sda                                | 12 |
| Access                             | 14   | sdc                                | 12 |
| Mount Partition                    | 14   | Search                             |    |
| Mount Point                        | 14   | string                             | 39 |
|                                    |      | service                            |    |
|                                    |      | lpd status                         | 70 |
| nano                               | 26   | Service                            |    |
| Navigating the Directory Structure | 15   | CUPS Restart                       | 70 |
|                                    |      | Shadow File Contents               | 36 |
| od                                 | 30   | Shell Interpreters                 | 45 |
|                                    |      | Shell Prompt                       | 45 |
| Package Managers - Alternative     | 55   | shred                              | 43 |
| Partition Management               | 12   | Shutdown                           | 6  |
| Partition Requirements             | 8    |                                    |    |

|                        |    |              |          |
|------------------------|----|--------------|----------|
| shutdown -h now        | 6  | less         | 29       |
| shutdown -r now        | 6  | more         | 29       |
| Soft Link              | 44 | od           | 30       |
| Starting X Windows     | 63 | tac          | 30       |
| SWAP                   | 9  | tail         | 30       |
| Switch User            | 60 | xxd          | 30       |
| Switch User by Command | 61 | eject        | 51       |
| Symbolic Link          | 44 | fg 52        |          |
| Synopsis               | 47 | find         | 57       |
| Syntax                 | 47 | fstab-sync   | 13       |
| System Architecture    | 54 | glimpseindex | 60       |
| System Help            | 46 | glimpse      | 59       |
|                        |    | grep         | 39       |
|                        |    | group        | 38       |
| tac                    | 30 | groupadd     | 37       |
| tail                   | 30 | head         | 29       |
| Tarballs               | 57 | hexdump      | 30       |
| Text Editor            |    | history      | 40       |
| cat                    | 27 | info         | 48       |
| emacs                  | 25 | inode        | 44       |
| joe                    | 26 | less         | 29       |
| nano                   | 26 | ln 44        |          |
| pico                   | 26 | ln -s        | 45       |
| touch                  | 27 | locate       | 58       |
| vi 22                  |    | ls -a        | 20       |
| Tilde                  | 16 | ls -d        | 20       |
| touch                  | 27 | ls -i        | 21       |
|                        |    | ls -l        | 17       |
|                        |    | ls -la       | 21       |
|                        |    | man          | 47       |
|                        |    | mkdir        | 42       |
|                        |    | mkpasswd     | 36       |
|                        |    | more         | 29       |
|                        |    | mount        | 14, 50p. |
|                        |    | mv           | 42       |
|                        |    | od           | 30       |
|                        |    | passwd       | 31       |
|                        |    | pwconv       | 35       |
|                        |    | pwd          | 16       |
|                        |    | rm           | 43       |
|                        |    | rmdir        | 42       |
|                        |    | rpm          | 53       |
|                        |    | shred        | 43       |
|                        |    | ssh          | 63       |
|                        |    | startx       | 63       |
|                        |    | su           | 60       |
|                        |    | sudo         | 61       |
|                        |    | tac          | 30       |

|                          |                |      |    |
|--------------------------|----------------|------|----|
| tail                     | 30             | /var | 11 |
| Text Editor              |                |      |    |
| cat                      | 27             |      |    |
| emacs                    | 25             |      |    |
| joe                      | 26             |      |    |
| nano                     | 26             |      |    |
| pico                     | 26             |      |    |
| touch                    | 27             |      |    |
| vi 22                    |                |      |    |
| tty                      | 53             |      |    |
| umount                   | 52             |      |    |
| useradd                  | 31             |      |    |
| useradd -p               | 32             |      |    |
| usermod                  | 34, 38, 46, 62 |      |    |
| whereis                  | 59             |      |    |
| which                    | 59             |      |    |
| xxd                      | 30             |      |    |
| Yum                      | 55             |      |    |
| > 40                     |                |      |    |
| > >                      | 40             |      |    |
| (Pipe)                   | 38             |      |    |
|                          |                | V    |    |
| vi 22                    |                |      |    |
| vi Editor                |                |      |    |
| Command Mode             | 22             |      |    |
| Text Mode                | 22             |      |    |
| Viewing Text Files       | 28             |      |    |
|                          |                | W    |    |
| Whack                    | 38             |      |    |
| whereis                  | 59             |      |    |
| which                    | 59             |      |    |
|                          |                | X    |    |
| X Windows Logout         | 64             |      |    |
| Ximian Red Carpet        | 54             |      |    |
| XTERM                    | 64             |      |    |
| xxd                      | 30             |      |    |
|                          |                | Y    |    |
| Yum Installation utility | 55             |      |    |
|                          |                | /    |    |
| / \“the root\”           | 9              |      |    |
| /boot                    | 9              |      |    |
| /home                    |                |      |    |
| Directory                | 11             |      |    |
| /media                   |                |      |    |
| Directory                | 51             |      |    |
| /opt                     |                |      |    |
| Directory                | 11             |      |    |
| /usr                     | 11             |      |    |

