CS 223 Linux Information Sheets

**LX0-101 Exam Objectives**

The Linux+ Powered by LPI Exam: LX0-101 exam covers the following topics (version 3.5.0).

|  |  |  |
| --- | --- | --- |
| **#** | **LX0-101 Exam Objectives** | **Module.Section** |
| **101** | **System Architecture** |  |
| 101.1 | Determine and configure hardware settings  Candidates should be able to determine and configure fundamental system hardware.   * Enable and disable integrated peripherals * Configure systems with or without external peripherals such as keyboards * Differentiate between the various types of mass storage devices * Set the correct hardware ID for different devices, especially the boot device * Know the differences between coldplug and hotplug devices * Determine hardware resources for devices * Tools and utilities to list various hardware information (e.g. lsusb, lspci, etc.) * Tools and utilities to manipulate USB devices * Conceptual understanding of sysfs, udev, hald, dbus   The following is a partial list of the used files, terms, and utilities:   * /sys * /proc * /dev * modprobe * lsmod * lspci * lsusb | 8.1 8.2 8.3 |
| 101.2 | Boot the system  Candidates should be able to guide the system through the booting process.   * Provide common commands to the boot loader and options to the kernel at boot time * Demonstrate knowledge of the boot sequence from BIOS to boot completion * Check boot events in the log file   The following is a partial list of the used files, terms and utilities:   * /var/log/messages * dmesg * BIOS * bootloader * kernel * init | 3.1 3.3 |
| 101.3 | Change runlevels and shutdown or reboot system Candidates should be able to manage the runlevel of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching runlevel and properly terminate processes. This objective also includes setting the default runlevel.   * Set the default runlevel. * Change between run levels including single user mode. * Shutdown and reboot from the command line. * Alert users before switching runlevels or other major system event. * Properly terminate processes. * Knowledge of basic features of systemd and Upstart.   The following is a partial list of the used files, terms and utilities:   * /etc/inittab * shutdown * init * /etc/init.d * telinit | 3.3 3.4 3.5 |
| **102** | **Linux Installation and Package Management** |  |
| 102.1 | Design hard disk layout Candidates should be able to design a disk partitioning scheme for a Linux system.   * Allocate filesystems and swap space to separate partitions or disks. * Tailor the design to the intended use of the system. * Ensure the /boot partition conforms to the hardware architecture requirements for booting. * Knowledge of basic features of LVM.   The following is a partial list of the used files, terms and utilities:   * / (root) filesystem * /var filesystem * /home filesystem * swap space * mount points * partitions | 2.1 2.2 7.1 |
| 102.2 | Install a boot manager Candidates should be able to select, install and configure a boot manager.   * Providing alternative boot locations and backup boot options. * Install and configure a boot loader such as GRUB Legacy. * Perform basic configuration changes for GRUB 2. * Interact with the boot loader.   The following is a partial list of the used files, terms and utilities:   * /boot/grub/menu.lst, grub.cfg and other variations. * grub-install * MBR * superblock | 3.2 |
| 102.3 | Manage shared libraries Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary.   * Identify shared libraries. * Identify the typical locations of system libraries. * Load shared libraries.   The following is a partial list of the used files, terms and utilities:   * ldd * ldconfig * /etc/ld.so.conf * LD\_LIBRARY\_PATH | 5.3 |
| 102.4 | Use Debian package management Candidates should be able to perform package management using the Debian package tools.   * Install, upgrade and uninstall Debian binary packages. * Find packages containing specific files or libraries which may or may not be installed. * Obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).   The following is a partial list of the used files, terms and utilities:   * /etc/apt/sources.list * dpkg * dpkg-reconfigure * apt-get * apt-cache * aptitude | 5.2 |
| 102.5 | Use RPM and YUM package management Candidates should be able to perform package management using RPM and YUM tools.   * Install, re-install, upgrade and remove packages using RPM and YUM. * Obtain information on RPM packages such as version, status, dependencies, integrity and signatures. * Determine what files a package provides, as well as find which package a specific file comes from.   The following is a partial list of the used files, terms and utilities:   * rpm * rpm2cpio * /etc/yum.conf * /etc/yum.repos.d/ * yum * yumdownloader | 5.1 |
| **103** | **GNU and Unix Commands** |  |
| 103.1 | Work on the command line Candidates should be able to interact with shells and commands using the command line. The objective assumes the bash shell.   * Use single shell commands and one line command sequences to perform basic tasks on the command line. * Use and modify the shell environment including defining, referencing and exporting environment variables. * Use and edit command history. * Invoke commands inside and outside the defined path.   The following is a partial list of the used files, terms and utilities:   * . * bash * echo * env * exec * export * pwd * set * unset * man * uname * history | 1.1 1.2 1.3 |
| 103.2 | Process text streams using filters Candidates should be able to apply filters to text streams.   * Send text files and output streams through text utility filters to modify the output using standard UNIX commands found in the GNU textutils package.   The following is a partial list of the used files, terms and utilities:   * cat * cut * expand * fmt * head * od * join * nl * paste * pr * sed * sort * split * tail * tr * unexpand * uniq * wc | 10.3 |
| 103.3 | Perform basic file management Candidates should be able to use the basic Linux commands to manage files and directories.   * Copy, move and remove files and directories individually. * Copy multiple files and directories recursively. * Remove files and directories recursively. * Use simple and advanced wildcard specifications in commands. * Using find to locate and act on files based on type, size, or time. * Usage of tar, cpio and dd.   The following is a partial list of the used files, terms and utilities:   * cp * find * mkdir * mv * ls * rm * rmdir * touch * tar * cpio * dd * file * gzip * gunzip * bzip2 * file globbing | 1.7 1.8 7.8 |
| 103.4 | Use streams, pipes and redirects Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.   * Redirecting standard input, standard output and standard error. * Pipe the output of one command to the input of another command. * Use the output of one command as arguments to another command. * Send output to both stdout and a file.   The following is a partial list of the used files, terms and utilities:   * tee * xargs | 1.6 |
| 103.5 | Create, monitor and kill processes Candidates should be able to perform basic process management.   * Run jobs in the foreground and background. * Signal a program to continue running after logout. * Monitor active processes. * Select and sort processes for display. * Send signals to processes.   The following is a partial list of the used files, terms and utilities:   * & * bg * fg * jobs * kill * nohup * ps * top * free * uptime * killall | 9.1 9.2 |
| 103.6 | Modify process execution priorities Candidates should be able to manage process execution priorities.   * Know the default priority of a job that is created. * Run a program with higher or lower priority than the default. * Change the priority of a running process.   The following is a partial list of the used files, terms and utilities:   * nice * ps * renice * top | 9.1 9.2 |
| 103.7 | Search text files using regular expressions Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.   * Create simple regular expressions containing several notational elements. * Use regular expression tools to perform searches through a filesystem or file content.   The following is a partial list of the used files, terms and utilities:   * grep * egrep * fgrep * sed * regex(7) | 1.11 |
| 103.8 | Perform basic file editing operations using vi Candidates should be able to edit text files using vi. This objective includes vi navigation, basic vi modes, inserting, editing, deleting, copying and finding text.   * Navigate a document using vi. * Use basic vi modes. * Insert, edit, delete, copy and find text.   The following is a partial list of the used files, terms and utilities:   * vi * /, ? * h,j,k,l * i, o, a * c, d, p, y, dd, yy * ZZ, :w!, :q!, :e! | 1.5 |
| **104** | **Devices, Linux Filesystems, Filesystem Hierarchy Standard** |  |
| 104.1 | Create partitions and filesystems Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions.   * Use various mkfs commands to set up partitions and create various filesystems such as:   + ext2/ext3/ext4   + xfs   + reiserfs v3   + vfat   The following is a partial list of the used files, terms and utilities:   * fdisk * mkfs * mkswap | 7.1 7.2 |
| 104.2 | Maintain the integrity of filesystems Candidates should be able to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.   * Verify the integrity of filesystems. * Monitor free space and inodes. * Repair simple filesystem problems.   The following is a partial list of the used files, terms and utilities:   * du * df * fsck * e2fsck * mke2fs * debugfs * dumpe2fs * tune2fs * xfs tools (such as xfs\_metadump and xfs\_info) | 7.4 |
| 104.3 | Control mounting and unmounting of filesystems Candidates should be able to configure the mounting of a filesystem.   * Manually mount and unmount filesystems. * Configure filesystem mounting on bootup. * Configure user mountable removable filesystems.   The following is a partial list of the used files, terms and utilities:   * /etc/fstab * /media * mount * umount | 7.3 |
| 104.4 | Manage disk quotas Candidates should be able to manage disk quotas for users.   * Set up a disk quota for a filesystem. * Edit, check and generate user quota reports.   The following is a partial list of the used files, terms and utilities:   * quota * edquota * repquota * quotaon | 7.5 |
| 104.5 | Manage file permissions and ownership Candidates should be able to control file access through the proper use of permissions and ownerships.   * Manage access permissions on regular and special files as well as directories. * Use access modes such as suid, sgid and the sticky bit to maintain security. * Know how to change the file creation mask. * Use the group field to grant file access to group members.   The following is a partial list of the used files, terms and utilities:   * chmod * umask * chown * chgrp | 7.6 7.7 |
| 104.6 | Create and change hard and symbolic links Candidates should be able to create and manage hard and symbolic links to a file.   * Create links. * Identify hard and/or softlinks. * Copying versus linking files. * Use links to support system administration tasks.   The following is a partial list of the used files, terms and utilities:   * ln | 1.9 |
| 104.7 | Find system files and place files in the correct location Candidates should be thoroughly familiar with the Filesystem Hierarchy Standard (FHS), including typical file locations and directory classifications.   * Understand the correct locations of files under the FHS. * Find files and commands on a Linux system. * Know the location and purpose of important files and directories as defined in the FHS.   The following is a partial list of the used files, terms and utilities:   * find * locate * updatedb * whereis * which * type * /etc/updatedb.conf | 1.10 1.11 |

**Shell Facts**

The Linux shell is the Command Line Interface (CLI) or Text User Interface (TUI) that administrators use to control a Linux operating system. Users and programs use the shell to send commands to the system. A shell might be opened inside a Graphical User Interface (GUI), or might be the sole method used to run the computer.

Although most Linux distributions now include a graphical interface, and many administration tools have been converted to a graphical format, many tasks are best performed from the command prompt. In addition, while graphical elements vary between distributions, shell commands are more likely to be consistent between distributions.

The following table describes many common shell types:

|  |  |
| --- | --- |
| **Shell Type** | **Description** |
| bash | The Bourne-again shell (bash) is the standard shell used in most Linux computers. It uses commands similar to a UNIX shell. Bash includes features such as:   * Command completion when pressing the tab key * Command history * Improved arithmetic functions |
| sh | The Bourne shell is an earlier version of the Bash shell, and is similar in many ways. **Sh**is the original shell created by Steve Bourne. |
| ksh | The Korn shell was developed by David Korn. Ksh has scripting features not found in bash. |
| csh | The C-shell uses syntax similar to syntax used in the C programming language. |
| tcsh | The tcsh shell is an improved version of csh. It offers command line editing and completion features not available with csh. |

Despite their differences, all shells share some common characteristics:

* A Linux system can use multiple shells at the same time.
* A list of shells is stored in the **/etc/shells** file.
* All shells are interfaces with the kernel, separate and distinct from it.
* Shells are run both interactively by end users and automatically by the computer's processes.
* Shells can run within one another either interactively when a user starts a second shell from the first shell's command line, or automatically by scripts or programs.
* Shells use configuration files to establish their operating environments.

**Shell Configuration Facts**

Shell configuration files are scripts that execute when a shell starts. The shell type determines which shell configuration files are executed. Shell types include:

* *Login shells* run when the system starts and is only using the Text User Interface (TUI) as the user interface.
* *Non-login shells*run when the system boots into a Graphical User Interface (GUI) and a user starts a terminal session.

 The following are the names of the files used when the shell starts:

|  |  |  |
| --- | --- | --- |
| **Configuration File** | **Description** | **Used by shell type** |
| **~/.bashrc** | **~/.bashrc**stores shell preferences for individual users. | non-login (login on some distributions) |
| **/etc/profile** | **/etc/profile**stores system-wide configuration commands and is used primarily to set environment variables. | login |
| **~/.bash\_profile** | **~/.bash\_profile**stores shell preferences for individual users. | login |
| **~/.bash\_login** | **~/.bash\_login**stores commands that execute when a user logs in. | login |
| **~/.profile** | **~/.profile**stores configuration preferences similar to **/etc/profile**, but for individual users. | login |
| **~/.bash\_logout** | **~/.bash\_logout**stores commands that execute when a user logs out. | login |

Be aware of the following:

* Login shells execute the configuration scripts they use in the following order:
  1. **/etc/profile**
  2. **~/.bash\_profile**(If this file is found, the shell does not look for additional configuration script files)
  3. **~/.bash\_login**(If this file is found, the shell does not look for additional configuration script files)
  4. **~/.profile**(This file only executes in the absence of the preceding two)
* The **su -l** command switches to a user into a login shell; however, without the**-l** option, a non-login shell is started.

**Help Facts**

Help pages are part of every Linux distribution. They provide information about options and uses for the nearly 1000 commands that are available on a Linux computer. The following table describes help options available for a Linux system.

|  |  |  |
| --- | --- | --- |
| **Option** | **Description** | **Examples** |
| Man page | A *manual* (man) page is text-based help file for a specific command stored on the computer. A man page shows the command's syntax, options, and related files and commands. Be aware of the following details:   * The **man** command opens the corresponding file in the command prompt window using the default paging program. * Man pages are typically stored in the**/usr/man** or **/usr/share/man** directory. Subdirectories store man pages for different types of commands or languages. * The MANPATH environment variable can be altered to specify a different location for man pages. * Some distributions set this variable using the **/etc/man.config** script, with corresponding scripts in each user's home directory to allow variation for individual users. * Press Q to exit the man page. * Use the **-k** option to search the man pages for a command. | **man userdel** shows the man pages for the **userdel** command. **man man** shows the man pages for the**man** command. **man -k user** lists all the commands with*user* in the command name or description. |
| Info pages | For GNU software, info pages are the primary documentation source. Info pages use hypertext links to navigate the pages. Use the following keys to navigate through an info page:   * *h* shows a navigation help screen. * *Tab* moves to the next hyperlink. * *Home* moves to the beginning of a node. * *Enter* follows the selected hyperlink. * *u* moves up a node level. * *Space* moves to the next screen. * *Del* moves to the previous screen. * *q* exits an info page. | **info mkfifo** opens the info page for the**mkfifo** command. **info man** opens the info page for the**man** command. |
| On-screen help | Many commands include on-screen help as one of the options for the command. In most cases, help displays an abbreviated list that shows of the command syntax and available options with brief descriptions. If the command is typed incorrectly, many commands display the help information automatically.  To view the help available for a command, type*:*   * ***command* --help** * ***command*** **-h**   Although some commands support both switches, the **--help** switch is more common. If necessary, use the **more** command to scroll through the on-screen help. | **jobs --help** for help with the **jobs**command. **echo -h** for help with the **echo**command. |
| whatis database | The whatis database is an index of the man pages on the system. Use the whatis database to find man pages that contain a specific word (i.e., search string).   * Run**/usr/sbin/makewhatis** to create the whatis database. The database must be created first. * Use **makewhatis -u**to update the database after changes have been made to the man pages.   The following commands will return a list of the man pages containing the search string in the whatis database:   * **apropos** *searchstring* * **whatis** *searchstring* | **makewhatis -u** updates the database. **apropos** **grep**returns all man pages containing the term *grep*. |

**Environment Variable Facts**

An *environment variable* is a setting that the operating system or programs working in the operating system access. Environment variables make up the user environment.  Be aware of the following details:

* The standard for writing variables names (called *variable identifiers*) is to use upper case (e.g., SHELL and EUID)
* Changing environmental variables from the defaults result in *user-defined* variables.
* Auser-defined variable applies only to the current session; export the user-defined variables so they apply to child sessions.
* Add user-defined variables to the shell configuration files to make them persistent.

The table below lists common environment variables:

|  |  |
| --- | --- |
| **Variable** | **Description** |
| BASH | The location of the bash executable file |
| SHELL | The user's login shell. |
| CPU | The type of CPU. |
| DISPLAY | Location where X Windows output goes. |
| ENV | The location of the configuration file for the current shell. |
| EUID | The ID number of the current user. |
| HISTFILE | The filename where past commands are stored. |
| HISTSIZE | The number of past commands that HISTFILE stores for the current session. |
| HISTFILESIZE | The number of past commands that HISTFILE stores for the multiple sessions. |
| HOME | The absolute path of the user's home directory. |
| HOST | The name of the computer. |
| HOSTNAME | HOSTNAME is identical to HOST, but used on certain distributions. |
| INFODIR | The path to the computer's information pages. |
| LOGNAME | The user name of the current user. |
| MAIL | The path to the current user's mailbox file. |
| MANPATH | The path to the computer's man pages. |
| OLDPWD | The path of the directory the user was in prior to the current path. |
| OSTYPE | The type of operating system. Usually this is Linux. |
| PATH | The directory prefixes used to search for programs and files.   * Use a colon to separate entries in the PATH variable. * Do not include a period (.) in the PATH variable. A period indicates that the working directory is in the path, and this poses a security risk. |
| PS1 | The characters the shell uses to indicate normal user ($), root user (#) and similar items. |
| PWD | The path of the current working directory. |
| LANG | The language the operating system uses. |
| PAGER | Used by the **man** command to specify the program in which to display man pages. |

The table below lists the most common environment variable commands:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **echo $***variable* | View the variable's value. | **echo $SHELL** displays the current shell's path. |
| **env** | Display the values for environment variables applied to child sessions. |  |
| **set** | Set shell environment variables. Without options, set displays the set environment variables for the system. |  |
| **unset** *variable* | Remove an environment variable. | **unset HOMEDIR** removes the HOMEDIR variable. |
| ***VARIABLE*=***value* | Create a user-defined environment variable.  **Note:** To append information to an environment variable, put the current variable in the command. For example, **PATH=$PATH:/bin/additionalpath**. | **HOMEDIR=/projects** gives the HOMEDIR variable a value of **/projects**. |
| **export** *variable* | Export a user-defined variable to make it available to child sessions. | **export HOMEDIR** makes the HOMEDIR user-defined variable available to child sessions. **PATH=$PATH:/bin/special ; export PATH** appends a directory to PATH and immediately exports the variable. |

**Alias Facts**

An *alias* is a custom command that performs a specific action. Most distributions have aliases that are invoked at startup; however, an alias can be invoked from the shell. Be aware of the following:

* Aliases defined with the **alias** command are *not* persistent across reboots.
* Add the alias to **/etc/profile** or **home/*user*/.bashrc** to make them persistent across reboots.

The following table describes the commands that create and remove aliases.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **alias** | Display the currently defined aliases on the system. |  |
| **alias***name* | Create a custom command that:   * Adds additional functionality to an existing command. * Performs multiple functions.   **Note**: When creating the alias, encapsulate the command(s) with quotation marks or apostrophes. | **alias ls='ls --ignore=\*.elf'**prevents the ls command from displaying .elf files even if **ls -a** is used. **alias securebackup='cp ./\*.\* /dev/st0/\*.\*;shred -fuvz ./\*'**creates a command that copies all files to the storage tape, then shreds the original files. **alias forcelogout="killall /usr/bin/Xorg"**creates a shortcut kills all Xserver processes. |
| **unalias** *name* | Remove an alias. | **unalias ls** removes all aliases specified for the **ls** command and places it back in its original state. **unalias forcelogout** deletes the forcelogout alias if it exists. |

#### Vim Commands

Vim (vi improved), also referred to as *vi*, is a utility that creates and modifies text files. It is the standard command-line text editor included with Linux distributions. Vim has the following modes:

* *Command*mode is the initial mode vim uses when started. It has commands that cut and replace text, and it is the mode vi uses to enter the other modes.
* *Command line*modeis the mode that works with the file system. Use it to save files after editing them.
* *Edit*modeis the mode that vim uses to write and edit text in the file. It has two operation modes:
  + *Insert*mode adds text between the preceding and subsequent text.
  + *Replace*mode overwrites subsequent text.

The table below lists some of the most common vi commands.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Mode** |
| **vi** | Start vim. Type the command at the shell prompt. | N/A |
| **1vi** *filename* | Start vim and immediately begin working on the named file (either a new, non-existent file or an existing file). Type the **vi** command at the shell prompt. | N/A |
| **Insert** key **i s** | Enter insert mode from command mode | Command |
| **Esc**key | Enter the command mode from the edit mode. | Insert/Replace |
| **Delete**key | Delete text | Insert/Replace |
| **Insert** key | Change between insert and replace mode while in edit mode. | Insert/Replace |
| ***#***(line number) | Go to a specific line in the document while in command mode. For example **94**, moves the cursor to line 94. | Command |
| **dw** | Cut a whole word and trailing space. | Command |
| **de** | Cut a whole word but omit the trailing space. | Command |
| **d$**or**D** | Cut all text following the cursor to the end of the line. | Command |
| **dd** | Cut a line from the text. | Command |
| **p** | Place text in memory into the document. | Command |
| **u** | Undo the last action. | Command |
| **O** | Open a new line above the current line. | Command |
| **o** | Open a new line below the current line. | Command |
| **Ctrl+g** | Display file name, total number of lines in the file and the cursor position. | Command |
| **/***term* | Search forward for all instances of a term. Press n to go to the next term and N to go to the previous term. | Command |
| **?***term* | Search backward for all instances of a term.  Press n to go to the previous term and N to go to the next term. | Command |
| **yy** | Copy a line of text into memory. | Command |
| **a** | Append text after the cursor | Command |
| **A** | Append text after the current line | Command |
| **C** | Change text from current cursor position to the end of the line. | Command |
| **cc** | Change text of the entire line. | Command |
| **ZZ** | Save current file and exit vim. | Command |
| **h** | Move the cursor one space to the left. | Command |
| **j** | Move the cursor down a line. | Command |
| **k** | Move the cursor up a line. | Command |
| **l** | Move the cursor one space to the right. | Command |
| **z** | Exit without saving | Command |
| **:** | Enter command line mode from command mode | Command |
| **w** | Save the current document. | Command line |
| **w** *filename* | Name and save the file. | Command line |
| **w!** *filename* | Overwrite the file | Command line |
| **q** | Exit vim. This produces an error if the text was modified. | Command line |
| **q!** | Exit vim without saving. | Command line |
| **wq** or **exit** | Save the document and exit vim. | Command line |
| **e!** | Reload the file from the last saved version. This discards all edits and reloads the last saved version of the file into vim. | Command line |

#### Redirection and Piping Facts

**Administrators often use the following methods to create, send, or gather information on a Linux system:**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Redirection | *Redirection* directs standard input, output, and error streams from and to locations other than the default. Be aware of the following redirection details:   * By default, the Linux system classifies information with the following file descriptors:   + Standard input (stdin) comes from the keyboard. In redirection, 0 represents stdin.   + Standard output (stdout) displays on the monitor. In redirection, 1 represents stdout.   + Standard errors (stderr) display on the monitor. In redirection, 2 represents stderr. * Linux commands use the greater-than symbol (**>**) to show redirection of output, the less-than symbol arrow (**<**) to indicate redirection of input, and the double greater-than symbol (**>>**) to append the output to another file or command. * The **tee** command reads from standard input and writes to standard output *and* files. |
| Piping | *Piping* directs the output of one command into the input of another command. Pipes:   * Use the pipe symbol (**|**). * Can combine several commands to make a stream. |

**The following table shows the results of several redirection and piping commands:**

|  |  |
| --- | --- |
| **Example** | **Result** |
| **ls /usr > /tmp/deleteme** | **ls /usr > /tmp/deleteme** places the list of files in the **/usr** directory into a file named **/tmp/deleteme**. |
| **ls /nonesuch > /tmp/deleteme** | **ls /nonesuch > /tmp/deleteme** does not write anything to a file and sends an error message to the monitor '/nonesuch not found'. |
| **ls /nonesuch 2 > /tmp/deleteme** | **ls /nonesuch 2 > /tmp/deleteme** writes the standard error message to a file named **/tmp/deleteme**. |
| **ls /bin /nonesuch > /tmp/deleteme** | **ls /bin /nonesuch > /tmp/deleteme** writes the contents of the **/bin** directory to the **/tmp/deleteme** file, but sends the error message '/nonesuch not found' to the screen. |
| **ls /bin /nonesuch > /tmp/deleteme 2>&1** | **ls /bin /nonesuch > /tmp/deleteme 2>&1** directs the standard output to the **/tmp/deleteme** file, then directs that the standard error messages be sent to the same place as the standard output. Both the list of files in the **/bin** directory and the error message are written to the file. |
| **ls /bin /nonesuch 2>&1 > /tmp/deleteme** | **ls /bin /nonesuch 2>&1 > /tmp/deleteme** writes the contents of the /bin directory to the **/tmp/deleteme** file, but sends the error message '/nonesuch not found' to the screen. This is because standard error messages are directed to the same place that standard output goes, but this is before standard output has been directed to the file. |
| **ls /bin >> /tmp/deleteme** | **ls /bin >> /tmp/deleteme** appends the list of files from the **/usr** directory on to the end of the **/tmp/deleteme** file. |
| **sort < unordered\_file.txt > ordered\_file.txt** | **sort < unordered\_file.txt > ordered\_file.txt** takes input from the unordered\_file.txt file sends it to the sort command, and then writes a new file named **ordered\_file.txt**. |
| **cat /usr/wordlist1 /usr/wordlist2 | sort** | Sends the output of the cat command, the contents of wordlist1 and wordlist2, to the input of the sort command. The result is a sorted list of the combined contents of **wordlist1** and **wordlist2**. |
| **cat /usr/wordlist1 /usr/wordlist2 | mail jdoe** | Mails the combined list of words in **wordlist1** and **wordlist2** to the user jdoe. |
| **ls /bin | sort | mail jdoe** | Lists the contents of /bin then sorts the combined contents and mails them to jdoe. |
| **cat /usr/wordlist1 /usr/wordlist2 | sort | tee sortedwordlist** | Lists the contents of **wordlist1** and **wordlist2** then sorts the combined contents, then sends the results to the monitor and a file named **sortedwordlist**. |
| **cat /usr/wordlist1 | tee log.txt** | Writes to the standard output and the log.txt file. |

#### Xargs Facts

The **xargs** command reads items from the standard input and breaks up long lists of arguments into smaller, usable lists. Xargs:

* Makes it easier to pipe input into commands that take arguments.
* Overcomes a 128 KB shell command size restriction in older Linux kernels.
* Commonly takes input from the following commands:
  + **find**
  + **ls**
  + **locate**
  + **grep -l**

The following table describes the common **xargs** options:.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **-0** | Ignore space names in files | **find / -print0 -name \*.odt | xargs -0 rm**deletes all .odt files in the file system, even those with spaces in the file names. |
| **-I***variable* | Replace the initial argument of a command with the argument from the standard input. | **find / -name '\*.jpg' | xargs -I var1 cp var1 /home/user/Pictures** finds all the .jpg files on the computer and copies them into the**/home/user/Pictures** directory |

#### Directory Commands

The following table describes several basic commands when managing directories:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **pwd** | See the present working directory. | Shells open to the home directory of the current user. For example, if a user named Fred opened a shell and typed **pwd** at the prompt, **/home/Fred** is displayed. |
| **cd** | Change the present working directory. | **cd directory1** changes to a directory named **directory1** if it exists in the present working directory. (This is the *relative* path.) **cd /home/Fred/directory1** opens **directory1** regardless of the present working directory. (This is the *absolute* path.) **cd ..** changes to the parent directory. **cd /** changes to the root directory. |
| **ls** | Display the contents of a directory. Options include:   * **-a** displays all directory contents, including hidden content. * **-l** displays a long listing for directory contents, including the owner, modified date, size, and permissions. * **-R** displays the contents of the directory and all sub-directories. * **-d** displays directories but not files. * **-r** reverses the sort order. | **ls -al** displays the long listing of all the contents in the present working directory. **ls -d** only displays the directories. **ls -R /etc** displays all the contents of the **/etc**directory and all sub-directories. |
| **mkdir** | Create a new directory. Use the**-p** option to create the directories that do not exist. | **mkdir work\_files** creates a directory in the present working directory. **mkdir /home/Fred/work\_files** creates a directory at the specified path. |
| **cp -r cp -R** | Copy directories. Copying leaves the source contents (directories and files) intact. | **cp -r /temp /home/user** copies the entire **/temp** directory with all of its files, sub-directories, and files in the sub-directories to the **/home/user**directory. |
| **mv** | Move or rename directories (and files). Moving directories erases the source directory and places it in the destination. Options include:   * **-f** overwrites directory that already exist in the destination directory. * **-i** prompts before overwriting a directory in the destination directory. * **-n** never overwrites files in the destination directory. | **mv /temp/station ~/doc/** moves **station** from the **/temp** directory to the **~/doc** directory. |
| **rmdir** | Delete an empty directory. | **rmdir ~/Fred/work\_files** deletes the **work\_files** directory provided it is empty. |
| **rm** | Remove the directory (and file) inode, but not actually delete the data. Options include:   * **-r** deletes directories (and all files) in the directories. * **-f** deletes without prompting. | **rm -rf /home/user/temp** deletes the temp directory with all its sub-directories and files without prompting. **rm -r /home/user/\*** deletes all directories and files in the **/home/user** directory. |

#### File Commands

The following table describes several basic commands when managing files:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **touch** | Create a blank file if the file does not exist, or to update the file's modification and last accessed times if the file exists. | **touch myfile** makes a blank file named **myfile**. |
| **cat** | Display the contents of the file in the shell. This can include displaying multiple files at once. | **cat myfile** displays the contents of the file **myfile**. **cat myfile** **yourfile** displays the contents of the file **myfile**and **yourfile** together. |
| **less** | Display the file one screen at a time.   * Use the SpaceBarto scroll to the next screen. * Use the Up arrow and Down arrow to scroll up and down. * Type **q** to exit. | **less bigfile** displays the contents of **bigfile** one screen at a time so it can be read. |
| **head** | List the first 10 lines of a specified file, by default. The **-n** option specifies a specific number of lines. | **head /home/user/myfile** lists the first 10 lines of **myfile**. **head -n 20 /home/user/myfile** lists the first 20 lines of **myfile**. **head -n -35 /home/user/myfile** displays all lines in **myfile**, omitting the last 35 lines. |
| **tail** | List the last 10 lines of a specified file, by default. Options include:   * **-n** specifies a specific number of lines. * **-f**monitors the file. | **tail /home/user/myfile** lists the last 10 lines of **myfile**. **tail -n 20 /home/user/myfile** lists the last 20 lines of **myfile**. **tail -n -15 /home/user/myfile** displays all lines in **myfile**, omitting the first 15 lines. |
| **file** | Show the file type. The **file** command might often be necessary because Linux does not require file extensions. **file**uses file signatures in:   * **/usr/share/misc/magic** * **/usr/share/misc/magic.mgc** * **/etc/magic** | **file myfile** shows whether **myfile** is a text, data, xml or other type of file. |
| **cp** | Copy files. Copying leaves the source file intact. Options include:   * **-f** overwrites files that already exist in the destination directory. * **-i** prompts before overwriting a file in the destination directory. | **cp /temp/document\_ab.txt ~/doc/document.txt** copies **document\_ab.txt** from the **/temp** directory to the**~/doc** directory and renames the file to **document.txt**. **cp /temp/\*.txt ~/doc**copies all text files from the **/temp** directory to the **~/doc** directory. |
| **mv** | Move or rename files (and directories). Moving files erases the source file and places it in the destination. Options include:   * **-f** overwrites files that already exist in the destination directory. * **-i** prompts before overwriting a file in the destination directory. * **-n** never overwrites files in the destination directory. | **mv /temp/document.txt ~/doc/document.txt** moves **document.txt** from the **/temp** directory to the **~/doc**directory. **mv /temp/\*.txt ~/doc/\*.txt**copies all text files from the **/temp** directory to the **~/doc** directory. |
| **rm** | Remove the file (and directory) inode, but not actually delete the data. The **-f**option deletes without prompting. | **rm myfile** deletes a file in the current directory named **myfile**. **rm /home/user/myfile** deletes **myfile** from the **/home/user** directory regardless of the current directory. **rm -f /home/user/temp/\*** deletes all files in the temp directory without promptings. |
| **shred** | Deletes the file and overwrites the file information. **shred** is useful when deleting files that contain proprietary company information or other sensitive data.  Options include:   * **-n**specifies the times to overwrite. The default is 25 times. * **-u** deletes the inode. * **-v**display the progress of the file deletion. * **-z**overwrites the filename with zeros. | **shred -u -z companysecrets.txt** deletes the file **companysecrets.txt**, overwrites the file with random information, then leaves zeros in place of the file. |
| **lsattr** | List file attributes.   * **-R** recursively list attributes of directories and their contents. * **-V** displays the program version. * **-a** lists all files in directories. * **-d** lists directories like other files, rather than listing their contents. * **-v**lists the file's version/generation number. | **lsattr /etc/grub/grub.conf** lists the attributes of the **grub.conf** file. |

#### Link Facts

Links are files that point to another file. When links are accessed, they reference the source file's inode*.* The *inode*specifies where a file's data physically exists on a disk. Link types include:

|  |  |
| --- | --- |
| **Type** | **Description** |
| Hard link | A *hard link* is a duplicate entry in the file system that points to a specific piece of data on the disk drive. A hard link:   * Creates a duplicate file inode. * Is indistinguishable from the original files. * Maintains a valid inode for the file data even if the original file is deleted. * Has a dash (-) as the first character in the permission string (which is the same for original files). For example, **-**rwxr-xr-x. |
| Symbolic link | A *symbolic link* (also known as a *soft link*) is a file system entry that points to another file system entry, which in turn points to a valid piece of data. A symbolic link:   * Has a distinct inode. * Can work across volumes and file systems. * Is similar to shortcuts in the Windows OS. * Has a lower-case L (l) as the first character in the permission string. For example, **l**rwxrwxrwx indicates a symbolic link. |

The table below describes the commands for creating hard links and symbolic links:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **ln***source link\_name* | Create links.   * **ln -s** creates a symbolic link to a file. * **ln -b** creates backup of a file. * **ln -i**determines the inode for hard or symbolic links. * **ln**(with no options) creates a hard link between files. | **ln /home/jsmith/project1 /home/edunford/project1** creates an exact copy of **/home/jsmith/project1** in **/home/edunford/**. **ln -s /home/jsmith/project1 /home/edunford/project1\_ln** creates a pointer named **/home/edunford/project1\_ln** that points to **/home/jsmith/project1**. **ln -s /home/jsmith/project1 /home/edunford/project1\_ln** creates a pointer named **/home/edunford/project1\_ln** that points to **/home/jsmith/project1**. **ln -i** displays the inodes for the contents in the present working directory. **ln -b /home/jsmith/file1 /bup**copies **file1**as **file1~**in **/bup**. |
| **cp***source link\_name* | Copy files and create links.   * **cp -l** creates hard links rather than copying the files. * **cp -s** creates symbolic links rather than copying the files. | **cp -l /home/jed/fil1 /home/esam/proj1** creates an exact copy of **/home/jed/fil1** in **/home/esam/**. **cp -s /home/mkon/text /home/ytew/text\_ln** creates a symbolic link named **/home/ytew/text\_ln** that points to **/home/mkon/text**. |

#### Filesystem Hierarchy Standard (FHS) Facts

The Filesystem Hierarchy Standard (FHS) governs the unified file system for Linux systems by defining a standard set of directories, subdirectories, and files. FHS is a subset of the Linux Standards Base (LSB) which is an organization and a set of guidelines for promoting a set of standards to increase Linux distribution compatibility.

|  |  |
| --- | --- |
| **Directory** | **Description** |
| **/** | The **/** character represents the root directory of the Linux system. All directories are below the / (root directory) of the system. |
| **/bin** | The **/bin** directory contains binary commands that are available to all users. |
| **/boot** | The **/boot** directory contains the kernel and bootloader files. |
| **/dev** | The **/dev** directory contains device files that represent the devices used by the system, such as a hard drive, mouse, and printer. |
| **/etc** | The **/etc** directory contains configuration files specific to the system. |
| **/home** | The **/home** directory contains by default the user home directories. |
| **/lib** | The **/lib** directory contains shared program libraries and kernel modules. |
| **/media** | The **/media** directory contains the /cdrom and /floppy directories. |
| **/mnt** | The **/mnt** directory is an empty directory, and is often used for temporarily mounted filesystems. |
| **/opt** | The**/opt** directory contains the additional programs on the system. |
| **/proc** | The **/proc** directory contains information about the system state and processes. |
| **/root** | The **/root** directory is the root user's home directory. Do not confuse **/root** with the root of the system (/). |
| **/sbin** | The **/sbin** directory contains system binary commands. |
| **/srv** | The **/srv** directory contains files for services such as HTTP and FTP servers. |
| **/sys** | The**/sys** directory contains the *sysfs* virtual filesystem which displays information about devices and drivers. |
| **/tmp** | The **/tmp** directory contains temporary files created by programs during system use. |
| **/usr** | The **/usr** directory contains system commands and utilities. |
| **/var** | The **/var** directory contains data files that change constantly. Standard subdirectories include:   * /var/mail (holds e-mail in boxes) * /var/spool (holds files waiting for processing, such as print jobs or scheduled jobs) * /var/www (holds www or proxy cache files) |

#### File Location Commands

Use the following commands to find file locations:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **find** | Search through all files based on the file system by name, file size, time created, and other options. Be aware of the following **find** options:   * **-name**locates a file or directory by name in a specific path. When using **-name**:   + Enclose name strings in single quotes.   + Use wildcards for partial names.   + Use **-iname** for case insensitive. * **-user** finds files owned by a specific user. * **-size**finds files of a specific size. Use the following options:   + **c** for bytes   + **k** for kilobytes   + **M** for megabytes. * **-mtime**finds files last modified before or after a specified number of days ago. * **-type [fd]** specifies whether to find files or directories. * **-maxdepth** specifies how many levels down to search. * -**print0**finds filenames with spaces. * **-o** specifies the *or* parameter when searching with multiple criteria. * **.** (period) specifies the search locations as the current directory and subdirectories. | **find /user/home -name '\*.txt'** finds all plain text files in the **/user/home** directory. **find / -name '\*paper\*'** looks through the entire directory for any folder or directory name with the term *paper* in it, such as**termpaper.odt** or **wallpaper.jpg**. **find /user/home -size -300k** finds all files in the **/user/home** directory smaller than 300K.  **find /user/home -size +300k** finds all files in the **/user/home** directory larger than 300K.  **find /user/home -mtime -5** finds all files in the **/user/home** directory modified within the last five days. **find / -type f -name '\*paper\*'**finds only files with the string *paper* in the name. **find / -type d -name '\*paper\*'** finds only directories with the string *paper* in the name. **find -maxdepth 3 / -name '\*.txt'**finds text files three directory levels down from the root directory. **find -print0 -name '\*.txt'** finds myreport.txt and 'my report.txt'. Without the -print0 option, 'my report.txt' is not listed. |
| **locate** | Search an index file for specific parameters. **locate**:   * Is much faster than **find**. * Searches **/var/log/locatedb** as the index file. * Uses the **updatedb** command to update the file index. (**/etc/updatedb.conf** is the configuration file for **updatedb**) * Searches from the root (/) directory if no path is specified. * Finds all files that contain the specified string without using wildcards. * Does not by default verify that the file exists if its file index is outdated. * Does not display files created after the last time the file index was updated. * Does not search for files by attribute.   Be aware of the following **locate** options:   * **-c**counts the number of entries rather than list them. * **-e**lists files only after verifying that they exist. * **-i**ignores case. * **-l**limits the number of files listed. * **-b**searches for the string in only file or directory base names. | **updatedb**updates the index file, **/var/log/locatedb**. **locate /user/home paper** locates all files with the string *paper* as any part of the file name or directory path under the /user/home directory. **locate lib** locates all files with the string *lib* anywhere in the file name or directory path. **locate -c lib**counts the number of files with the string *lib*. For example **46512**. **locate -e .odt**verifies that all .odt files listed in the file index actually exist before it lists them. **locate -i LibraryFines.csv** finds the **libraryfines.csv** file regardless of case. **locate -l 25 lib** lists only the first 25 files from the file index that contain the string *lib*. **locate -b lib** displays**/var/lib** and **/user/home/libraryfines.csv** but not **/var/lib/usbutils/usb.ids** |
| **which** | Display the path to a command and determine whether a package is installed. | **which ls** shows the path to the **ls** binary (executable) file. **which photorec** shows the path to the **photorec** binary file if **photorec** is installed. If the command does not display a path, then the **photorec** utility is not installed. |
| **whereis** | Display the path to the binary files, the manual pages, and the source code. Be aware of the following options:   * **-b**lists the path to the binary file (similar to **which**). * **-m**lists the path to the man page files. * **-s**lists the location of the source code. * **-u**lists entries that do not have source code, binary file, and man page locations.   **Note**: When no options are specified, **whereis** shows all available data. | **whereis -m -u \*** lists all entries that have no man page location. |
| **type** | Display the category of the command. Possible categories include:   * A built-in shell command * A command that the shell calls * An aliased command * A function   **Note**: If a called command has been used recently, the output says that the command is *hashed*, which means that it is in the shell's hash table. | **type cd** displays *cd is a shell built-in.* **type more** displays the path to the binary file for **more**. |

**Note**: The term *file globbing*refers to the use ofwildcards (e.g., **\***, **\*.\***, **\*.txt**) to match specific files.

#### Content Search Commands

**Grep** searches through file text for specific words or character patterns. The following table describes the **grep**, **egrep**, and **fgrep** commands and lists several of their options.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **grep** | Search through files for a specified character string. By default **grep** is context sensitive and displays the string in the context of the line containing the string.   * **-A***number*prints a specified number of lines following the matching lines. * **-a** searches binary (executable) files as though they were text files. * **-B***number*prints a specified number of lines before the matching lines. * **-C***number*prints a specified number of lines of context around the matching lines. * **-c**shows the number of matches of the string for the file. * **-E**uses regular expressions for the text pattern. * **-e***pattern*specifies a literal pattern. * **-f**searches for multiple strings using a file that lists the string patterns. * **-F** uses a file as the source for the string patterns. * **-i**ignores the case of the string. * **-l**lists just the names of the files with a match. This is used when search multiple files. * **-m***number*shows only a specified number of matches for a file. * **-n**displays the line number of the lines containing the term. * **-r** searches the directory and all sub-directories for files containing the term. * **-v** displays non-matching lines. * **--include=***file\_name*searches only in files with names that match a specified string. * **--exclude=***file\_name*searches in files with names that do *not* match a specified string. * **-w**searches for whole words only. | **grep -A 3 Midway ~/docs/WWII-report** searches **WWII-report**for the pattern *Midway* and prints the line and the next three lines. **grep -a var11 /bin** searches all files, including binary files, in the **/bin** directory for the pattern *var11*. **grep -c 3 Midway ~/docs/WWII-report**shows only the number of times the pattern *Midway* is found in the **WWII-report**file. **grep -e '--count' ~/docs/doc1**looks for the pattern *--count* in the **doc1** file rather than interpreting it as an option. **grep -l -r Midway ~/docs**shows the name of all files in the **/home/user/docs**directory that contain the term*Midway*. **grep -m 2 battle ~/docs/WWII-report**shows only the first two times the term *battle* is found in the file. **grep -n -i customVariable1 ~/java/program1.java**shows the line numbers of lines that have the term*customVariable1*in the**program1.java** file. This is case insensitive. **grep -r battle ~/docs/**searches the directory and all sub-directories for the term *battle*.  **grep -w tank ~/docs/WWII-report**searches only for the whole word *tank* in the file. |
| **egrep** | Use regular expressions in the search strings. **egrep** uses the same options and syntax as **grep**, and is identical to**grep -E**. Constructors for **egrep** regular expressions include:   * **^**matches terms that occur at the beginning of a line. * **$** matches terms that occur at the end of a line. * **\<**matches words that begin with the term. * **\>**matches words that end with the term. * **[asdf]**matches any one of the characters in the brackets. * **[0-9]**matches any of the range of numbers 0-9. * **[^xyz]**omits any one of the letters in the list * **.**matches any single character. * **[asdf]+**matches one or more of the characters in the list. * **\***matches any number, or none of the preceding single character * **|** matches either of the terms. * **\** displays the literal value of a character used for expressions. * **()** groups expressions. | **egrep ^FAILURE ~/error\_logs**matches the term *FAILURE* when it is at the beginning of the line in **error\_logs**. **egrep tty7$ ~/.bash\_history**matches the term *tty7* when it is at the end of the line. **egrep \<are ~/myfile**matches all words or strings that begin with *are*. This includes *are*, *area*, and *arena*.  **egrep \>are ~/myfile**matches all words or strings that end with *are*. This includes *are*, *hare*, and *aware*. **egrep watche[ds] ~/myfile** matches either *watched* or *watches*. **egrep exhibit[0-9] ~/myfile**matches *exhibit1*, *exhibit3*, or *exhibit8*. **egrep [^Xx]mas ~/myfile**matches *Christmas*but not *xmas* or *Xmas*. **egrep .are ~/myfile** matches *hare* and *care*, but not *aware* or *are*. **egrep file[0-9]+ ~/myfile** matches *file0*, *file10*, and *file15636*. **egrep fil\* ~/myfile** matches *fil*, *filll*, and *fillllllllllllllll*.  **egrep fil.\* ~/myfile**matches *file*, *fill*, *file102*, and *filings*. **egrep men|women ~/myfile** matches *men* or *women*. **egrep Hello\? ~/myfile** matches *Hello?*. |
| **fgrep** | Search for fixed strings, rather than regular expressions. **fgrep**:   * Uses the same options **grep** uses, and has the same syntax. * Is identical to **grep -F**, but searches faster than **grep**. * Interprets the pattern as a list of fixed strings, any of which can be matched. | f**grep Midway Nimitz ~/docs/myfile** searches **myfile** for lines containing *Midway* or *Nimitz*. |

Linux is extremely flexible in regard to where user and group information is stored. The options for storing the information are:

* Local file system.
* LDAP-compliant database.
* NIS, network information system. NIS allows many Linux computers to share a common set of user accounts, group accounts and passwords.
* A Windows domain.

When the files are stored in the local file system, the following files are used.

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/passwd** | The **/etc/passwd** file contains the user account information. Each user's information is stored in a single line in this file. The syntax for the file is:  *USER:PW:UID:GID:FULL\_NAME:HOME:SHELL*   There are two types of accounts in a Linux system:   * Standard accounts that are user accounts * System user accounts that are used by services |
| **/etc/shadow** | The **/etc/shadow** file contains the users' passwords in encrypted format. The shadow file is linked to the**/etc/passwd** file. There are corresponding entries in both files and they must stay synchronized. The syntax for the file is:  *USER:PASSWD:LASTMOD:MINDAYS:MAXDAYS:WARN:DIS:EXP*  There are password and user management utilities provided by the system that will allow you to edit the files and keep them synchronized.  You can use the following commands to identify errors and synchronize the files.   * **pwck** - verifies each line in the two files and identifies discrepancies. * **pwconv** - adds the necessary information to synchronize the files |
| **/etc/group** | As with Active Directory, groups can be used to simplify user access to network resources. The**/etc/group** file contains information about each group. The syntax for the file is:  *GROUP:PASSWORD:GID:USERS* |
| **/etc/gshadow** | Some distributions use the **/etc/gshadow** file to store group passwords. The syntax for the file is:  *GROUP:PASS:GROUP\_ADMINS:MEMBERS* |

Be aware of the following configuration files when managing user accounts:

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/default/useradd** | The **/etc/default/useradd** file contains default values used by the **useradd** utility when creating a user account, including:   * Group ID * Home directory * Account expiration * Default shell * Secondary group membership |
| **/etc/login.defs** | The **/etc/login.defs** file contains:   * Values used for the group and user ID numbers. * Parameters for passwords encryption in the shadow file. * Password expiration values for user accounts. |
| **/etc/skel** | The **/etc/skel** directory contains a set of configuration file templates that are copied into a new user's home directory when it is created, including the following files:   * **.bashrc** * **.bash\_logout** * **.bash\_profile** * **.kshrc** |

Although it is possible to edit the **/etc/passwd** and **/etc/shadow** files manually to manage user accounts, doing so can disable your system. Instead, use the following commands to manage user accounts:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **useradd** | Create a user account. The following options override the settings as found in **/etc/default/useradd**:   * **-c** adds a description for the account in the GECOS field of **/etc/passwd**. * **-d** assigns an absolute pathname to a custom home directory location. * **-D** displays the default values specified in the **/etc/default/useradd**file. * **-e**specifies the date on which the user account will be disabled. * **-f** specifies the number of days after a password expires until the account is permanently disabled. * **-g**defines the primary group membership. * **-G**defines the secondary group membership. * **-M**does not createthe user's home directory. * **-m**creates the user's home directory (if it does not exist). * **-n, N**does not create a group with the same name as the user (Red Hat and Fedora respectively). * **-p**defines the encrypted password. * **-r**specifies that the user account is a system user. * **-s** defines the default shell. * **-u** assigns the user a custom UID. This is useful when assigning ownership of files and directories to a different user. | **useradd pmaxwell** creates the *pmaxwell* user account**. useradd -c "Paul Morril" pmorril** creates the *pmorril* account with a comment**. useradd -d /tmpusr/sales1 sales1** creates the *sales1* user account with home directory located at */tmpusr/sales1*. **useradd -u 789 dphilips** creates the *dphilips* account with user ID *789*. |
| **passwd** | Assign or change a password for a user.   * **passwd** (without a username or options) changes the current user's password. * Users can change their own passwords. The root user can execute all other **passwd** commands.   Be aware of the following options:   * **-S *username***displays the status of the user account.   + LK indicates the user account is locked.   + PS indicates the user account has a password. * **-l** disables (locks) an account. This command inserts a !! before the password in the **/etc/shadow**file, effectively disabling the account. * **-u** enables (unlocks) an account. * **-d** removes the password from an account. * **-n** sets the minimum number of days a password exists before it can be changed. * **-x** *s*ets the number of days before a user must change the password (password expiration time). * **-w** sets the number of days before the password expires that the user is warned. * **-i**sets the number of days following the password expiration that the account will be disabled. | **passwd jsmith** changes the password for the *jsmith* account.  **passwd -d** removes the password from an account. **passwd -d jsmith** removes the password from the *jsmith* account. **passwd -x 40 jsmith** requires *jsmith* to change his password every 40 days. **passwd -n 10 jsmith** means that *jsmith* cannot change his password for 10 days following the most recent change. **passwd -w 2 jsmith** means that *jsmith* will be warned 2 days before his password expires. **passwd -i 7 jsmith** disables the *jsmith* account after 7 days if the password is not changed. **passwd -l jsmith** locks the *jsmith* account. **passwd -u jsmith** unlocks the *jsmith* account. |
| **usermod** | Modify an existing user account. **usermod** uses several of the same switches as **useradd**. Be aware of the following switches:   * **-c**changes the description for the account. * **-l** renames a user account. When renaming the account:use   + Use **-d** to rename the home directory.   + Use **-m** to copy all files from the existing home directory to the new home directory. * **-L**locks the user account. This command inserts a ! before the password in the **/etc/shadow**file, effectively disabling the account. * **-U** unlocks the user account. | **usermod -c "Paul Morril" pmorril** changes the comment field for user *pmorril*. **usermod -l esmith -d /home/esmith -m ejones** renames the *ejones* account to *esmith*, renames the home directory, and moves the old home directory contents to the new location. **usermod -s /bin/tsch esmith** points the shell for *esmith* to */bin/tsch*. **usermod -U** **esmith** unlocks the *esmith* account. |
| **userdel** | Remove the user from the system. Be aware of the following options:   * **userdel *username*** (without options) removes the user account. * **-r** removes the user's home directory. * **-f**forces the removal of the user account even when the user is logged into the system. | **userdel pmaxwell**deletes the *pmaxwell* account while leaving the home directory on the hard drive. **userdel -r pmorril**removes both the account and the home directory. |

Use the following commands to manage group accounts and group membership:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **groupadd** | Create a new group. The following options override the settings as found in **/etc/login.defs**:   * **-g** defines the group ID (GID). * **-p**defines the group password. * **-r**creates a system group. | **groupadd sales** creates the *sales* group. |
| **groupmod** | Modify the existing group. Be aware of the following options:   * **-A**adds specified users from the group (SUSE distribution) * **-R**removes specified users from the group (SUSE distribution) * **-n**changes the name of a group. | **groupmod -R rsem sales** removes the *rsem* account from the *sales* group. **groupmod -n sales2 sales** renames the *sales* group to *sales2*. |
| **groupdel** | Delete a group. | **groupdel mktg** deletes the *mktg* group |
| **gpasswd** | Change a group password.   * ***groupname*** prompts for a new password. * **-r** removes a group password). | **gpasswd sales** prompts for a new group password |
| **newgrp** | Log in to a new group with the group password. | **newgrp sales** prompts for the password for the *sales* group before logging in. |
| **usermod** | Modify group membership for the user account. Be aware of the following options:   * **-g** assigns a user to a primary group. * **-G** assigns a user to a secondary group (or groups). Follow the command with a comma-separated list of groups. * **-G ""** Remove the user from all secondary group memberships. Do not include a space between the quotes.   When assigning a user to one or more secondary groups, all existing secondary group memberships are removed before assigning the user account to new groups. | **useradd -g pmaxwell pmaxwell** assigns primary group membership for user *pmaxwell* to the *pmaxwell* group.  **usermod -G sales,mktg pmorril**removes all existing secondary group assignments for *pmorril* and makes the user account a member of the *sales* and *mktg* groups.  **usermod -G "" pmaxwell** removes the *pmaxwell* from all groups. |
| **groups** | Display the primary and secondary group membership for the specified user account. | **groups pmaxwell** displays group membership for the *pmaxwell* account. |

When considering user security, keep in mind the following:

* Users should be trained to use secure passwords. Secure passwords use numbers and letters, and are more than 7 characters in length.
* Passwords should expire periodically, but not too often.
* Administrators can limit the resources that user can access.

The following table describes commands used to promote user security and restrictions:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **chage** | Set user passwords to expire. Be aware of the following options:   * **-M** sets the maximum number of days before the password expires. * **-W** sets the number of days before the password expires that a warning message displays. * **-m** sets the minimum number of days that must pass after a password has been changed before a user can change the password again.   Look in the **/etc/shadow** file to see current limits for users. | **chage -M 60 -W 10 jsmith** sets the password for *jsmith* to expire after 60 days and gives a warning 10 days before it expires. |
| **ulimit** | Limit computer resources used for applications launched from the shell. Limits can be hard or soft limits. Soft limits can be temporarily exceeded up to the hard limit setting. Users can modify soft limits, but only root can modify hard limits.  Options include:   * **-c** limits the size of a core dump file. The value is in blocks. * **-f** limits the file size of files created using the shell session. The value is in blocks. * **-n** limits the maximum number of open files. * **-t** limits the amount of CPU time a process can use. This is set in seconds. * **-u** limits the number of concurrent processes a user can run. * **-d** limits the maximum amount of memory a process can use. The value is in kilobytes. * **-H** sets a hard resource limit. * **-S** sets a soft resource limit. * **-a** displays current limits. The default shows soft limits. | **ulimit -H -f 1024** uses a hard limit to limit the size of files to 1020 KB. **ulimit -H -a** shows current hard limits. **ulimit -a** shows the current soft limits. **ulimit -S -u 10** sets a soft limit that limits the number of processes that a single user can use to 10. **ulimit -t 600** limits CPU time for a process to 10 minutes. This sets both hard and soft limits. **ulimit -d unlimited** removes all restrictions for process memory usage. |

Use the**/etc/security/limits.conf**file to limit resource use for all applications. This file is from the *pam\_limits* module of the Pluggable Authentication Modules (PAM) module set. Entries in **/etc/security/limits.conf** use the following syntax:

*Entity    Type    Limit    Value*

The following table describes the entry options in the **/etc/security/limits.conf**file:

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Type** | **Limits** | **Value** |
| When specifying the Entity:   * Specify a single user with a user name. * Use an at sign (@) to specify a group. * Use an asterisk (\*) as a wildcard. | For the Type:   * Use **hard** to set a limit that cannot be exceeded. * Use **soft** to set a limit that can be exceeded temporarily. | Limits include:   * **core** limits the size of core dump files. The value uses kilobytes. * **data** limits the amount of ram an application can use. The value uses kilobytes. * **fsize** limits maximum file size. The value uses kilobytes. * **nofile** limits the number of concurrently open data files. * **cpu** limits the amount of CPU time a process can use. The value uses minutes. * **nproc** limits the number of concurrent processes a user can have. * **maxlogins** limits the number of concurrent logins. * **priority** sets process priority limits. The value range is from -20 (highest priority) to 19 (lowest priority) with 0 being the default. * **rss** limits the total amount of memory a user can use. The value uses kilobytes. | Values include integers, such as 1, 5, or 3000. |

The following are examples of entries in the **/etc/security/limits.conf**file:

|  |  |
| --- | --- |
| **Example** | **Description** |
| **jsmith        hard    fsize             1024** | Limits the maximum file size that *jsmith* can create to 1024 KB. |
| **@guests    hard    maxlogins    3** | Limits the number of concurrent logins from the *guest* group to three. |
| **\*                hard    maxlogins    1** | Limits concurrent logins from the same user to one. |
| **\*                soft      cpu              10** | Sets a soft limit of 10 minutes on the amount of CPU time any single process for any user can take. |
| **rss             hard    rss                5000** | Limits the total amount of memory available to a single user to 5 MB |

Use **cat /etc/group** or **groups *username***

**Installation Design Facts**

Be aware of the following advantages to planning and designing a Linux installation:

* A plan ensures that the installer knows exactly what should happen during the installation. The plan places all the information in the installer's hands before installation begins.
* There are fewer variables involved when diagnosing and resolving problems.
* The plan ensures that there is something concrete to reference if managers add requests after the installation begins. It gives the IT team something to use as a point of reference if the changes require a change in resources or schedule.

The following table describes general steps in an effective installation design:

|  |  |
| --- | --- |
| **Step** | **Description** |
| Perform a needs assessment | An effective assessment determines the goals of the installation, creates a plan to meet those needs, and measures the results of the plan. This involves:   * Interviewing managers to determine the goals they want to achieve, what problems they need to solve, the expected results. * Writing clear, measureable statements that specifically address the goals. * Indentifying the stake holders. * Confirming correct authorization. * Aligning the installation with current organizational strategy and technology. * Verifying funding. * Creating a support strategy. * Determining the scope:   + Identify deadlines.   + Determine the tasks that must be completed.   + Plan for human resource allocation. |
| Pick a distribution | Picking a distribution involves:   * Determining whether the computer should be a server or a workstation. Most distributions can be either, but some are better designed for specific functions and even specific types of servers. For example:   + The SUSE Linux Enterprise Server is optimized to be used as a server.   + The SUSE Linux Enterprise Desktop is optimized to be used as a desktop. * Determining whether end users will be comfortable with the distribution, or whether configuration changes might be necessary. * Determining whether required software is available for the distribution. Make a list of applications and ensure that they are provided on the distribution. * Ensuring that the distribution has the necessary support. |
| Determine the hardware requirements | Some computer hardware is incompatible with some distributions. Ordering hardware without first ensuring that the operating system can actually be installed can lead to a stack of useless hardware. Ensure that:   * The computer's hardware is on the distribution's hardware compatibility list. * The computers have sufficient CPU speed, memory and other system requirements to run the distribution and the installed software. * The correct version of the distribution is installed on the computer based on the computer's CPU architecture. These include:   + x86 for 32 bit CPUs   + x64 for 64 bit CPUs   + IA-64 for Itanium CPUs   + ALPHA for Alpha CPUs   + PPC for Power PC (Apple) CPUs |
| Plan the file system | The file system determines how a computer's files are organized on a hard drive. Linux supports several file system types that have different characteristics, including:   * *ext2* has volume integrity features that may take several minutes to run after a system crash. * *ext3* uses journaling to ensure that only incomplete transactions are checked after a system crash. It is the default file system on most distributions. * *ReiserFS* also uses journaling and also implements additional security features based on its file structure.   Pick the one that best meets your organization's needs. Considerations include:   * Maximum volume size * Maximum file size * File name size * Permissions and file security * Encryption support * Recovery support and speed * Backup support * Journal support |
| Plan the partitions | Partition planning is another element of file system planning. An efficient strategy is to create multiple partitions based on the types of files held on the partition and the user access needs to the data. Consider creating separate partitions for the following directories. Set mount options based on the type of files in the directory.   * **/**(root) needs to be at least 4 GB, but should be much larger. The partition holding the root directory should be on a primary partition and must be formatted with a Linux filesystem. * **/home** should be a minimum of 5 - 10 GB, or as large as needed to store the user files. * **/boot** should be 100 - 200MB. It needs to be in the first 1024 cylinders of the disk for older BIOS versions. * **/opt** should be at least 1 GB based on the number of applications that will be installed. * **/tmp**should be 1GB. Temporary files are cleaned out by the operating system periodically. * **/usr** should be 5 GB - 16 GB. based on the number of installed packages. * **/var** should be 3 - 10 GB. Make it large enough that log file size does not affect the rest of the computer. * **/swap**should be 1 - 1.5 times larger than the amount of installed RAM. Linux can use either a swap file or a swap partition for the swap area. Whenever possible, create a separate swap partition.   **/etc**, **/bin**, **/sbin**, **/lib**, **/dev**,and**/proc** must all be on the same partition. These directories have system configuration files that are necessary for Linux to function properly. |
| Identify software | Determine which software packages need to be installed, and only install those packages. This ensures that system resources are conserved, and that vulnerabilities are limited. |
| Identify the users | Determine the users who will use the computer. Consider the following:   * Ensure correct name spelling for the users. * Determine whether users log in locally or over the network. * Have a list of groups to which the users will belong.   The root user is always installed. Use this account only when necessary to ensure security. |
| Gather network information | Gathering network information includes the following types of information:   * IP address * Default gateway * Subnet mask * Server information for DNS servers, mail servers, and other network servers * Network topology information such as domain names * Naming conventions for servers and workstations * Domain names |
| Select an installation source | Installation sources include:   * CD * DVD * Network share   **Note:** Determine whether to install the distribution locally, or over the network as well. |

**Linux Installation Facts**

The following table describes the general steps necessary to install a Linux operating system:

|  |  |
| --- | --- |
| **Step** | **Description** |
| Start the install | In many cases, this step involves booting the computer from installation media. The BIOS may need configuration to boot from the media. Linux installers often offer different installation options, such as:   * New install * Reinstall * Upgrade * Installation recovery * Reduced feature installation |
| Select a language | The selected language becomes the default language for all users; however, it can be changed later, if necessary. |
| Set the system time | The system time can be set to local time or Coordinated Universal Time (UTC) time.   * For networks dispersed over multiple time zones, choosing UTC simplifies administration tasks. * Some distributions have options to synchronize the time over the network. |
| Format partitions | A *partition* is a logical division of a storage device associated with a hard disk drive. Create the partition structure according to the specifications in your installation plan. Determine the directory structure and file system type for each partition. This may require initializing a hard drive. Partitioning options can include:   * **Use all space**makes a single partition from the entire disk drive. * **Replace existing Linux system(s)**re-partition**s** all the previously installed Linux Operating systems. * **Shrink partitions** reduces the size of an existing partition making room for additional partitions. * **Use free space** creates a partition from unpartitioned space on the disk drive. * **Create custom layout** manually creates partitions according to the specific needs of the system or administrator.   Keep the following in mind when creating partitions:   * Initializing a hard drive removes all existing data. * A *swap* partition is required for virtual memory. Swap partitions should be between one to two times as large and the amount of RAM on the computer. * Linux computers can only have four partitions, but a single extended partition can be sub-divided into additional partitions. * Create separate partitions for the following directories to keep logs or abnormally large user files from taking all disk space, and make recovery of data easier if the operating system crashes.   + **/home** (user directories)   + **/opt**(installed software)   + **/var** (log files)   The operating system can be reinstalled on the root partition (/) and the others can then be remounted, with no loss to data. |
| Select applications and services | Installing applications and services depends on the role of the system. Applications and services include the following:   * The *boot loader* determines which operating system boots by default if more than one operating system exists on a computer. * *Package patterns* include packages necessary for a specific computer role, such as Graphical Desktop or Web Server. * *Package repositories* are locations on the Internet where software packages are maintained. Specific Linux utilities search and install software automatically from these package repositories. |
| Set the root password | The secure password for the root user (and any other user) should typically:   * Include 8 characters or longer (longer passwords are harder to crack). * Include the use of numbers and symbols in addition to letters. * Do not include a username or a dictionary word (or common variations). |
| Specify a host name | The name of the computer identifies the computer on a network. A domain may be required. |
| Configure network connections | Configure the network connections. |
| Configure services | Occasionally services must be configured based on the role of the system, such as a Web Server. |
| Add new users and groups | Create user accounts and groups for the users who will use the computer.   * Installations usually require at least one standard user account. * Network login options enable the system to access a server for login information rather than maintaining local authentication information. |
| Configure the hardware settings | Hardware configuration settings might require appropriate drivers or language settings. |

**Localization Facts**

A *locale* is a set of files that Linux uses to determine country and language-specific settings for various applications. Locales:

* Determine the way data displays on a computer. This includes:
  + The language and encoding of the text displayed on screen
  + Character classes
  + Sort order
  + Number formatting
  + Currency type and format
  + Date and time display
* Use configuration files that are part of the system library and are located in **/usr/share/locale** on most distributions.
* Use language codes specified in ISO-639, and county codes specified in ISO-3166.
* Use the following command format: *Language\_territory.codset modifier*. Examples of locale codes with modifiers include **en\_GB.UTF-8** and **de\_DE.euro**.

Environment variables implement the locale codes. For example **LANG=en\_US.UTF-8** specifies that the computer uses US English with a UTF-8 encoding when displaying information. The following table lists the configurable environment variables.

|  |  |
| --- | --- |
| **Variable name** | **Explanation** |
| LANG | LANG defines all locale settings at once, while allowing further individual customization via the LC\_\* settings below. When LANG=C, programs display output without passing it through locale translations. This is helpful when the output is being corrupted by the locale, and will help avoid some types of problems, such as when using pipelines and scripts that pass on a program's data to another program in binary form. **Note**: Localization support is the responsibility of the program's author. Many programs only support one language or a small subset of languages. |
| LC\_CTYPE | LC\_CTYPE defines the character handling properties for the computer. This determines whether characters are recognized as alphabetical, numeric and so on. This also determines the character set used, if applicable. |
| LC\_MESSAGES | LC\_MESSAGES specifies localizations for applications that use a message-based localization scheme. |
| LC\_COLLATE | LC\_COLLATE defines the alphabetical ordering of strings, such as the output of sorted directory listings. |
| LC\_NUMERIC | LC\_NUMERIC defines formatting for numeric values that are not monetary. It affects things such as the thousands separator and the decimal separator. |
| LC\_MONETARY | LC\_MONETARY defines currency units and formatting of currency type and numeric values. |
| LC\_TIME | LC\_TIME defines formatting for dates and times. |
| LC\_PAPER | LC\_PAPER defines the default paper size. |
| LC\_NAME | LC\_NAME specifies personal name format. This includes things like whether the surname comes first or last. |
| LC\_ADDRESS | LC\_ADDRESS is used for address formatting. |
| LC\_TELEPHONE | LC\_TELEPHONE defines telephone number format. |
| LC\_MEASUREMENT | LC\_MEASUREMENT determines what measurement units are used. |
| LANGUAGE | LANGUAGE is used as an override for LC\_MESSAGES. |
| LC\_ALL | LC\_ALL is a special variable for overriding all other settings. It sets all locales to the same setting. |

The following table lists and describes the command and options to configure locale settings:

|  |  |
| --- | --- |
| **Use...** | **To...** |
| **locale** | Display the current locale settings for the computer. Be aware of the following options:   * **charmap** displays the character encoding. * **-a** lists all installed locales. * **-m** lists all installed character encoding options. |
| **iconv** | Convert encoding from one encoding type to another. Be aware of the following options:   * **-f** specifies the old encoding type. * **-t** specifies the new encoding type. * **-o** specifies the input and output file. |

**GRUB Facts**

The Grand Unified Boot Loader (GRUB) is a utility that boots a Linux kernel, or any other operating system. GRUB:

* Has two stages:
  + Stage 1 is the information stored in the master boot record. It holds the location of the boot information.
  + Stage 2 is the operating system software located on the boot partition.
* Uses a menu to allow the user to select between multiple operating systems.
* Loads a default operating system after a specified time period.
* Creates a GRUB root using the **/boot**directory. The GRUB root contains the installed Stage 2 files.
* Omits**/boot** from the path if the **/boot** directory is in a separate partition.
* Uses the **hd#,#** syntax to specify the location of the root file system.
* Is installed by most distributions. If it was not installed, use the **grub-install** command from the shell prompt to install GRUB. For example, use **grub-install --root-directory=/boot sd1** to install GRUB on the second hard drive.
* Uses one of the following as the configuration file:
  + **/boot/grub/menu.lst**
  + **/boot/grub/grub.conf**

The following table describes several common options in the configuration file:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **default** | Specify the operating system that boots as the default. The value may be the entry number (e.g., 0, 1, 2) or the name of the entry (e.g., Fedora). | **default=0** boots the first operating system in the menu. **default=Fedora** boots the operating system with Fedora as the title. |
| **timeout** | Set the number of seconds GRUB waits before automatically booting the default operating system. | **timeout=10** waits 10 seconds before booting the default operating system. |
| **gfxmenu splashimage** | Specify the image file which is displayed for the graphical boot menu. | **splashimage=(hd0,0)/grub/splash.xpm.gz** identifies the default splash image. |
| **hiddenmenu** | Disable/Enable the menu that lists the operating system options.  **Note**: Adding the pound (#) symbol to this line only prevents the operating systems from being displayed on startup. If the user hits any key before GRUB selects the default operating system, the operating systems are displayed. | **hiddenmenu** disables GRUB from displaying the available operating systems.  **#hiddenmenu** displays the available operating systems. |
| **title** | Specify the title a user sees in the menu. | **title Fedora**displays an option that is named Fedora. |
| **root** | Specifies the location of the root file system. | **root (hd0,0)** specifies the first partition on the first hard drive as the root file system. **root (hd0,1)** specifies the second partition on the first hard drive as the root file system. **rootnoverify (fd0)** specifies the floppy drive as the root file system and that it should be loaded regardless of whether GRUB recognizes it. |
| **kernel** | Specifies the kernel for the entry and kernel options. | **kernel /vmlinuz-2.6.33.3-85.fc13.i686** |
| **initrd** | Specifies the initial RAM disk (initrd image) file. | **initrd /initramfs-2.6.33.3-85.fc13.i686.img** |
| **chainloader** | Specify the number of sectors to be read. | **chainloader +1** specifies that GRUB should read one sector. |
| **password** | Require authentication for the options in the GRUB menu. To create an encrypted password:   1. In a shell prompt, type **grub-md5-crypt**. 2. At the prompt, type and confirm the password, then press return. 3. Copy the hashed output from the shell. 4. Type **password --md5** *hashed\_output* in the GRUB configuration file. | **password  $3cur3** sets the string *$3cur3* as the password. **password --md5 $1$frLco/$/E4pglv5halSBQadGQgb1**uses the hashed output for the password. |
| **lock** | Prevent unauthorized boot of the operating system. When the **lock** keyword is present for an operating system, the password is required before the user can select and boot an operating system. |  |

**GRUB2 Facts**

The GRUB2, the updated version of the Grand Unified Boot Loader (GRUB) utility, is any version of GRUB 1.98 or later. Earlier versions of GRUB are sometimes known as GRUB Legacy. Be aware of the following details about GRUB2:

* **/boot/grub/grub.cfg** is the configuration file for GRUB2. The configuration file:
  + Is similar to GRUB Legacy's**/boot/grub/menu.lst**.
  + Should *not* be edited directly.
* The **update-grub** command generates the **/boot/grub/grub.cfg** file. Specifically, the **update-grub** command uses the **/etc/default/grub**file and the **/etc/grub.d/**directoryto generate the **/boot/grub/grub.cfg**configuration file.
* The **/etc/grub.d/** directory holds script files that are read when the**update-grub** command is used. Important script files in this directory include:

|  |  |
| --- | --- |
| **Script File** | **Description** |
| **00\_header** | Sets initial appearance items such as the graphics mode, default selection, timeout, etc. These settings are typically imported from the **/etc/default/grub**file. |
| **05\_debian\_theme** | Sets the GRUB2 background image, text colors, selection highlighting, and themes. |
| **10\_linux** | Identifies kernels on the root device for the operating system in use and creates menu entries. |
| **30\_os-prober** | Executes*os-prober*to search for other operating systems (i.e., Windows, Linux, etc.,) and place the results in the GRUB2 menu. |
| **40\_custom** | Allows for custom menu entries which are imported directly into **/boot/grub/grub.cfg** without any changes. |

* The **/etc/default/grub** file is the primary configuration file for changing menu display settings. The following table describes several common options in the configuration file:

|  |  |  |
| --- | --- | --- |
| **Option** | **Description** | **Examples** |
| **GRUB\_DEFAULT** | Sets the default menu entry. Typical entries include:   * + Numeric (i.e., 0, 1, 2, etc.)   + Complete menu entry quotation (i.e., "Ubuntu, Linux 2.6.31-9-generic") | **GRUB\_DEFAULT=0**sets the first menu entry as the default. **GRUB\_DEFAULT="Ubuntu, Linux 2.6.31-9-generic"**sets a menu entry as the default. |
| **GRUB\_SAVEDEFAULT** | Automatically sets the last selected OS from the menu as the default OS on the next boot.  **Note**: **GRUB\_DEFAULT=saved** is required for this option to work correctly. | **GRUB\_SAVEDEFAULT=true** sets the last selected OS from the menu as the default OS on the next boot. |
| **GRUB\_HIDDEN\_TIMEOUT** | Determines how long a screen without the GRUB 2 menu will be displayed. Options include:   * + **0** immediately boots to the default OS   + **X** (an integer value) pauses and shows a blank screen for X seconds. If a user presses any key, the GRUB menu is displayed.   + (blank) uses the value specified in the GRUB\_TIMEOUT entry | **GRUB\_HIDDEN\_TIMEOUT=0**immediately boots to the default OS. **GRUB\_HIDDEN\_TIMEOUT=3**displays a blank screen for 3 seconds and then boots to the default OS if there is no user interaction. |
| **GRUB\_HIDDEN\_TIMEOUT\_QUIET** | Displays a counter (countdown). Options include:   * + **true** does not display a counter   + **false** displays the counter for the duration specified in the GRUB\_HIDDEN\_TIMEOUT entry | **GRUB\_HIDDEN\_TIMEOUT\_QUIET=true**does not display a counter. |
| **GRUB\_TIMEOUT** | Determines how long to wait for user interaction before booting into the default operating system. Options include:   * + **X** (an integer value of 1 or higher) sets the display duration   + **-1** causes the menu to display until the user makes a selection.   Be aware of the following:   * + The GRUB 2 menu is hidden by default unless another OS is detected by the system.   + If there is no other OS, this line may be commented out unless the user changes it. | **GRUB\_TIMEOUT=4** causes the menu to display for four seconds and then boots into the default operating system. **GRUB\_TIMEOUT=-1** causes the menu to display until the user makes a selection. |
| **GRUB\_CMDLINE\_LINUX** | Adds entries to the end of the 'linux' command line (GRUB Legacy's "kernel" line) for both normal and recovery modes. It is used to pass options to the kernel. |  |
| **GRUB\_GFXMODE** | Sets the resolution of the graphical menu (i.e., the menu text size).  Multiple resolutions may be specified if they are separated by commas. | **GRUB\_GFXMODE=640x480** sets the resolution to 640 x 480. |
| **GRUB\_INIT\_TUNE** | Plays a single beep just prior to the GRUB2 menu display |  |
| **GRUB\_BACKGROUND** | Sets the background image during GRUB2 menu display. The full path should be used. | **GRUB\_BACKGROUND=/usr/share/images/back.png**displays *back.png* as the background image. |
| **GRUB\_DISABLE\_OS\_PROBER** | Enables/disables the os-prober check of other partitions for operating systems, including Windows, Linux, etc., during execution of the **update-grub** command. If the os-prober is enabled, operating systems found will be placed in the GRUB2 menu. | **GRUB\_DISABLE\_OS\_PROBER=true** disables the os-prober. **GRUB\_DISABLE\_OS\_PROBER=false** enables the os-prober and will add found operating systems to the GRUB2 menu. |

**Note**: Use the **grub-install -v**or **grub2-install -v** command to determine which version of GRUB is installed.

**Shutdown Facts**

Turning off the power without executing the proper shutdown procedure to a computer can result in data loss and filesystem corruption. Linux provides several different shutdown options. The table below shows common commands for shutting down the system.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **shutdown -h now  halt** **init 0** | Shut the system down immediately.   * **-h** specifies that the system halt or poweroff after shutdown. * **now**forces the system to shut down without a delay.   **Note**: Any of the three commands shut the system down. |  |
| **shutdown -r now  reboot init 6** | Shut the system down immediately and reboot.  **Note**: Any of the three commands reboot the system. |  |
| **shutdown -h** *time message* **shutdown -r** *time message* | Shut the system down in the designated amount of time and send a message. | **shutdown -h +5** **System is going down**sends a message and shuts the system down in five minutes.  **shutdown -h 22:00** shuts the system down at 10:00 pm.  **shutdown -r +15** reboots the system in 15 minutes.  **shutdown -r 24:00** **System is going down at Midnight**sends a message and reboots the system at midnight. |
| **shutdown -c**  Ctrl+c | Terminate the shutdown process. |  |
| **shutdown -rf***time* | Reboots the system and skips the **fsck** utility on reboot. The **-f** parameter stands for *reboot fast*. | **shutdown -rf**reboots the system and skips **fsck**.  **shutdown -r +15** reboots the system in 15 minutes and uses**fsck**. |
| **shutdown -k***message* | Sends a warning message, but does not shut down the system down. If used in combination with**-h** or **-r**, it will terminate the shutdown process after the message is sent. | **shutdown -k Please log out of the system**sends a message but does not shut the system down. |
| **shutdown -a** | Use the **/etc/shutdown.allow** file to shut down the system.  The most common use of this switch is to edit the **/etc/inittab** file and add the **-a** switch to the CTRL-ALT-DELETE section. When the switch is present, shutdown reads the **/etc/shutdown.allow** file:   * If a listed user or root is logged into the system, the system shuts down. * If a listed user or root is *not* logged into the system, shutdown is not allowed. * If the **/etc/shutdown.allow** file does not exist, there are no restrictions on who can shut down the system. |  |

When using the **shutdown** command to shut down the computer, the system does the following:

1. Sends a SIGTERM message to open programs to allow them to close.
2. Notifies logged on users that the shutdown process has initiated and the length of time before shut down.
3. Blocks users from logging into the system.
4. Uses init and **/etc/inittab** to shut down processes and the system.

Su –

# Switch user admin

Cd /boot/grub/

# Change directory to grub

Ls

# Show stuff

**Runlevel and Init Facts**

A *runlevel* is collection of services that defines a specific system state. For example, Microsoft Windows has Safe Mode and regular mode which are somewhat equivalent to Linux runlevels. The table below describes the runlevels:

|  |  |
| --- | --- |
| **Runlevel** | **Description** |
| 0 | This is the halt state. In runlevel 0, the system has no daemons in memory and is ready to be turned off. |
| 1 | This is single user mode. In single user mode, the system uses only enough daemons to allow a single user to log in, and is often used for maintenance tasks. The user is automatically logged in as the root user. |
| 2 | This is multi-user mode. In multi-user mode, the system allows multiple users to log in. It also provides networking services with the exception of the Network File System. |
| 3 | This is extended multi-user mode. In extended multi-user mode, the system provides multi-user mode support in addition to all network services, including Network File System. |
| 4 | This runlevel is undefined, but can be defined if necessary. |
| 5 | This is graphical mode. In graphical mode, the system provides the same capabilities as in extended user mode. However, the system also supports graphical log ins. |
| 6 | This is the reboot runlevel. In this runlevel, the system re-starts itself. |

During the boot process, the init (initialize) daemon loads all the other daemons that control the system. Init uses the **/etc/inittab**file to determine the default runlevel, and then starts the appropriate daemons for that runlevel; however, some distributions place code for initiating runlevels in separate files. The table below describes the format of the lines in the **/etc/inittab**file:

|  |  |
| --- | --- |
| **Field** | **Description** |
| label: | This field organizes the file to allow the init daemon to read it alphabetically. |
| runlevel(s): | This field specifies the runlevel(s) to which the line corresponds. |
| action: | This field tells init what action to take (e.g., respawn, wait, boot, bootwait, powerfail, powerwait). |
| command | This field designates a shell command to execute. |

The following lines below are typical lines in the **/etc/inittab**file:

* **id:3:initdefault:**indicates that init should set the system runlevel at 3 by default.
* **si::sysinit:/etc/rc.d/rc.sysinit** indicates that that init should execute the**/etc/rc.d/rc.sysinit** command prior to entering a runlevel when the system initializes.
* **cmd:123:wait:/sbin/custom** runs the special script file (**/sbin/custom**) for runlevels 1, 2, and 3.
* **l5:5:wait:/etc/init.d/rc 5**determines which script runs when invoking an **init** command.
* **ca::ctrlaltdel:/sbin/shutdown -r -t 4 now**specifies what happens when a user presses Ctrl+Alt+Del.

The following table describes the commands that determine and change the current runlevel. (The commands require root privileges.)

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **runlevel** | Display the previous runlevel and the current runlevel, respectively.   * The previous runlevel is the first number. * The current runlevel is the second number. * An **N** as the first number specifies that the current runlevel is the runlevel into which the computer booted. * An **S** specifies that the runlevel is single user mode (i.e., runlevel 1). | **[root@COMP ~]# runlevel** 3 5  Runlevel is 3 was the previous runlevel; Runlevel 5 is the current runlevel. |
| **init  telinit** | Change the runlevel of the computer. | **init 0** changes the system to runlevel 0, shutting the system down. **init s** changes the runlevel to 1, which is single-user mode. **init 3** changes the runlevel to 3, which is extended multi-user mode. **telinit 3** changes the runlevel to 3. **telinit 5** changes the runlevel to 5, which is graphical multi-user mode. |
| **init q init Q** | Have init re-examine the inittab file. |  |

**Kernel Option Facts**

*Kernel options* allow customization of Linux boot parameters to permit administrators to fix several problems related to booting. The following table describes common kernel parameters:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| *runlevel\_number* | Boot into a specific runlevel. | *kernel\_parameters* **3**starts the computer in runlevel three. |
| **vga** | Change monitor display settings. | **vga=0x307** sets the monitor resolution to 1280x1024 with 256 colors. |
| **init** | Change the program that the kernel starts at boot time. | **init=/bin/bash**starts the bash shell at boot time. |
| **acpi** | Enable or disable the advanced configuration and power interface (ACPI). | **acpi=off** disables ACPI. |
| **apm** | Enable or disable advanced power management (APM). | **apm=off** disables APM. |

To add kernel options while the GRUB menu is displayed:

* Use the following sequence to insert the options directly into the existing boot options:
  1. Select the operating system.
  2. Press the **a** key.
  3. Add the kernel boot options to the existing boot options.
* Use the following sequence to add the options to the options listed in the**/boot/grub/grub.conf** file:
  1. Select the operating system.
  2. Press the **e** key.
  3. Select the kernel line.
  4. Press the **e** key.
  5. Add the kernel boot options to the existing boot options.

Symbolic mode for umask

Owner: U, Group: G World: O

“service –status-all | more

**Init Script Configuration Facts**

Init script configuration is the process of specifying whether specific daemons start at a specified runlevel. Init scripts:

* Are configured differently for BSD (SUSE) and System V (Fedora and Red Hat) distributions.
* Are stored in the following locations:
  + **/etc/rc.d/init.d** directory (System V distribution)
  + **/etc/init.d** directory (BSD distribution)
* Have symbolic links that are stored in subdirectories that correspond to the runlevel under which each script should start.
* Can be started and stopped manually.
* Are started at boot using the init script.
* Have code in the scripts that determine the appropriate runlevels on which the script can operate. Configuration commands use this information to configure the appropriate levels at which scripts can start and stop. This code includes:
  + The default-start line defines the runlevels in which the script starts by default.
  + The required-start line defines services that must be running before this service can start
  + The should-start line defines the services that are recommended to start before this service starts.

Init script directories also contain other important scripts, including the following:

* *rc* (BSD and System V) switches between runlevels.
* *halt* (BSD and System V) stops and reboots the computer. It runs when the **init 0** or **init 6** commands are invoked.
* *boot* (BSD) or *rc.sysinit*(System V) is run by the init process when a computer starts. These scripts perform tasks that include:
  + Loading the kernel module
  + Checking the file system
  + Setting the system clock
* *boot.local* (BSD) or *rc.local* (System V) runs specific tasks at startup as specified by the administrator.

When using init scripts, keep the following in mind:

* At boot time, init uses the **/etc/inittab**file to determine the default runlevel, such as runlevel 5.
* When a runlevel is specified, init looks at the directory associated with the runlevel to determine what processes to start. The directory for runlevel 5 is named rc5.d. Additional directories named rc0.d through rc6.d specify what processes to start for each runlevel.
* Each rc directory contains symbolic links that point to a specific init script:
  + Link names starting with an**S** start a script for the runlevel.
  + Link names starting with a **K** kill a running process when the computer changes runlevels.
* Init follows the links and runs the scripts to start or stop processes.
* Init repeats the process using the appropriate rc directory whenever the runlevel changes.

Use the following commands to manage daemons and the init scripts:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **service***daemon\_name* | Manage the current state of a daemon. Options include:   * **start**starts a daemon that is not currently running * **stop**halts a running daemon. * **restart**stops and restarts a daemon. * **reload** requests that a daemon read and apply its configuration files without stopping. * **status**shows the status of a single daemon or daemons. * **--status-all**shows the status of all daemons.   **Note**: As an alternative method, use the absolute path to the daemon script and the option to configure the daemon (e.g., **/etc/rc.d/init.d/httpd stop**) | **service atd****start**starts the atd daemon. **/etc/rc.d/init.d/httpd start** starts the httpd daemon. **service httpd****stop**halts the httpd daemon. **/etc/rc.d/init.d/httpd restart**restarts the httpd daemon. **/etc/init.d/httpd reload**implements a new configuration for the httpd daemon without halting the service. **service httpd reload**reloads the httpd daemon. **/etc/init.d/httpd status**shows whether the httpd daemon is running. **service --status-all**shows the status of all daemons. |
| **insserv** | Configure default runlevels for a daemon on a BSD distribution. **insserv** references the INIT INFO script section of each daemon to determine the default runlevels for the daemon and dependent daemons. Be aware of the following options:   * ***script\_name***starts at the runlevels specified in the init block script code. * **-r**keeps a script from starting at any runlevel. * **-d** restores a daemon to the default runlevels defined in the scripts. | **insserv httpd** causes the httpd daemon to start at the runlevels specified in the script. **insserv -r httpd** stops the httpd daemon from starting when a computer boots. |
| **chkconfig** | Configure default runlevels for a daemon. Be aware of the following options on a System V distribution:   * **--add**adds a new service to be managed by **chkconfig**, and makes sure the service has a start or kill entry at every runlevel. * **--del** removes a service from **chkconfig** management, and removes symbolic links to the service from /etc/rc*0-6*.d. * **--level** *s*pecify the level to which a service should belong. * **--level** **on|off|reset**starts, stops, or resets the named service in the specified runlevel. The **on** and **off** options affect levels 2, 3, 4, and 5 when the runlevel is omitted. * **--list**lists of services and their runlevels.   Be aware of the following options on a BSD distribution:   * **-l**lists services and their runlevels. * **-s** *s*pecify the level to which a service should belong. | **chkconfig --add atd** starts the atd daemon. **chkconfig --del ldap** removes the ldap daemon. **chkconfig --level 5 lpd** specifies level 5 for the lpd daemon. **chkconfig --level 345 nfslock off** turns the nfslock daemon off in runlevels 3, 4, and 5. **chkconfig ypxfrd on** turns the yp transfer daemon on in levels 2, 3, 4, 5. |

**Upstart and systemd Facts**

Upstart is an event-based replacement for the /sbin/init daemon; it will start and stop tasks and services whenever the Linux system enters a specific runlevel, but will also start and stop services upon receiving information that something on the system has changed (known as an *event*).

* Upstart runlevels are as follows:
  + **0**= halt state
  + **1**= single user mode
  + **2** = graphical, multi-user mode with networking (Default)
  + **3**, **4**, **5** = same as runlevel 2, but not used
  + **6** = reboot (the system restarts itself)
* A *job* is a series of instructions that init reads which typically include a program and the name of an event. The Upstart init daemon runs the program when the event is triggered. Jobs are divided into tasks and services.
  + A *task* is a job that performs its work and returns to a waiting state when it is done.
  + A *service* is a job that does not normally terminate by itself. The init daemon monitors each service, restarting the service if it fails and killing the service when it is stopped manually or by an event.
* To run and stop a job manually, use the following commands:
  + **start**starts a daemon that is not currently running
  + **stop**halts a running daemon.
  + **restart**stops and restarts a daemon.
  + **reload** requests that a daemon read and apply its configuration files without stopping.
  + **runlevel** displays the runlevel information.

**Note**: Each of these commands are symbolic links to the **initctl** command.

* Traditionally, the default runlevel was encoded in the**/etc/inittab**file. However, with Upstart, this file is no longer used (it is supported by Upstart, but its use is deprecated).
* To change the runlevel immediately, use one of the following commands:
  + **reboot**
  + **shutdown**
  + **telinit**
* Use the **DEFAULT\_RUNLEVEL** environment variable in**/etc/init/rc-sysinit.conf** to set the default runlevel.

Systemd is a replacement for both the /sbin/init daemon and upstart, and is designed to allow services to be started in parallel at system startup. Be aware of the following systemd details:

* Systemd:
  + Is compatible with SysV and Linux Standards Base (LSB) init scripts
  + Uses socket and D-Bus activation for starting services
  + Allows you to start daemons on-demand
  + Tracks processes using Linux control groups
  + Supports snapshotting and restoring of the system state
* Control groups (i.e., cgroups) are used to track processes (instead of process IDs) for systemd. cgroup details include:
  + A cgroup is a collection of processes that are bound together by common criteria.
  + Cgroups are hierarchical. They are organized into parent-child relationships; a child group inherits parameters from its parent group.
* Use **systemctl** to manage services and runlevels. **systemctl** combines the functionality of both **service** and **chkconfig**.
* The **/etc/systemd/system/default.target** symbolic link controls the default runlevel. It links to one of the files in the **/lib/systemd/system/**directory, such as the following:
  + **default.target**(default file)
  + **multi-user.target** (runlevel 3)
  + **graphical.target** (runlevel 5)

**X Window System Facts**

X Window System is the Graphical User Interface (GUI) for Linux systems. It is a modular system that gives administrators control over the components that make up X windows. The components are listed and described in the following table.

|  |  |
| --- | --- |
| **Component** | **Description** |
| X server | The *X server* is the main component of the X Window System. The X server:   * Manages input devices such as the mouse and keyboard, and it controls output to monitors and printers. * Is networked, which means that it can be accessed locally or by other computers. * Uses the display environment variable to control output; the environment variable is **DISPLAY=0** for local systems. * Has two commonly-used implementations:   + Xwindows (also known as X.org or X11) is the most common X server.   + XFree86 is an open source version for X server. |
| Window manager | A *window manager* controls the placement and appearance of windows on a Linux computer, such as moving, hiding, resizing, or closing them, as well as controlling what they display. Most distributions come with multiple window managers. Types of window managers include:   * Enlightenment (E) * Sawfish * Flexible Virtual Window Manager (FVWM) * KDE Window Manager (KWin) * Tab Window Manager (TWM) * Window Maker (Wmaker) |
| Desktop environment | A *desktop environment* controls the desktop features including desktop menus, screensavers, wallpapers, desktop icons, and taskbars. The two most common are:   * GNU Network Object Model Environment (GNOME) * Kool Desktop Environment (KDE) |
| X font server | An *X font server* is an optional server on the network that manages fonts for client computers, but is seldom used. X font servers were developed when client systems were not powerful enough to handle font rendering without using significant CPU cycles, so a single networked server provided font rendering for the client computers.  The*fontpath* section of the **xorg.conf** file controls where a computer looks for fonts.   * An entry similar to **FontPath "unix:/7100"**specifies that the local computer is the font server. * An entry similar to **FontPath "tcp/fonts.acme-u.edu"**gives the name of anetworked font server.   **Note**: Many Linux distributions have font tools and systems that are unique to that distribution and may handle fonts differently. Using these tools is the recommended way to handle fonts. |

**X Server Configuration Facts**

Before configuring the X server, administrators must gather certain information about their hardware, such as:

* Manufacturer and model number of video board and monitor
* Video board specifications, including:
  + Amount of memory
  + Maximum resolution
  + Maximum color depth
  + Chipset
* Monitor sync rate (horizontal and vertical)

Configuration settings are contained in a file in the **/etc** directory. For the two most common X server implementations, the configuration files are:

* **/etc/X11/xorg.conf**(X.org)
* **/etc/X11/XF86Config**(XFree86)

The configuration files include the following sections:

|  |  |
| --- | --- |
| **Section Name** | **Description** |
| Files | The Files section Lists paths to files the server needs. These can include font paths, RGB color database, server modules and so forth. |
| ServerFlags | The ServerFlags section contains global server options. Many of these are for advanced configuration and debugging. |
| Module | The Module section specifies which server modules should load. |
| InputDevice | The InputDevice section has settings that give the properties for input devices. These include driver paths, device names, and options such as keyboard languages. |
| Device | The Device section holds the information the server needs to use the graphics cards. |
| VideoAdaptor | The VideoAdaptor section has description for X video. |
| Monitor | The Monitor section holds monitor specific information such as the sync rate and the refresh rates. |
| Mode | The Mode section holds settings for using different monitor modes, if necessary. |
| Screen | The Screen section has configurations for resolution , color depth, and so on. |
| ServerLayout | The ServerLayout section holds information that binds all input and output devices together to form a complete configuration. |
| DRI | DRI (Direct Rendering Infrastructure) has specifications for video rendering and 3d acceleration. |
| Vendor | The Vendor section allows vendors to create vendor-specific configuration settings. |

While administrators can edit the configuration file sections manually, it is recommended to use the configuration utilities that come bundled with the distribution. The following table describes utilities used with the X server:

|  |  |  |  |
| --- | --- | --- | --- |
| **X Server Type** | **Utility** | **Description** | **Command** |
| X.org | YaST | Yet another Setup Tool (YaST) is used in openSUSE. | **yast2** |
| SaX | SaX is a SUSE configuration tool similar to YaST. | **sax2** |
| system-config-display | Fedora and Red Hat computers running X.org use this tool for configuring video board, and monitor settings. | **system-config-display** |
| system-config-keyboard | Fedora and Red Hat computers running X.org use this tool for configuring keyboard settings. | **system-config-keyboard** |
| system-config-mouse | Fedora and Red Hat computers running X.org use this tool for configuring mouse settings. | **system-config-mouse** |
| Xorg | Xorg is a command-line tool that creates an initial xorg.conf file. It detects hardware and creates a basic configuration file for it. This tool is used when no graphical environment is present. | **Xorg -configure** |
| xorgconfig | Xorgconfig is a command-line tool that scans your hardware and configures it. You can also use it to change settings on hardware devices. This tool is used when no graphical environment is present. | **xorgconfig** |
| xorgcfg | Xorgcfg is a graphical tool that can make changes on a computer that has a running graphical environment. |  |
| **xorgcfg** |  |  |  |
| XFree86 | Xconfigurator | Xconfigurator sets up the necessary configuration files and file links to use XFree86 on a Red Hat system. Use **--hsync and --vsync** to set the horizontal and vertical sync settings. | **Xconfigurator** |
| XFree-86 | XFree-86 is a command-line tool that creates an initial XF86Config file. It detects hardware and creates a basic configuration file for it. This tool is used when no graphical environment is present. | **XFree-86 -configure** |
| xf86config | Xf86config is a command-line tool that scans your hardware and configures it. You can also use it to change settings on hardware devices. This tool is used when no graphical environment is present. | **xf86config** |
| xf86cfg | Xf86cfg is a graphical tool that can make changes on a computer that has a running graphical environment. | **xf86cfg** |
| xvidtune | Xvidtune enables fine tuning of the monitor settings in XFree86 computers. It is also compatible with and used on X.org Xservers. | **xvidtune** |

When configuring the X server, keep the following in mind:

* Always test the configuration before implementing it using the testing tools associated with the editing tool.
* Restart the X server to implement the changes. Use one of the following methods:
  + Restart the computer
  + Log out and in
  + Press Ctrl-Alt-Backspace
* Type **startx** in a command line to start an X server if no graphical interface is running.
* If manual edits to the X server configuration file is unavoidable, make a backup of the old file before making the changes.

**Window Manager and Desktop Environment Configuration Facts**

When installing a Linux distribution, the installation utility often installs several window managers, but only one desktop environment. The following steps describe how to configure a new window manager or desktop environment on a System V distribution:

1. Download the Red Hat Package Manager (RPM) package from the Internet or get the installation files from the distribution medium.   
   **Note**: Ensure that you have the correct package version for the CPU architecture and distribution.
2. Install the files using the RPM utility.
3. Log out of the computer. When logging back in, find the tool for selecting installed window managers and desktop environments.
4. Modify the hidden file to make the new window manager or desktop environment the default, by replacing the **$WINDOWMANAGER** value with the name of the window manager or desktop environment you want to use.
   * In runlevel 3, modify the hidden file **.xinitrc**.
   * In runlevel 5, modify one of the following hidden files: **.Xsession**, **.Xdefaults**, or **.Xclients**.
5. Restart the X server.

Use the following commands to start a specific application if the system does not have a graphical interface enabled.

|  |  |
| --- | --- |
| **Use...** | **To launch the...** |
| **exec startkde** | KDE desktop environment |
| **exec gnome-session** | Gnome desktop environment |
| **exec fvwm** | Flexible Virtual Window Manager |
| **exec** **sawfish** | Sawfish window manager |
| **exec** **wmaker** | Window Manager |
| **exec** **twm** | Tab Window Manager |

**Display Manager Facts**

The display manager is a modular tool that manages the graphical display on a computer and has functions that provide users with a graphical login prompt to log in to a Linux computer from a remote computer. The main display manager types for Linux are:

* X Display Manager (XDM)
* Gnome Display Manager (GDM)
* KDE Display Manager (KDM)

The following table describes different tasks when configuring the display manager:

|  |  |
| --- | --- |
| **Task** | **Description** |
| Select the default display manager | Selecting the default display manager depends on the distribution.   * Some distributions use a configuration file that is often located at **/etc/sysconfig**.   + The name and often the location of the file vary by distribution; use the **grep** utility to find a file that has DISPLAYMANAGER text in it.   + The entry should look similar to **DISPLAYMANAGER="KDM**". * Some distributions use a script in the **/etc/init.d** directory.   + The script is named xdm, kdm, or gdm depending on the distribution.   + Replace the script to change the display manager. |
| Start or stop the display manager | If at runlevel 3, you might need to manually start or stop the display manager. Using a shell prompt, navigate to the **/etc/init.d/** directory and use one of the following commands:   * **[xdm] [kdm] [gdm] start** * **[xdm] [kdm] [gdm] stop** |
| Enable or disable the display manager at startup | To permanently enable or disable the display manager at startup, use one of the following commands:   * **chkconfig [xdm] [kdm] [gdm] on** * **chkconfig [xdm] [kdm] [gdm] off** |
| Change the color depth | To change the color depth of the display manager, use a setting in the servers file for the type of display manager in use.   * The **startx** command can change color depth by using the**-depth** option, followed by the color depth in bits. * Lower color depth conserves video memory for a better resolution (increased color depth can sometimes affect the maximum resolution of older video cards). * For example, to start a session in 16 bit mode, use the following command:   **startx -depth 16** |
| Support X Terminals | An X Terminal is a system which typically connects only to an X server and does not have its own local system resources. To support X Terminals, complete the following general steps:   1. Select and configure a display host to listen connection requests on the XDMCP protocol. 2. Restart the display manager. 3. Make sure port 177 is allowed through the firewall. 4. Specify the computers or domain which can make connection requests to the display host. |

**XDM Configuration Facts**

The X Display Manager (XDM) is an X Window system display manager that allows users to log in to a Linux computer locally or from another computer on the network. XDM configuration files are often located at one of the following paths:

* **/etc/X11/xdm**
* **/usr/X11R6/lib/X11/xdm**

The following table describes the XDM configuration files:

|  |  |
| --- | --- |
| **File** | **Description** |
| Xresources | The Xresources file modifies the look of the login display including colors, border size, and similar functions.   * The*xlogin\*Foreground:* line controls the color used to display the foreground. * The*xlogin\*Background:* line controls the color used to display the background. * The*xlogin\*greetColor:* line controls the color used to display the greeting. * The*xlogin\*failColor:* line controls the color used to display the failure message. * The*xlogin\*login.Font:* line controls the font used to display the input typed by the user. * The*xlogin\*login.greetFont:* line controls the font used to display the greeting. * The*xlogin\*login.promptFont:* line controls the font used to display prompts. * The*xlogin\*login.failFont:* line controls the font used to display the message when the authentication fails. * The*xlogin\*greeting:* line changes the greeting that displays when someone logs on to the server. * The*xlogin\*namePrompt:* line displays the string displayed to prompt for a user name. * The*xlogin\*fail:* line controls the message displayed when the authentication fails. |
| Xservers | The Xservers file lists the location of the server. When configuring an Xservers file, be aware of the following:   * The server can be on the local computer or on a remote computer. * Each specification consists of the following parts:   + Display name   + Display class   + Display type   + A command line to start the server (if on a local computer) * A **-nolisten tcp** flag in the line prevents the X server from listening on TCP ports, and renders it inoperable. * If specifying a remote computer, it must be configured with appropriate permissions. * Use the**-bpp** parameter to change the color depth. Options include:   + 8 (256 colors)   + 16 (65,536 colors)   + 24 bits (16,777,216 colors)   + 32 (over 4.2 billion colors) bits.   The following command sets the color depth to 24 bits:  **:0 local /usr/bin/X11/X vt7 -bpp 24** |
| xdm-config | The xdm-config file lists the names and locations of other configuration files. The file contains a command that defines general permissions and access, such as the following:**DisplayManager.requestPort: 177**   * If the request port is set to 177, XDM listens to incoming XDMCP requests. This is the port on which the display manager protocol usually operates. * If the request port is set to 0, XDM does not listen for incoming requests; this is one way to disable the display manager when using XDM. |
| Xaccess | The Xaccess file lists the computers which can make requests to the server using XDM and the computers which can see *chooser broadcasts*. A chooser broadcast sends out a signal that lets the user see that its display manager is enabled and that it is accepting requests. In the Xaccess file:   * An asterisk acts as a wildcard that indicates that every computer has access. * A line that consists of only an asterisk grants all computers access to the login screen. * A line followed with the term CHOOSER BROADCAST defines what computers can see the chooser broadcast. * An exclamation point at the start of the command prohibits a computer from accessing the display manager.   The following command set allows access for only those computers listed. Any computer in the acme-u.edu domain can see the chooser broadcast and has access to the login screen. The computer mycomputer.mynetwork.com can access the login screen, but cannot see the chooser access information.  \*.acme--u.edu mycomputer.mynetwork.com \*.acme-u.edu   CHOOSER BROADCAST |

Be aware of the following:

* SUSE requires the following configuration changes to the **/etc/sysconfig/displaymanager** file to enable remote logins:
  + **displaymanager\_remote\_access="yes"**
  + **displaymanager\_xsrever\_TCP\_port\_6000\_ope="yes"**
* Because XDM runs as root, it is less secure than the other display managers. If any scripts that XDM uses have security flaws, an attacker could potentially gain root access to the file system.

**KDM Configuration Facts**

The KDE Display Manager (KDM) is the display manager supported by the KDE software organization. KDM:

* Offers features such as letting users select a session type and shut down the computer from the login prompt screen.
* Uses XDM configuration files, but might use a configuration file named **kdmrc**. The kdmrc file is typically located in one of the following locations:
  + **/etc/kde/kdm/**
  + **/etc/x11/kdm/**
* Looks to XDM during installation and copies the settings for KDM to use. This is optional.
* Has a GUI tool called *Kcontrol* that can configure most KDM settings.

The following table lists the general sections of the kdmrc file and describes the most commonly used options.

|  |  |
| --- | --- |
| **Sections** | **Options** |
| General | The general section holds options that affect the overall use of KDM. Be aware of the following:   * *Xservers* specifies a path to the X server definitions file or contain X server definitions. * The format is similar to that used by XDM. |
| XDMCP | The XDMCP support section has settings that customize how users access remote computers that use KDM. Be aware of the following:   * *Enable* specifies whether kdm listens to incoming XDM Control Protocol (XDMCP) requests.   + The default value is true.   + If set to false, KDM does not accept incoming requests. * *Port* specifies the UDP port number KDM uses to listen for incoming XDMCP requests.   + The default is 177.   + If set to 0, KDM does not accept incoming requests. * *Xaccess* contains the path to the Xaccess file.   + The Xaccess file lists the computers that can access the server using XDMCP, and the computers that can see the chooser broadcasts. It might also list servers to which the requests are forwarded.   + The default path is**{kde\_confdir}/kdm/Xaccess**. |
| Xgreeter | The Xgreeter section has options to configure the look and content of the login screen. Be aware of the following:   * *GreetString* holds the string users see when logging in. An empty greeting means none at all. The following character pairs query the computer and show the values for the computer in the string:   + **%d** returns the name of the current display   + **%h** returns the local host name, possibly with the domain name   + **%n** returns the local node name, most probably the host name without the domain name   + **%s** returns the operating system   + **%r** returns the operating system version   + **%m** returns the machine (hardware) type   + **%%** returns the a single % * The default is "Welcome to %s at %n". |

**GDM Configuration Facts**

The Gnome Display Manager (GDM) is a tool that manages login functions on a local computer or over a network. GDM:

* Lets users select a session type and shut down the computer from the login prompt screen.
* Uses **/etc/X11/gdm/gdm.conf** as its main configuration file. This is often a plain text file, but some distributions might use an XML file.
* Has a GUI-based configuration tool called *gdmsetup*. This tool configures color styles, accessibility, and most GDM other functions.

The **/etc/X11/gdm/gdm.conf** file is divided into sections. The following table lists the sections and describes the most commonly configured options in each section these sections

|  |  |
| --- | --- |
| **Section** | **Description** |
| Xserver definitions | The X server definitions section tell GDM about the installed X servers. Be aware of the following options:   * *Name* specifies the name that is displayed to the user; used with chooser broadcasts. * *Command* contains the path to the server executable file. * *Flexible* specifies whether the Xserver is an on-demand server that starts when GDM receives multiple login requests. * *Chooser* specifies whether the server displays a chooser list of all Xservers running on the computer rather than a login screen.   **Note**: To define servers, use the **/etc/X11/gdm/gdm.conf** file, not the gdmsetup tool. |
| Servers | The Servers section has commands for initializing the X servers. The lines can be a path to the X server executable or the name of an Xserver definition. GDM can append additional arguments to the command for the path and server definitions. |
| Security options | The security options section has options that can implement increased or decreased security depending on the function of the server. Be aware of the following options:   * *AllowRoot* specifies whether users can log in locally using the root account. * *AllowRemoteRoot* specifies whether users can log in from a remote computer using the root account. * *DisallowTCP* specifies whether remote connections are disallowed. * *RelaxPermissions* specifies user permission levels.   + When set to 0 it only allows users to access files and directories they own. The default is 0.   + When set to 1, they can access group-owned files and directories.   + When set to 2, they can access world-writable files and directories. |
| XDMCP support | The XDMCP support section has settings that customize how users access remote computers that use GDM. Be aware of the following options:   * *Enable* specifies whether GDM can listen for XDM Control Protocol (XDMCP) requests on the UDP port specified in the Port option. * *HonorIndirect* enables chooser broadcasts when set to true. * *MaxSessions* limits the number of simultaneous sessions. * *Port* identifies the port that GDM uses to listen for XDMCP requests. The default is 177. |
| XDMCP chooser | The XDMCP chooser section has settings that determine whether users can see and select from a list of available servers. Be aware of the following options:   * *Broadcast* enables chooser broadcasts when set to true. * *Hosts* lists the names of hosts outside of the local network that display on chooser broadcast lists. |
| Greeter | The greeter section has options to configure the look and content of the login screen. Be aware of the following options:   * *ChooserButton* places a button on the screen that lets the user see a list of available Xservers. * *RemoteWelcome* holds the string that users see when they log in remotely. * *Welcome* holds the string that users see when they log in locally. The following character pairs query the computer and show the values for the computer in the string:   + **%%** returns the '%' character   + **%d** returns the display's hostname   + **%h** returns the fully qualified hostname   + **%m** returns the machine (i.e., processor type)   + **%n** returns the Nodename (i.e., hostname without .domain)   + **%r** returns the operating system version.   + **%s** returns the operating system. |

GDM provides scripts used to customize the login. The scripts are typically located at **/etc/gdm/**.

* The *Init* script runs as root when GDM initializes the login display. It lets you initialize the display further, and specify programs to run on the login screen.
* The *PostLogin* script runs just after the user types correct login credentials, but before session setup. Use it to set up the home directory, which must be done before the user starts a session.
* The *PreSession* script runs as the session is starting and some setup has already occurred. Use this script to register the session with utmp/wtmp.
* The *PostSession* script runs after the session ends. Use this script to unregister the session with utmp/wtmp.

**Accessibility Facts**

Accessibility options (also known as Assistive Technologies (AT)) allow people with tactile, audible, and visual impairments to use Linux systems.

The keyboard accessibility options (also known as AccessX) set options such as filtering out accidental keypresses and using shortcut keys without having to hold down several keys at once. The following table describes keyboard accessibility options:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Sticky keys | *Sticky keys* cause keyboard modifiers keys (i.e., Ctrl, Alt, or Shift) to "stick" when pressed. This affects the next regular key to be pressed even after the release of the sticky key. This is useful for users how have difficulty pressing multiple keys at the same time. |
| Mouse Keys | *Mouse keys* control the mouse pointer with the number keypad. |
| Slow keys | *Slow keys* require a key to be pressed for a specified time period before acceptance. This is useful for individuals who tend to accidentally press keys. |
| Toggle keys | *Toggle keys* associate sounds when the Caps Lock and/or Num Lock is on. |
| Repeat rate | *Repeat rate* affects how quickly the action associated with the key is repeatedly preformed when the key is pressed and held down. For example, if you press-and-hold a character key, the character is typed repeatedly according to the repeat rate. |
| Bounce keys | *Bounce keys* ignore fast key presses of the same key, compensating for when users accidentally press a single key multiple times. |

Visual accessibility options include the following:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Onscreen keyboard | An *onscreen keyboard*displays an image of keyboard where a user to use a mouse to select keys as if they were pressed on a real keyboard. The GNOME On-Screen Keyboard (GOK) provides the onscreen keyboard, but also helps users navigate the GNOME desktop with the use of alternative input methods, such as:   * Blowing and sipping to activate a pneumatic switch * Blinking an eye or a directed gaze with an eye tracking system * Moving the head * Contracting muscles or moving limbs |
| Mouse gesture | A *mouse gesture*allows users to configure Linux to complete a specified task when the mouse is moved in a certain pattern. |
| Screen reader | A *screen reader*reads the text on a screen including menu and button text. Popular screen readers on Linux systems include the following:   * *Orca* is a free, open source scriptable screen reader which works with the GNOME desktop. * *Emacspeak* is a free screen reader which is often bundled with text editors. |
| Screen magnifier | A *screen magnifier* creates an enlarged view of the area around the mouse pointer by default. |
| Braille devices | Linux can use the following Braille hardware devices:   * A *Braille display* is a special type of computer monitor which creates a tactile display of textual information. Many Linux text-mode applications manage Braille display with no configuration changes. * A *Braille Embosser* prints a hard copy of a text document using embossed Braille characters.   BRLTTY provides a Linux daemon that redirects text-mode output to a Braille device. |
| Desktop themes | A *desktop theme* is a preset package containing graphical appearance details. Desktop themes of an assistive nature include:   * A *high contrast* theme displays the background and text colors to improve readability. * A *large print* theme displays the text in large print to improve readability. |

**RPM Facts**

The Red Hat Package Manager (RPM) is a utility that installs application packages. RPM:

* Runs on SUSE, Red Hat, and Fedora distributions.
* Installs and configures pre-compiled, pre-configured applications, and services on the system.
* Accesses a library containing thousands of packages where the source code is built, compiled, and ready to be installed on a supported Linux architecture or distribution.
* Installs, updates, verifies, queries, and uninstalls packages.
* Uses a database stored at **/var/lib/rpm**that keeps track of all installed packages, their current status, and available updates.
* Checks for dependencies on other packages and prompts to install these packages if necessary. A *dependency* is an application's reliance on another package to perform correctly.

RPM uses a standard naming convention. Be aware of the following naming convention details:

* The syntax is *packagename***-***version***-***release***.***architecture.rpm*.
* Release numbers might contain distribution data:
  + **fcx** is for Fedora
  + **rhlx** is for Red Hat
  + **suse***xxx* is for version *xxx* of SUSE
* The architecture type specifies the processor:
  + **i386** is for any Intel 80386 or newer processor.
  + **i586** is for any Intel Pentium I or newer processor.
  + **i686** is for any Intel Pentium II or newer processor.
  + **athlon** is for any AMD Athlon processor.
  + **noarch** is for any architecture (not architecture specific)
* For example, **acroread-8.1.3-51.6.i586.rpm** means the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Package Name** | **Version Number** | **Release** | **Architecture** |
| acroread | 8.1.3 | 51.6 | i586 |

The following table lists and describes several common commands for managing RPM packages:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **rpm** | Use the Red Hat Package Manager (RPM) to manage packages. Be aware of the following options:   * **--rebuilddb**rebuilds the database indices from the installed package headers. * **--initdb**creates a new database. * **--checksig**checks the authenticity of the package. The option checks the package's digital signing key against the package to ensure it has not been altered. * **-i**installs a package. Use the entire package filename when installing. * **-h**prints hash marks as the package archive is unpacked. * **-v** displays a verbose version of the installation. * **--test**tests a package for uninstalled dependencies without actually installing it. * **--nodeps**installs the package without checking for dependencies. This is not recommended. * **--force**install the package regardless of whether, a newer version of the package is already installed, package files overwrite files from previously installed packages, or if the package replaces other installed packages. * **-e**uninstalls (i.e., erases) a package. To uninstall a package, use the package name, not the file name. If dependencies exist, the dependent packages must first be removed. * **-U**updates an installed package to the newest version. * **-F** upgrades the package, but only if an earlier version currently exists on the system. * **-q**queries the computer for information about installed packages.  **Note:** Use this with **-a** to list all packages and **-l** show the files associated with the package. * **-V**verifies that packages are free from errors by performing an MD5 checksum on the package. RPM only gives output when packages have errors. If errors are present, the command displays the error code and the file name. The error codes are:   + S indicates a problem in size of a file.   + M indicates a problem with a file's mode.   + 5 indicates a problem with the MD5 checksum of a file.   + D indicates a problem with a file's revision numbers.   + L indicates a problem with a file's symbolic link.   + U indicates a problem with a file's ownership.   + G indicates a problem with a file's group.   + T indicates a problem with the modification time of a file.   + c indicates the specified file is a configuration file.   + '**.**' in place of a code letter indicates that no error is present in that area. | **rpm --checksig acroread**checks the authenticity of the acroread package. **rpm -i BackupPC-3.1.0-3.fc9.src.rpm**installs the BackupPC package. **rpm -ihv http://rpm.sh-linux.org/rpm-fc9/target-SRPMS/BackupPC-3.1.0-3.fc9.src.rpm**installs the specified package directly from the Internet. **rpm -i --test dbus-python-0.83.0-2.fc9.src.rpm**tests the computer for uninstalled dependencies for the dbus-python package. **rpm -i --nodeps dbus-python-0.83.0-2.fc9.src.rpm** installs the package but does not check for missing dependencies. **rpm -i --force dbus-python-0.83.0-2.fc9.src.rpm** installs the package regardless of effects on other packages. **rpm -e dbus-python**removes the package form the computer. **rpm -e --nodeps dbus-python**removes the package form the computer but does not check for dependent packages. **rpm -U dbus-python-0.83.0-2.fc9.src.rpm**removes any version older than the specified version and installs the specified package. **rpm -U --replacepkgs dbus-python-0.83.0-2.fc9.src.rpm**reinstalls the dbus-python package. This option is for fixing errors. **rpm -qa** displays a list of all installed packages. **rpm -qi BackupPC** shows all available information about the BackupPC package. **rpm -q --whatrequires gmp** lists the packages that are dependent on the gmp package. **rpm -ql metacity** shows the files associated with the metacity package. **rpm -q --provides gmp** lists the functions that the gmp package provides. **rpm -q --requires gmp** lists the functions that the gmp package requires. **rpm -q --whatprovides /usr/lib/libstlport\_gcc.so** shows the package that provides the libstlport\_gcc.so file. **rpm -V BackupPC** verifies the BackupPC package. **rpm -Va** verifies all installed packages. |
| **rpm2cpio** | Convert RPM packages into a cpio archive. This is useful for extracting files from an RPM package without installing and searching for the specific files. | **rpm2cpio logrotate-1.0-1.i386.rpm > logrotate.cpio**converts the files from the **logrotate** package into a cpio archive. |

**YUM Facts**

The Yellowdog Updater Modified (YUM) is a robust utility that installs Red Hat Package Manager (RPM) packages. YUM:

* Is used on Red Hat, Fedora, and other compatible distributions.
* Checks the dependencies of the RPM packages, and automatically installs or updates any dependencies as needed.
* Keeps and updates a file that lists all packages in Internet repositories.
* Uses **/etc/yum.conf**as the configuration file. The configuration file contains:
  + URLs of RPM repositories.
  + Directories where it saves downloaded packages.
  + Locations of logs.
* Uses**/etc/yum.repos.d/** to keep a .repo file for each Internet repository. To change where Linux looks for new or updated packages, add .repo files to this directory.
* Uses **/var/usr/yum.log**as a log file to track when packages are installed, removed, or downloaded.

The following table lists and describes several common commands for managing RPM packages:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **yum** | Install RPM packages included their dependencies. Be aware of the following actions and options:   * **list**displays lists of packages. * **install**installs a package. Use the entire package filename when installing. * **list updates**displays whether updates are available for packages. * **update**updates RPM packages. * **list available**lists packages that are available to install. * **search**searches all packages for a specified term. * **info**displays detailed package information. * **provides, whatprovides**displays what packages are associated with a specific file. * **remove, erase**uninstalls a package. * **-y**bypasses confirmation prompts. | **yum list all** shows all packages in the repository and installed on the computer. **yum list javahelp2.noarch** searches for the javahelp2.noarch package in the repository. **yum list \*help\*** lists all packages in the repository that have the string help somewhere in the name.  **yum list installed mtools.i686** shows whether the mtools.i686 package is installed on the computer. **yum install BackupPC-3.1.0-3.fc9.src.rpm**installs the BackupPC package and any package dependencies. **yum install http://rpm.sh-linux.org/rpm-fc9/target-SRPMS/BackupPC-3.1.0-3.fc9.src.rpm**installs the specified package directly from the Internet. **yum list update mtools.i686**looks for an update for the mtools.i686 package and updates it if one is available. **yum update sssd.i686** updates the sssd.i686 package. **yum update** updates all installed packages. **yum list available** shows the available packages. **yum search Java**searches all package information and descriptions for the term *Java*. **yum info zuff.i686** shows information about the zuff.i686 package. **yum whatprovides /etc/updatedb.conf**shows what packages are associated with the updatedb.conf file. **yum remove kdegames.i686** uninstalls the kdegames.i686 package from your computer. **yum -y update** updates all packages without requesting confirmation prompts. |
| **yumdownloader** | Download a package without installing it. | **yumdownloader zuff.i686**downloads the zuff package, but does not install it. |
| **createrepo** | Create a repository list of RPM packages stored locally or on a network. Be aware of the following options:   * **-g** specifies an xml file for the repository. * **-x** excludes specific file globs. | **createrepo -g** **packagefile.xml**creates a list of packages using the .xml file |

**Debian Package Facts**

Debian packages are preconfigured installation packages similar to RPM packages. Debian packages:

* Are compatible with distributions that do not use RPMs, including Ubuntu, Knoppix, and Linspire.
* Use naming conventions similar to RPM naming conventions, but use a .deb file extension.
* Include dependency information.
* Are not natively compatible with RPM.

The following table lists and describes several common commands for managing Debian packages:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **dpkg** | Install Debian packages on Debian distributions. Be aware of the following **dpkg** options:   * **-i**installs a package. * **--configure**reconfigures an unpacked package. * **-r**removes the package but does not delete the configuration files. * **-P**completely uninstall the package including the configuration files. * **-p**lists information about a currently installed Debian package. * **-I**(uppercase i)or **--info**lists information about packages that are not installed. * **-l**(lowercase L) displays all packages with names that match a specified pattern. * **-L**shows the installed files for a package. * **-S**finds a package associated with specified files. * **-C**searches for packages that have been installed only partially on the system. * **B**disables packages that have dependencies on the package being removed. * **--ignore-depends**ignores dependency-checking for specified packages. * **-no-act**prevents changes from being written. * **-G**prevents a package from being installed if a newer version of the package already exists on the computer. * **-E**does not install the package if the same version of the package is already installed. * **-R** installs the package recursively.   **Note**: The **dpkg-reconfigure** command reconfigures an already installed package. | **dpkg -i docbook\_4.5-4\_all.deb** installs the docbook package. **dpkg -r docbook** removes the docbook package. **dpkg -P docbook** removes the docbook package and its configuration files. **dpkg -i docbook** displays information about the package. **dpkg -I dwm-tools\_26-2\_i386.deb** displays information about the dwm-tools package. **dpkg -l kcheckers\*** lists all packages that begin with kcheckers. **dpkg -L docbook** lists all files installed with the docbook package. **dpkg -S /usr/share/base-files/motd** shows the package associated with the motd file. **dpkg -B -r docbook** removes the docbook package and disables any package dependant on the docbook package. **dpkg -G -i docbook\_4.5-4\_all.deb** installs the docbook package if it is a newer version than a previously installed package. |
| **apt-cache** | Retrieve information about the Debian package database. Be aware of the following **apt-cache** options:   * **showpkg** displays information about a package in the database. * **stats**shows the number of packages installed, dependency information, and other package cache statistics. * **unmet** lists any missing dependencies in the package cache. * **depends**shows all of the package’s dependencies. * **pkgnames**displays whether a package is installed on the system. When the package name is left off, the command shows information for all packages on the computer. * **search**searches for a package in the cache. | **apt-cache showpkg 3dchess\_0.8.1-15\_i386.deb**shows information about the 3dchess package. **apt-cache depends 3dchess\_0.8.1-15\_i386.deb**shows dependency information for the 3dchess package. **apt-cache pkgnames 3dchess**displays whether the 3dchess package is installed. **apt-cache search kde**searches for all packages that contain kde anywhere in the name. |
| **apt-get** | Download and install packages. **apt-get:**   * Is similar to the **yum** utility on an RPM distribution. * Gets its information about the application repositories from the **/etc/apt/sources.list** file.   Be aware of the following **apt-get** options:   * **update**updates **/etc/apt/sources.list** with the latest information about available packages. * **upgrade**upgrades all installed packages to the latest versions in accordance with the information in **/etc/apt/sources.list**. * **dist-upgrade**shows all of the package’s dependencies. * **install**installs a package using the package name. The package name is not the file name. During the install, **apt-get** retrieves the most recent version of the package. * **remove**removes a specified package, but leaves the configuration files. * **--purge** removes the package and the configuration files. * **source**retrieves the latest version of the package. The command accesses the **/etc/apt/sources.list** file to determine whether the latest package version is installed. * **check**checks the package database for consistency and errors. * **clean**removes unneeded package information files and logs. This command is needed when not using the dselect utility to install Debian packages. * **autoclean**removes information files about packages that can no longer be downloaded. * **-d**downloads packages without installing them. * **-f**attempts to fix a computer with unsatisfied dependencies. Use this with **apt-get install** and **apt-get remove**. * **-m**ignores package files that cannot be accessed or located. * **-q**shows less progress information. * **-s**simulates package installation without doing an actual install. * **-y**automatically provides a *yes* response to *yes / no* questions in the package installation script. | **apt-get dist-upgrade 3dchess**shows dependency information for the 3dchess package. **apt-get install 3dchess\_0.8.1-15\_i386.deb**installs the 3dchess package. **apt-get remove 3dchess**removes the 3dchess package. **apt-get source 3dchess**determines whether a newer version of 3dchess is available, and if so, installs it. **apt-get -d install 3dchess\_0.8.1-15\_i386.deb**downloads the 3dchess package without installing it. **apt-get -f install 3dchess\_0.8.1-15\_i386.deb**tries to fix dependency issues for the 3dchess package. **apt-get -m remove 3dchess**removes the 3dchess package but ignores missing files. **apt-get -q remove 3dchess**removes the 3dchess package but shows less of the information during the process. **apt-get -s install 3dchess\_0.8.1-15\_i386.deb**tests the installation process of the 3dchess package without installing it. **apt-get -y install 3dchess\_0.8.1-15\_i386.deb**installs the 3dchess package and automatically provides a yes answer to any yes/no prompts. |
| **aptitude** | View the list of packages and perform package management tasks such as installing, upgrading, and removing packages in the Advanced Packaging Tool (APT). **aptitude** is the front-end to APT. It displays a list of software packages and allows the user to interactively pick packages to install or remove. |  |

**Shared Libraries Facts**

*Shared libraries* are code loaded into memory and reused by several different programs. This allows the program file size to be relatively small, as it will use a shared amount of library code when necessary. In addition, updated code within the shared libraries allows each applicable program to take advantage of the improvements. Shared libraries can create the following software management complications:

* Changes in the shared library can be incompatible with some of the programs that use the library.
* Programs may not be able to locate the shared library.
* A shared library can become inaccessible if it is overwritten or updated. For instance, problems may occur if two different packages that include the same shared library are installed.

There a two types of shared libraries:

|  |  |
| --- | --- |
| **Type** | **Description** |
| Dynamic | *Dynamic libraries* are not integrated into the code of the application. Dynamic libraries:   * Have a **.so** or **.so.*version*** extension (**.so** stands for shared object). * Are typically stored in**/usr/lib/** and **/usr/local/lib/**. * Can degrade program load time if the library isn't already in use by another program. * Are similar to Dynamic Link Libraries (DLLs) in Windows.   Be aware of the following management programs and files for dynamic libraries:   * **/lib/ld.so**is a program which finds and loads the shared libraries needed by a program. It also prepares the program to run and executes it. * **/etc/ld.so.conf** is a file which contains a list of colon, space, tab, newline, or comma-separated directories in which to search for libraries. Some lines in the file begin with the **include** directive which list files that are to be included as if they were part of the part of the main file. * **/etc/ld.so.cache**is a cached list of libraries found in the directories specified in **/etc/ld.so.conf**. The system uses this cached list instead of loading **/etc/ld.so.conf** every time a program runs.   Use the following methods for configuring dynamic libraries on a Linux system:   * Modify **/etc/ld.so.conf** to add the path of the libraries. * Use the**LD\_LIBRARY\_PATH** environment variable to specify additional directories to search for library files. |
| Static | *Static libraries*are integrated into the code of the application when the code is compiled. Static libraries:   * Have an **.a** filename extension. * Are used when dynamic libraries are not available. * Increase the size of the application. |

Be aware of the following library management commands:

|  |  |
| --- | --- |
| **Use...** | **To...** |
| **ldd** | Discover which libraries are used by another library (i.e., library dependencies).   * When using **ldd** to track down problems, check the complete dependency chain. * Run **ldd** as root (recommended).   Be aware of the following options:   * **-v** displays all information. * **-u** displays unused direct dependencies. * **--version** displays the version number of **ldd**. |
| **ldconfig** | Reload the library cache every time libraries are added or removed, and update the symbolic links. This creates the necessary links and cache for the most recent shared libraries found in the directories specified on the command line, in the file **/etc/ld.so.conf**, and in the trusted directories (**/lib** and **/usr/lib**). Be aware of the following options:   * **-v** summarizes the directories and files it is registering as it reloads the cache. * **-N** updates symbolic links, but does *not* update the cache. * **-n** updates the links contained in the directories specified on the command line. * **-X** updates the cache but does *not* update symbolic links. * **-f** changes the configuration file from **/etc/ld.so.conf** default. * **-C** changes the cache location for the **/etc/ld.so.cache**default. * **-r** treats a new directory as if were the root directory. This is helpful when you are recovering a badly corrupted system or installing a new OS. * **-p** displays the current library cache, including all the library directories and their respective libraries. |

**User and Group Facts**

*User* accounts control the ability to log on to a system, access resources, and perform certain actions. *Groups* provide a means of grouping users for administrative purposes such as assigning permissions to files. Be aware of the following types of users and groups:

|  |  |
| --- | --- |
| **Type** | **Description** |
| Standard user | Standard user accounts can log into the system. Standard user accounts:   * Have friendly usernames (such as *mary* or *bkaun*). An administrator must create the user names. * Have an ID of 500 or more for Fedora, or 1000 or more for SUSE. The ID is automatically assigned by the system when the account is created. |
| System user | System user accounts are created by default during the Linux installation and are used by the system for specific roles. System user accounts:   * Have names that correspond with their roles, such as *ftp* and *mail*. * Cannot be used to log into the system.   **Note**: The *root* user account is created by default and has a UID of 0; however, it can be used to log into a system and perform tasks. |
| Primary group | Primary groups (also called the private group) are created when a standard user is created. Primary groups:   * Have the corresponding user as the only member. * Are automatically made the owner of files and directories when they are created. * Are similar to any other group; however, the only difference is that the user account specifically identifies the primary group for each user. |
| Secondary groups | Secondary groups are used to manage access to files and directories. Secondary groups:   * Have friendly names (such as *sales*or *accounting*). An administrator must assign secondary group names. * Receive their membership as assigned by the system administrator. |

The user and group databases are stored in the following files:

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/passwd** | The **/etc/passwd** file holds user account information. Be aware of the following details:   * Each entry identifies a user account. * Each entry contains multiple fields, with each field separated by a colon.   The following line is a sample entry in the **/etc/passwd** file:  pclark:x:501:501:Petunia Clark:/home/pclark:/bin/bash  The fields within this line are as follows:   1. User account name. 2. Password. An x in the field indicates passwords are stored in the **/etc/shadow** file. 3. User ID number. 4. Primary group ID number (also known as the default group ID number). Typically this number matches the UID number. 5. GECOS field. This field is typically used for a description or the user's full name. 6. Path to the home directory. 7. Path to the default shell. |
| **/etc/shadow** | The **/etc/shadow** file holds passwords and password expiration information for user accounts. Be aware of the following details:   * Using the **/etc/shadow** file to separate usernames from passwords increases the security of the user passwords. * Like the**/etc/passwd** file, each entry corresponds to a user account and each entry contains multiple fields, with each field separated by a colon.   The following line is a sample entry in the **/etc/shadow** file:  pclark:$ab7Y56gu9bs:12567:0:99999:7:::  The fields within this line are as follows:   1. User account name. 2. Password.    * $ preceding the password identifies the password as an encrypted entry.    * ! or !! indicates the account is locked and cannot be used to log in.    * \* indicates a system account entry and cannot be used to log in. 3. Last change. The date of the most recent password change, measured in the number of days since 1 January 1970. 4. Minimum password age. The minimum number of days the user must wait before changing the password. 5. Maximum password age. The maximum number of days between password changes. 6. Password change warning. The number of days a user is warned before the password must be changed. 7. Grace logins. The number of days the user can log in without changing the password. 8. Disable time. The number of days since 1 January 1970, after which the account will be disabled. |
| **/etc/group** | The **/etc/group**file holds group information including the group name, GID, and group membership information. Be aware of the following details:   * Each entry identifies a group. * Each entry contains multiple fields, with each field separated by a colon.   The following line is a sample entry in the **/etc/group** file:  sales:x:510:pclark,mmckay,hsamson  The fields within this line are as follows:   1. Group name. 2. Group password. An x indicates the group passwords are contained in the **/etc/gshadow** file. 3. Group ID. 4. Group members. Contains a comma-separated list of user accounts that are members of the group. |
| **/etc/gshadow** | The **/etc/gshadow** file holds passwords for groups.  Be aware of the following details:   * Like the /etc/group file, each line corresponds to a group. * Each line consists of fields separated by colons.   The following line is a sample entry in the **/etc/gshadow**file:  sales:!:pclark:pclark,mmckay,hsamson  The fields within this line are as follows:   1. Group name. 2. Group password. The group password allows users to add themselves as members of the account.    * If the field contains a single exclamation point (!), the group account cannot be accessed using the password.    * If the field contains a double exclamation point (!!), no password has been assigned to the group account (and it cannot be accessed using the password).    * If there is no value, only group members can log in to the group account. 3. Administrators. Contains a comma-separated list of users who have authorization to administer the account. 4. Group members. Contains a comma-separated list of user accounts that are members of the group. |

**Note**: Be aware of the following details:

* User and group account information can also be managed through the following network services. Each service can query network servers for user authentication parameters, such as usernames and passwords.
  + OpenLDAP is an open source implementation of the Lightweight Directory Access Protocol (LDAP).
  + Network Information System (NIS) allows many Linux computers to share a set of username and password parameters.
  + Windows Domain allows access to a central directory database that stores security and user account information.
* The **pwck** command verifies the entries in the **/etc/passwd** and **/etc/shadow** files. Errors are displayed on the screen, and entries may be deleted to solve the errors.
* The **pwconv** command synchronizes the entries in the **/etc/passwd** and **/etc/shadow** files.

**Group Commands**

Use the following commands to manage group accounts and group membership:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **groupadd** | Create a new group. The following options override the settings as found in**/etc/login.defs**:   * **-g** defines the group ID (GID). * **-p**defines the group password. * **-r**creates a system group. | **groupadd sales** creates the *sales* group. |
| **groupmod** | Modify the existing group. Be aware of the following options:   * **-A**adds specified users from the group (SUSE distribution) * **-R**removes specified users from the group (SUSE distribution) * **-n**changes the name of a group. | **groupmod -R rsem sales** removes the *rsem* account from the *sales* group. **groupmod -n sales2 sales** renames the *sales* group to *sales2*. |
| **groupdel** | Delete a group. | **groupdel mktg** deletes the *mktg* group |
| **gpasswd** | Change a group password.   * ***groupname*** prompts for a new password. * **-r** removes a group password). | **gpasswd sales** prompts for a new group password |
| **newgrp** | Log in to a new group with the group password. | **newgrp sales** prompts for the password for the *sales* group before logging in. |
| **usermod** | Modify group membership for the user account. Be aware of the following options:   * **-g** assigns a user to a primary group. * **-G** assigns a user to a secondary group (or groups). Follow the command with a comma-separated list of groups. * **-G ""** Remove the user from all secondary group memberships. Do not include a space between the quotes.   **Note:**When assigning a user to one or more secondary groups, all existing secondary group memberships are removed before assigning the user account to new groups. | **useradd -g pmaxwell pmaxwell** assigns primary group membership for user*pmaxwell* to the *pmaxwell* group.  **usermod -G sales,mktg pmorril**removes all existing secondary group assignments for *pmorril* and makes the user account a member of the *sales* and *mktg* groups.  **usermod -G "" pmaxwell** removes the *pmaxwell* from all groups. |
| **groups** | Display the primary and secondary group membership for the specified user account. | **groups pmaxwell** displays group membership for the *pmaxwell* account. |

**Partition Management Facts**

A *partition* is a logical division of a storage device associated with a hard disk drive. A hard disk drive can have a single partition or multiple partitions. One common partitioning scheme divides a disk into primary and extended partitions.

|  |  |
| --- | --- |
| **Partition Type** | **Description** |
| Primary | A *primary* partition is one that is used to store the operating system. Primary partitions:   * Can hold operating system boot files. * Cannot be further subdivided into logical drives. * Can be formatted.   **Note**: There can be a maximum of four primary partitions on a single hard disk drive. |
| Extended | An *extended* partition is an optional partition that does not have an operating system installed on it and thus is not bootable. Extended partitions:   * Can be further subdivided into an unlimited number of logical drives. * Cannot be formatted.   **Note**: There can be a maximum of one extended partition on a single hard disk drive. |

Use the following tools to create and manage partitions:

|  |  |
| --- | --- |
| **Use...** | **To...** |
| **fdisk** | Launch the **fdisk** utility and create partitions on a hard disk. **fdisk** is an interactive utility which requires values and decisions to create partitions. Be aware of the following details:   * **fdisk** requests a beginning/ending cylinder or size when creating a partition. The size is indicated using K (Kilobytes), M (Megabytes), or G (Gigabytes). * **fdisk** uses hexadecimal codes to determine the partition type. Common hexadecimal codes include:   + 0x82 Linux swap   + 0x83 Linux partition   + 0x85 Linux extended partition   + 0x8e Linux logical partition * **fdisk -l** lists the current partition configuration on the system.   Type**fdisk *device\_name***at the command prompt to enter the**fdisk** utility. Be aware of the following options within **fdisk**:   * **l** lists the partition types supported. * **m** opens the help file. * **n** creates a new partition. * **p** displays the partition table for that device. * **q** exits **fdisk** without saving changes. * **w** writes the partition table to disk (i.e., saves the file) and exits the **fdisk** utility. * **d** deletes a partition. |
| **partprobe** | Request that the operating system re-read the partition table. The operating system kernel reads the partition table and recognizes the table changes. |

**Device File Names**

Storage devices in Linux are represented by device files.

* Device files are located in the **/dev** directory.
* The **/dev** directory contains files for all types of devices, even those that don't exist on the system.
* Not only do device files represent devices, they indicate how data is transferred to that device.
* Devices, like storage devices, that receive data in block transfers by using memory to buffer the transfers are called *block* devices. Devices that send data transfers character-by-character (like the keyboard) are called *character* devices.

The table below lists and describes the most common device files.

|  |  |
| --- | --- |
| **Device File** | **Description** |
| **/dev/sd*xn*** | *sd* files identify hard drives. A letter (beginning with *a*) follows the *sd* designation and identifies the ID of the hard drive. At the end is appended a number (beginning with 1) that identifies the partition on the drive. Examples include:   * sda2 is the second partition (2) on the hard drive with the lowest ID number (a) * sdc1 is the first partition (1) on the drive with the third lowest ID number (c) * sda1 is the first partition (1) on the hard drive with the lowest ID number (a) * sdb3 is the third partition (3) on the drive with the second lowest ID number (b) * sdc2 is the second partition (2) on the drive with the third lowest ID number (c) * sdd1 is the first partition (1) on the drive with the forth lowest ID number (d)   **Note:**Some systems will use **/dev/sr*x*** instead. |
| **/dev/cdrom** | This is a special designation used to identify the CD-ROM in the system. In reality, the **/dev/cdrom** ID is just a symbolic link to the actual device (sr0). |
| **/dev/fd*n*** | *fd* files identify floppy drives. Device numbering begins at 0. For example, **/dev/fd0** is the first floppy drive. |
| **/dev/tty*n*** | *tty* files identify local terminals on the system. Device numbering begins at 0. Subsequent terminals are represented with files that increment by one (e.g., the file for terminal two is /dev/tty2, and so on). |
| **/dev/ttyS*n*** | *ttyS* files identify serial ports. Device numbering begins at 0. Files for subsequent serial ports are represented by files that increment by one (e.g., the file for serial port two is **/dev/ttyS1**, and so on). |
| **/dev/lp*n*** | *lp* files identify parallel ports. Device numbering begins at 0. Files for subsequent parallel ports are represented by files that increment by one (e.g., the file for parallel port two is **/dev/lp1**, and so on). |
| **/dev/usb/*file\_name*** | USB devices have their own subdirectory of files to support up to 127 USB devices. |
| **/dev/psaux** | This file is for the PS/2 mouse port. |
| **/dev/st*n*** | st files identify SCSI tape devices. Device numbering begins at 0. |

**LVM Facts**

The Logical Volume Manager (LVM) provides an alternative methods to manage partitions on a Linux system. LVM gives a system administrator more flexibility in allocating storage on a system. Important aspects of LVM include:

* You can (within certain limits) resize and move logical volumes while they are still mounted and running.
* Logical volumes may be identified by using descriptive names (i.e., *research* or *marketing*) instead of physical disk names such as */dev/sda* and */dev/sdb*.

The following table describes common LVM commands:

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Examples** |
| **pvcreate** | Initializes physical volume for later use by the Logical Volume Manager (LVM). | **pvcreate /dev/sdb**creates a physical volume on the second hard disk in the system. **pvcreate /dev/sdd**creates a physical volume on the fourth hard disk in the system. |
| **pvscan** | Scans all disks for physical volumes and displays the result. | **pvscan** displays all found physical volumes on the system and their associated volume groups. |
| **vgcreate** | Creates a new volume group. | **vgcreate system /dev/sdb**creates a volume group named *system* on the second hard disk in the system. **vgcreate backup /dev/sdd** creates a volume group named *backup* on the fourth hard disk in the system. |
| **vgextend** | Adds one or more initialized physical volumes to an existing volume group to extend it in size. | **vgextend system /dev/sdc**adds the third hard disk in the system to the *system*volume group. |
| **lvcreate** | Creates a new logical volume in a volume group. Options include:   * **-L**specifies the size. Use the following size suffixes:   + K for kilobytes   + M for megabytes   + G for gigabytes   + T for terabytes   + P for petabytes   + E for exabytes * **-n**specifies the name | **lvcreate -L 20G -n data system** creates a 20 Gigabyte logical volume, named *data*, on the *system* volume group. **lvcreate -L 2T -n Storage1 backup** creates a 2 Terabyte logical volume, named*Storage1*, on the *backup* volume group. **lvcreate -L 1T -n Storage2 backup** creates a 1 Terabyte logical volume, named*Storage2*, on the *backup* volume group. |
| **lvscan** | Scans all known volume groups or all supported LVM block devices in the system for logical volumes and displays the result. | **lvscan**displays all of the logical volumes on the system. |
| **lvextend** | Extends the size of a logical volume. Options include:   * **-L**specifies the new size of volume. Be aware of the following:   + **-L +*size***specifies an increase in the size of volume.   + The omission of the**-L** option will increase the size of the logical volume by the amount of free space on physical volume. * ***logicalvolume***specifies which logical volume to extend. * ***physicalvolume*** specifies the physical volume to use for the extension. | **lvextend -L 30G data** extends the *data* logical volume to a total of 30 Gigabytes. **lvextend -L +10G data** extends the *data* logical volume by another 10 Gigabytes. **lvextend -L +10G data** **/dev/sde**extends the *data* logical volume by another 10 Gigabytes on thephysical volume for*/dev/sde*. **lvextend data** **/dev/sde**extends the *data* logical to all of the free space on thephysical volume for*/dev/sde*. |

**Quota Commands**

Disk quotas prevent a user or group from using a disproportionate amount of disk space on a volume. Quota implementations include:

* Limiting the number of files and directories a user or group can create.
* Limiting the amount of disk space a user or group can use.

Quota types include the following:

* A *soft* limit allows the user to extend the limits of the disk quota.
* A *hard* limit is a fixed limit that the user cannot modify.

The following table lists the general steps to implement quotas:

|  |  |
| --- | --- |
| **Step** | **Procedure** |
| Install quota package | To install the **quota** package (on a System V system):   1. Use **rpm -qi quota** to see if the package is installed. 2. Use **yum install***package* to install the quota package if required. |
| Edit mount options in **/etc/fstab** | Edit the **/etc/fstab** file to add the mount options for the filesystem so it can implement quotas:   * **usrquota** implements quotas for users. * **grpquota** implements quotas for groups. |
| Create quota files | Create the **aquota.user** and **aquota.group** files in the directory where the partition is mounted. |
| Enable quotas and view a quota report | Enable disk quotas, and then generate a disk usage and quota report. The report shows:   * How much space to allocate to each user. * How much space is currently in use by each user. * Whether some users are using a significant amount of disk space. |
| Edit quotas | Edit a quota for the specified user or group. Be aware of the following when editing the quotas:   * Set the soft and hard quotas for *blocks*. This limits the total amount of disk space per user or group. * Set the hard and soft quotas for *inodes*. This limits the total number of files and directories per user or group. * Users may exceed soft quotas for a number of days specified in the *grace* period (seven by default.) When the grace period expires, users cannot create additional files. * Users cannot exceed hard quotas. * When setting block quotas, 1000 blocks is about 1 MB, and 1,000,000 blocks is about 1 GB. * Setting the quota limits to 0 removes all quotas. |

The table below describes common commands for working with quotas:

|  |  |  |
| --- | --- | --- |
| **Use....** | **To...** | **Examples** |
| **quotacheck -mavug** | To create the**aquota.user** and**aquota.group** files in the filesystem (after placing the quota entries in**/etc/fstab**). Common options include:   * **-m** updates the quota database even if other processes are running on the filesystem. * **-a** updates the quota database. * **-v** runs the command in verbose mode. * **-u** and **-g** run the database updates for users and groups, respectively. | **quotacheck -mavug /home**creates the **aquota.user** and**aquota.group** files in the root (/) directory. |
| **quotaon** | Enable quotas for the mounted filesystem.   * **-a** enables all mounted filesystems listed in **/etc/mtab**. * **-v** runs the command in verbose mode. | **quotaon -av /**enables quotas for the root (/) directory. |
| **quotaoff** | Disable quotas for the mounted filesystem. | **quotaoff /home**disables quotas for the**/home** directory. |
| **repquota** | Display a summary of the disc usage and quotas for the specified filesystems, including the specific number of files and used space by user. Common options include:   * **-v** reports all quotas, even if there is no usage. * **-n**does not resolve user and group names to speed printing time . * **-u** and **-g** reports for users and groups, respectively. * **-a** gives information for all filesystems listed in **/etc/mtab**. | **repquota /home -uv**creates a user quota report for the**/home**directory. |
| **edquota** | Open and edit a user's quota, a groups quota, or change the grace period:   * **-u** changes the users quota * **-g** changes a groups quota * **-t** changes the grace period | **edquota -u** **mtomm**opens the quota file for the *mtomm* user account. |
| **quota** | Display the current user's quota:   * **-u** shows the quota for a user. * **-g** shows the quota for a group * **-v** shows current the current usage, the hard quota and the soft quota for blocks and inodes. | **quota**displays the quota report only for the current user account. **quota -u dhanson**displays the quota report only for the *dhanson*user account. |

**Ownership Commands**

When a user creates a file (or directory), the user and the user's primary group receive ownership for the file (or directory). Only a file owner and the root user can change file ownership or permissions.

The table below lists the most common commands for managing file ownership.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **ls -l** | View a long file listing. A long file listing shows the ownership for the files (among other information). | drwxr-xr-x 22 **root** **root** 4096 Jun 19 15:01 sales (Root is the file owner and the group owner for this example) |
| **chown** | Change the ownership of a file or directory. Be aware of the following options:   * **-R** changes the ownership of the file recursively throughout the directory tree. * ***user***changes the file ownership only. * ***user*:*group*** change the user and group ownership of the file. * **:*group***changes the group ownership only. * ***.group***changes the group ownership only. | **chown pmorril /sales/report**makes pmorril the user owner of the /sales/report file. **chown -R pmorril /sales** makes pmorril the owner of all files in the /sales directory (and below). **chown pmaxwell:sales /sales/report** makes pmaxwell the user owner and sales the group owner of the file. **chown :sales -R /sales** makes the sales group the owner of all files in the /sales directory. |
| **chgrp** | Change the group owner of a file or directory. | **chgrp sales /sales/report** makes the sales group the group owner of the file |

**Permission Facts**

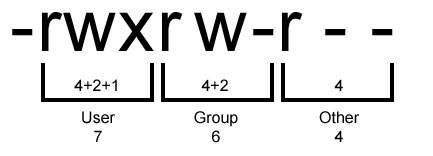
Every file has an *inode* (information node) that stores information about the file, including when the file was last modified, file size, data block location, permissions, and ownership (remember, directories are also files in the Linux system). The portion of the inode that stores permission information is called the *mode*. The mode has three sections:

* User (owner) permissions
* Group (group owner) permissions
* Other (everyone on the Linux system who is not an owner) permissions

There are three types of permissions contained in the mode, each of which is described in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Permission** | **Letter Abbreviation** | **Octal Value** | **Allowed Actions on Files** | **Allowed Actions on Directories** |
| Read | r | 4 | Open and read the file | List directory contents if the execute permission is also present |
| Write | w | 2 | Open, read, and edit the file | Add, delete, and rename files if the execute permission is also present |
| Execute | x | 1 | Execute the file (if it's a program file) or the shell script | Enter the directory and work with its contents |

Permissions are identified with either the letter abbreviation (i.e., r, w, x), or the octal number that corresponds to the permission. The following graphic shows a detailed depiction of how permissions are displayed and how they can be referenced.



Be aware of the following facts about the mode:

* A **d** preceding the permissions indicates that the object is a directory.
* Adash (-)identifies a file (the example above is for a file).
* Permissions are grouped according to user, group, or other permissions.
* If a permission has not been assigned, adash (-)takes its place in order.
* When using numbers to represent permissions, add the numbers together within each permission group. Then string the numbers together. For example, the permissions in the graphic above can be represented by the number *764*.
* The root user has all permissions to files and directories regardless of the mode settings.

The table below lists the most common commands for managing permissions:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **ls -l** | View a long file listing. A long file listing shows the permissions for the files (among other information). | **drwxr-xr-x** 22 root root 4096 Jun 19 15:01 sales (This is a directory with 755 as the permissions) |
| **chmod** | Change the permissions for the specified file. Be aware of the following syntax options:   * ***category*+*permission*** adds a permission for a user, group, or other (category) to a file. * ***category*-*permission***removes a permission for a user, group, or other from a file. * ***category*=*permission*** sets the permission equal to the permission specified for the user, group, or other for the file. * ***decimal\_value***sets the permissions for the file according to the numbers represented for each mode category. * **-R**sets the permission(s) recursively. | **chmod u+x,g+x,o+x myfile** adds the execute permission to the file *myfile* for user, group, and other. **chmod g-w,o-w myfile** removes the write permission for group and other from the file*myfile*. **chmod u=rwx myfile** grants the user read, write, and execute permission for the file*myfile*. **chmod 711 myfile**grants the user read, write, and execute permission (7) while group and other both receive execute permission (1) for the file *myfile*. |

**Umask Facts**

A *umask* changes (removes) the default file and directory permissions. By default, files receive rw-rw-rw- (666) permissions, and directories receive rwxrwxrwx (777) permissions when they are created. In most cases, the default assignment gives excessive permission to files and directories.

The umask identifies which permissions are *removed* from the default permissions when files and directories are created. The following table shows what happens when the mask is set to a value of 022.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Files (binary)** | | **Directories (binary)** | **Files  (letter abbreviation)** | | **Directories (letter abbreviation)** |
| **Default Permission** | 666 | | 777 | rw-rw-rw- | | rwxrwxrwx |
| **Umask** | 022 | 022 | | ----w--w- | ----w--w- | |
| **Result** | 644 | | 755 | rw-r--r-- | | rwxr-xr-x |

Additional examples of umask calculations are:

* A umask of 066 results in file permissions of rw-------- (600) and directory permissions of rwx--x--x (711).
* A umask of 033 results in file permissions of rw-r--r-- (644) and directory permissions of rwxr--r-- (744).
* A umask of 011 results in no changes to file permissions (the x permission is already removed by default) and directory permissions of rwxrw-rw- (766).

The table below lists the commands for managing the umask:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **umask** | View the current umask setting | 022 is the typical umask setting. |
| **umask** *number* | Change the default umask. | **umask 007** sets the umask to remove nothing from the user or group but to remove all permissions from other. |

Be aware of the following:

* The default umask value may vary depending on the Linux distribution (022 or 0022 is typically the default).
* Setting the umask with the **umask** command is only persistent for the shell session.
* To make the umask persistent through shell sessions and reboots, add the **umask** command to the shell configurationfile (depending on the distribution).

#### Special Permission Facts

Be aware of the following special permissions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Permission** | **Letter Abbreviation** | **Example** | **Octal Value** | **Description** |
| SUID (Set User ID) | **s** in the execute permission position of the user permissions | rw**s**rw-rw- | 4 | If the SUID bit is set, the program will run with the permissions of the file owner, not with the permissions of the user who runs the program.   * The most common use of SUID is to allow users to run a command as the root user. * Users do not become the root user, but rather the command or program runs as if executed by the root user. * Some programs require the SUID bit set for proper functionality. * Be careful in setting the SUID bit as it could give a program too many permissions. |
| SGID (Set Group ID) | **s** in the execute permission position of the group permissions | rwxrw**s**rw- | 2 | If the SGID bit is set:   * On a file, the program will run with the group permissions of the group owner. * On a directory, a newly-created file will receive the same group owner as assigned to the parent directory. |
| Sticky bit | **t** in the execute permission position of the other permissions | rwxrw-rw**t** | 1 | This marks the file (not directory) in such a way as to prevent the file's deletion from the system by anyone except the file owner. Setting the sticky bit works particularly well with shared files. |

Use the following commands when managing special permissions:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **ls -l** | View a long file listing. A long file listing shows the permissions for the files (among other information). | **drwsr-xr-x** 22 root root 4096 Jun 19 15:01 sales (This is a script with 4755 as the permissions; it has the SUID set.) |
| **chmod** | Assign a special permission. Be aware of the following syntax options:   * ***decimal\_value***sets the permissions for the file according to the numbers represented for each mode category.   + The special permission *precedes*the standard octal representation of a set of permissions   + Only the *first*number changes to identify the special permission group settings. * ***category*+*permission*** adds a special permission for a user, group, or other (category) to a file. * ***category*-*permission***removes a special permission for a user, group, or other from a file. | **chmod 4421** sets the SUID. **chmod u+s** sets the SUID. **chmod u-s** removes the SUID. **chmod 2421** sets the SGID. **chmod g+s** sets the SGID. **chmod 1421** sets the sticky bit.  **chmod u+t** sets the sticky bit.  **chmod u-t** removes the sticky bit.  **chmod 6421** sets both the SUID and SGID.  **chmod 7421** sets the SUID, GUID, and stickybit. |

#### Tar Facts

The **tar** (tape archive) utility takes the contents of several files and stores them as a single file. Tar:

* Uses the .tar file extension.
* Can backup entire directories or file systems.

The following table lists several options that tar uses.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **tar** | Combine multiple files into a single file. Options include:   * **-c**creates a new archive. * **-v**displays a list of all files being written into the archive. * **-f**specifies the file to create or unpack. Without this option tar uses standard input and output as the destination. * **-x**extracts the files. If no destination directory is specified, then tar extracts the files to the current working directory. * **-z**compresses/decompresses a file using the **gzip** utility (normally named with a .gz extension). * **-j**compresses/decompresses a file using the **bzip2** utility (normally named with a .bz2 extension). * **-C**changes to a specific directory to extract the files. * **-t**lists the contents of an archive. | **tar -cf /root/tarbackups/oct17backup.tar /home** writes a backup of the /home directory to the **/root/tarbackups/oct17backup.tar**file. **tar -cvf /root/tarbackups/oct17backup.tar /home** writes a backup of the /home directory to the **/root/tarbackups/oct17backup.tar**file. **tar -cvf /root/tarbackups/oct17backup.tar /home** writes a backup of the /home directory to the **/root/tarbackups/oct17backup.tar**file. **tar -xvf /root/tarbackups/oct17backup.tar -C /home** extracts the files and decompresses them to the**/home**directory. |
| **gzip** | Compress a file using **gzip**. Options include:   * **-c**writes the file to standard output. * **-d** decompresses the file. * **-l** displays information about files in an archive. * **-r** recursively compresses all files in directories and subdirectories.   **Note:** This is the same as the **tar -z**command. | **gzip file.tar** creates a compressed file and removes the original file. **gzip -c file.tar > file.tar.gz**creates a tar archiveleaving the original file unchanged.**gzip -d** **file.tar.gz**uncompresses the tar archive. |
| **gunzip** | Uncompress a file using **gunzip**. Options include:   * **-f** forces decompression even if the file has multiple links or the corresponding file already exists. * **-r** decompress all files in a directory tree. | **gunzip file.tar.gz**uncompresses the tar archive.  **gunzip file.cpio**uncompresses the cpio file. |
| **bzip2** | Compress/decompress a file using **bzip2**.   * **-d** decompresses the file. * **-k** keeps the original file unchanged.   **Note**: This is the same as **tar -j**command. | **bzip2 file.tar** creates a compressed file and removes the original file. **bzip2 -k file.tar**creates a tar archiveleaving the original file unchanged.**gzip -d** **file.tar.gz**uncompresses the tar archive. |

#### Cpio and dd Facts

Be aware of the following **cpio** and **dd** utility details:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **cpio** | Create a **cpio** archive or extract files from a **cpio** archive. The **cpio** command is used as another method to archive files, yet is different from other archive utilities because it only takes files names from standard input. Cpio:   * Copies files to an archive (copy-out mode). * Extracts files from an archive (copy-in mode). * Copies files to a different directory tree (copy-pass mode).   Cpio uses the following options:   * **-o**creates the archive by invoking copy-out mode. * **-v** invokes verbose output, showing file names as they're added to the archive. * **-i** extracts files by invoking copy-in mode. * **-u** overwrites existing files. * **-d** extracts directories during an extraction. * **-t** displays archive contents without extracting files. * **-p**copies files to a new directory (copy-pass mode). | **ls ~/4archive | cpio -ov > *filename*.cpio**creates a cpio archive from the files listed in the **~/4archive**directory. **cpio -iv < *filename*.cpio**extracts the files from the cpio archive. **ls ~/copyme | cpio -pvd ./newdirectory**copies files from**~/copyme**to**./newdirectory**. |
| **dd** | Copy information using records. The **dd** utility copies information using records instead of files. **dd** is useful for:   * Copying partitions to a single image file. * Copying a master boot record.   Parameters for **dd** include:   * **if=** specifies the input file. * **of=** specifies the output file. * **bs=** specifies the block size. * **count=** specifies the number of blocks to be copied. | **dd if=/dev/sdb1 of=/root/partition.image** copies the entire first partition if the second hard drive to a single file.  **dd if=/dev/sda of=/root/file.mbr bs=512 count=1** copies the master boot record of the first hard drive to a single file. |

#### Device Driver Facts

A *device driver* is a software component that allows a hardware device to communicate with the operating system of a computer. Drivers allow an operating system to correct interpret and implements the signals that come from the hardware device. The following table describes the two methods Linux uses to implement device drivers:

|  |  |
| --- | --- |
| **Method** | **Description** |
| Loaded as a kernel module | A kernel*module* is software that the kernel accesses only when it is needed. When in use, modules run as if they were part of the kernel and have the same access rights. Modules:   * Have an .o or .ko extension. * Are stored in the **/lib/modules/***kernel\_version***/kernel/drivers/***driver\_name* directory. * Are linked and unlinked dynamically. |
| Compiled into the kernel | When the drivers are compiled into the kernel, it is integrated into the kernel build when the kernel is recompiled. This method requires an administrator to recompile the kernel. Drivers compiled into the kernel:   * Should be limited to the hardware needed to boot the computer, such as drivers for the keyboard, mouse, and disk drive. * Increase the size and complexity of the kernel. * Requires considerable configuration expertise. * Consume additional computer resources. |

The following directories contain information about the hardware that is installed on the computer:

|  |  |
| --- | --- |
| **Directory** | **Contents** |
| **/proc** | The **/proc** directory contains information about the system state and processes. Its contents are created dynamically. Be aware of the following files and directories in the **/proc**directory:   * *cmdline*displays the boot options that were given to the kernel at boot time. * *cpuinfo* has information about the computer's CPU. * *devices* displays a list of hardware installed on the computer. * *dma* shows all the *direct memory access* assignments for the computer. Direct memory access gives hardware devices direct access the computer's memory independent of the CPU. * *interrupt* lists the interrupt request (IRQ) channels the computer uses. Interrupt requests are signals sent to the CPU that inform it that it needs to process input from a hardware device. * *iomem* contains a mapping of the memory allocated to each device and the input/output port assignments for the memory. * *modules* lists the kernel modules that the computer is currently using. * *version* gives information about the current kernel version. * *meminfo* displays detailed memory information on the system. * **/scsi** contains a file or directory for each SCSI device attached to the computer. * **/bus** contains a file or directory for each USB device attached to the computer. * **/ide** contains a file for the IDE devices attached to the computer, including the internal hard drives and other devices that attach to an IDE ribbon.   Changing the system through the **/proc** directory should be attempted only by experienced administrators. Be aware of the following facts about changing **/proc** files:   * Not every file can be modified. Some are marked read only and can only be viewed. * Do not use**vi**to view or modify files in the **/proc** directory. Instead, use the **echo**command to redirect commands to the appropriate files, or use other special commands. * Use the**cat**command or other special commands to view files in the **/proc**directory and sub-directories. |
| **/sys** | The **/sys** directory displays information about devices and drivers. Be aware of the following directories in the**/sys** directory:   * **/block** has an entry for each block device on the computer. Block devices such as flash drives and disk drives use data blocks. * **/bus** holds a sub-directory for SCSI, USB, PCI, and ISA devices. Each of these sub-directories has an additional directory for devices and drivers that has information for each device and driver in the category. * **/class** has files for each class of devices on the computer. * **/devices** lists every device that been discovered on the computer. The directory hierarchy places each device beneath the device to which it is connected. * **/module** has a sub-directory for each kernel module installed on the computer. |

Linux also includes utilities that provide extensive information about hardware configurations, including:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **lsusb** | Display information on all USB devices connected to the computer. This utility uses the following options:   * **-v** shows exhaustive information. * **-s***bus\_name*shows information for a specific bus. | **lsusb -v** shows all information about each USB device on the computer. |
| **hwinfo** | Display information about hardware on the computer. Be aware of the following options:   * **--*hardware\_item\_name*** probes for a specific hardware item. Common hardware names include:   + **bluetooth**   + **camera**   + **cdrom**   + **cpu**   + **disk**   + **dsl**   + **monitor**   + **mouse**   + **keyboard**   + **usb** * **--short** shows an abbreviated list of information. * **--listmd** displays RAID devices. | **hwinfo --cpu** shows information about the computer's CPU. |
| **lspci** | Display information for all PCI devices. Be aware of the following options:   * **-k**shows the kernel drivers that support the device. * **-t**displays a tree diagram that shows connections between all busses, bridges, and devices. | **lspci -k** shows the devices and the kernel drivers that support them. |

#### Kernel Module Management Facts

When the system boots, it uses one of the following files to automatically load kernel modules. (**Note**: The exact file used depends on the implementation.)

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/modprobe.conf** | Provides the **modprobe** utility with default commands for loading modules at boot time. Entries in the file include the following:   * **install** loads a module at boot time. * **alias** specifies a name as an alias for a module name. This alias can be used with module utilities. * **options**specifies options used while loading a module, including:   + **irq**for IRQ information   + **io**for I/O port information. |
| **/etc/modprobe.d** | Contains multiple configuration files used by **modprobe** at boot time if the**/etc/modprobe.conf** file does not exist. |

Use the following commands to manage kernel modules manually:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **lsmod** | List all loaded modules. The command formats information from the **/proc/modules**file. No options are associated with **lsmod**. |  |
| **modinfo** | See additional information about a module listed using the **lsmod** command. | **modinfo mii** shows information about the MII Hardware Support Library module. |
| **depmod** | Create a file that lists module dependencies. The file is placed at **/lib/modules/***kernel\_version\_number***/modules.dep**. Read the**/etc/modules.conf** file to identify modules. It then probes each module to identify dependencies and builds a list of those dependencies. Be aware of the following options:   * **-a**Shows information for all modules. * **-n**Shows what would happen on the screen rather but does not perform the action. * **-v**Uses verbose mode. | **depmod -an** performs the probe and display the results on the screen. **depmod -v**displays all module information to the screen as it updates the modules.dep file. |
| **insmod** | Install modules into the kernel.   * **insmod** does not look for dependencies, and fails if a module has unmet dependencies. * Include the full name of the module, including the .o or .ko extension. | **insmod mousedev.ko** loads the mousedev module. |
| **modprobe** | Load modules into the kernel along with any module dependencies. This utility also runs at startup to load modules into the kernel. The **/etc/modprobe.conf**file provides **modprobe** with its configuration rules. Be aware of the following options:   * **-l** lists all loaded modules. * **-r** removes a module. | **modprobe reiserfs** loads the reiserfs and all of its dependant modules. **modprobe -r reiserfs** removes the reiserfs module. |
| **rmmod** | Remove a module from the kernel. **rmmod**:   * cannot unload the module if it is in use. * Does not look for dependencies and can cause errors if a module depends on a module that is unloaded. | **rmmod mousedev** removes the mousedev module. |

#### Hotplug and Coldplug Device Facts

Be aware of the following device categories when managing hardware:

|  |  |
| --- | --- |
| **Category** | **Description** |
| Coldplug | *Coldplug*devices should only be removed or replaced when the power to the computer is off. Attempting to remove these devices while the power is on can damage the computer. Coldplug devices include:   * RAM (Random-access Memory) chips * CPU (Central Processing Unit) * Expansion cards, such as Peripheral Component Interconnect (PCI) or PCI Express cards * Hard disk drives |
| Hotplug | *Hotplug*devices can be removed while the computer is on. Linux uses software designed to detect these changes as the devices are added and removed. Hotplug devices include:   * USB flash drives * FireWire devices |

Linux uses the following components to manage devices:

|  |  |
| --- | --- |
| **Component** | **Description** |
| sysfs | *sysfs* is a virtual file system mounted at **/sys** which exports information about hotplug devices so that other utilities can access the information. |
| Hardware Abstraction Layer (HAL) daemon | The Hardware Abstraction Layer (HAL) daemon (hald) provides all applications with data about current hardware. hald runs constantly. |
| Desktop Bus (D-Bus) daemon | The Desktop Bus (D-Bus) daemon allows processes to communicate with each other and notify them of new hotplug devices. |
| /udev | *udev* is a virtual file system that dynamically creates device files as devices are added and removed. udev uses:   * **/etc/udev/udev.conf** as the configuration file. The configuration file contains the error reporting level for hotplug device errors. * **/etc/udev/rules.d/** to name devices. |

#### User Commands and Files

Be aware of the following configuration files when managing user accounts:

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/default/useradd** | The **/etc/default/useradd** file contains default values used by the **useradd** utility when creating a user account, including:   * Group ID * Home directory * Account expiration * Default shell * Secondary group membership |
| **/etc/login.defs** | The **/etc/login.defs** file contains:   * Values used for the group and user ID numbers. * Parameters for passwords encryption in the shadow file. * Password expiration values for user accounts. |
| **/etc/skel** | The **/etc/skel** directory contains a set of configuration file templates that are copied into a new user's home directory when it is created, including the following files:   * **.bashrc** * **.bash\_logout** * **.bash\_profile** * **.kshrc** |

Although it is possible to edit the **/etc/passwd** and **/etc/shadow** files manually to manage user accounts, doing so can disable your system. Instead, use the following commands to manage user accounts:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **useradd** | Create a user account. The following options override the settings as found in**/etc/default/useradd**:   * **-c** adds a description for the account in the GECOS field of**/etc/passwd**. * **-d** assigns an absolute pathname to a custom home directory location. * **-D** displays the default values specified in the **/etc/default/useradd**file. * **-e**specifies the date on which the user account will be disabled. * **-f** specifies the number of days after a password expires until the account is permanently disabled. * **-g**defines the primary group membership. * **-G**defines the secondary group membership. * **-M**does not createthe user's home directory. * **-m**creates the user's home directory (if it does not exist). * **-n, N**does not create a group with the same name as the user (Red Hat and Fedora respectively). * **-p**defines the encrypted password. * **-r**specifies the user account is a system user. * **-s** defines the default shell. * **-u** assigns the user a custom UID. This is useful when assigning ownership of files and directories to a different user. | **useradd pmaxwell** creates the *pmaxwell* user account**. useradd -c "Paul Morril" pmorril** creates the *pmorril* account with a comment**. useradd -d /tmpusr/sales1 sales1** creates the *sales1* user account with home directory located at*/tmpusr/sales1*. **useradd -u 789 dphilips** creates the *dphilips* account with user ID *789*. |
| **passwd** | Assign or change a password for a user.   * **passwd** (without a username or options) changes the current user's password. * Users can change their own passwords. The root user can execute all other **passwd** commands.   Be aware of the following options:   * **-S *username***displays the status of the user account.   + LK indicates the user account is locked.   + PS indicates the user account has a password. * **-l** disables (locks) an account. This command inserts a !! before the password in the **/etc/shadow**file, effectively disabling the account. * **-u** enables (unlocks) an account. * **-d** removes the password from an account. * **-n** sets the minimum number of days a password exists before it can be changed. * **-x** *s*ets the number of days before a user must change the password (password expiration time). * **-w** sets the number of days before the password expires that the user is warned. * **-i**sets the number of days following the password expiration that the account will be disabled. | **passwd jsmith** changes the password for the *jsmith* account.  **passwd -d** removes the password from an account. **passwd -d jsmith** removes the password from the *jsmith* account. **passwd -x 40 jsmith** requires *jsmith* to change his password every 40 days. **passwd -n 10 jsmith** means that *jsmith* cannot change his password for 10 days following the most recent change. **passwd -w 2 jsmith** means that *jsmith* will be warned 2 days before his password expires. **passwd -i 7 jsmith** disables the *jsmith* account after 7 days if the password is not changed. **passwd -l jsmith** locks the *jsmith* account. **passwd -u jsmith** unlocks the *jsmith* account. |
| **usermod** | Modify an existing user account. **usermod** uses several of the same switches as **useradd**. Be aware of the following switches:   * **-c**changes the description for the account. * **-l** renames a user account. When renaming the account:   + Use **-d** to rename the home directory.   + Use **-m** to copy all files from the existing home directory to the new home directory. * **-L**locks the user account. This command inserts a ! before the password in the **/etc/shadow**file, effectively disabling the account. * **-U** unlocks the user account. | **usermod -c "Paul Morril" pmorril** changes the comment field for user *pmorril*. **usermod -l esmith -d /home/esmith -m ejones** renames the *ejones* account to *esmith*, renames the home directory, and moves the old home directory contents to the new location. **usermod -s /bin/tsch esmith** points the shell for *esmith* to */bin/tsch*. **usermod -U** **esmith** unlocks the *esmith* account. |
| **userdel** | Remove the user from the system. Be aware of the following options:   * **userdel *username*** (without options) removes the user account. * **-r** removes the user's home directory. * **-f**forces the removal of the user account even when the user is logged into the system. | **userdel pmaxwell**deletes the *pmaxwell* account while leaving the home directory on the hard drive. **userdel -r pmorril**removes both the account and the home directory. |

#### File System Facts

The file system determines how a computer's files are organized on a hard drive. Linux supports many different file system types. The table below describes several common file systems.

|  |  |
| --- | --- |
| **File System Type** | **Characteristics** |
| ext2 | The second extended file system supports Access Control Lists to control individual permissions, but it does not support journaling. |
| ext3 | By layering it atop the ext2 file system, the third extended file system (ext3) supports journaling, so it has faster startup and recovery times. However, because it is tied to ext2, it doesn't offer the full performance capabilities available through a pure journaling file system. This is the most common Linux file system. |
| ext4 | Ext4 is the latest version in the ext file system family. Ext4 can handle files up to 16 terabytes and disk sizes up to 1 exabyte. |
| swap | A swap file system is used as virtual memory (the portion of the hard disk used to temporarily store portions of main memory) by the operating system. (A recommended practice is to make the swap file equal in size to the amount of memory on the computer.) |
| reiserfs | The Reiser file system (ReiserFS) is a newer Linux file system that calculates and proposes the best options for the file system. Because ReiserFS was independently built from the ground up, its journaling capability is native and more robust, offering a great deal of reliability. It is also more efficient at storing small files than other file systems. However, data corruption can occur if power goes out during disk synchronizations. Also, defragmentation tools are not available for this file system. |
| ntfs | Microsoft operating systems use NTFS (New Technology File System). Linux can only read ntfs partitions created using Windows 2000 and later, and can write to NTFS file systems created with MS operating system prior to Windows 2000. However, utilities such as ntfsprogs may enable read/write access on all ntfs partitions. |
| vfat | VFAT is a FAT32 filesystem for Linux and does not support journaling. VFAT includes long name support. Support for vfat must be compiled into the kernel for the system to recognize the vfat format. |
| xfs | The XFS file system was developed for Silicon Graphics IRIX operating system. An XFS file system is proficient at handling large files, offers smooth data transfers, and provides journaling. It also can reside on a regular disk partition or on a logical volume. |

A disk partition must be formatted using a file system. The following table describes the commands needed to format a partition.

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **mkfs** | Create an ext2, ext3, or fat file system. **Mkfs** uses the following options:   * **-t *file\_system\_type*** determines the file system. File system types include:   + **ext2**(identical to the **mkfs.ext2**command)   + **ext3**(identical to**mkfs.ext3**)   + **ext4**(identical to**mkfs.ext4**)   + **msdos**(identical to**mkfs.msdos**)   + **reiserfs** (identical to**mkreiserfs**) * **-b**specifies the block size. Supported values are 1024, 2048, or 4096. * **-i**determines how many inodes are on the partition and uses the same values as **-b**. * **-j** appends a journal to an ext2 file system.   **Note:** Without the **-b** and **-i** options, **mkfs** calculates the values automatically. | **mkfs -t ext2 /dev/sda4** creates an ext2 file system on the fourth partition on the first hard disk drive. **mkfs -t ext3 /dev/sda1** creates an ext3 file system on the first partition on the first hard disk drive. **mkfs -t ext3 /dev/sdc2** creates an ext3 file system on the second partition on the third hard disk drive. **mkfs -t ext4 /dev/sdb1** creates an ext4 file system on the first partition on the second hard disk drive. |
| **mkreiserfs** | Create a ReiserFS. | **mkreiserfs /dev/sda2**formats the second partition on the first hard disk with the Reiser file system. |
| **mkswap** | Create a swap partition. A *swap* partition is the location on the hard drive where an operating system writes memory information when it runs out of RAM. **Mkswap**:   * Requires the additional command **swapon** to activate the swap partition. * Uses the **swapoff** command to deactivate swap partitions.   **Note:** **Swapon** and **swapoff**use the**-a** option to specify all swap partitions listed in **/etc/fstab**. | **mkswap /dev/sda2** formats the second hard drive as the swap partition. **swapon /dev/sda2** activates second hard drive as the swap partition.  **swapon -a** activates all swap partitions. **swapoff /dev/sda2** deactivates second hard drive as the swap partition.  **swapoff -a** deactivates all swap partitions. |
| **mke2fs** | Create an ext2, ext3, or ext4 file system. Command options include:   * **-b**specifies the block size of the file system in Bytes (valid sizes are 1024, 2048 and 4096 bytes per block) * **-j** creates the file system with an ext3 journal * **-L** sets the volume label for the file system * **-n** displays what **mke2fs** would do if it created a file system, but does not actually create the file system * **-t**specify the file system type (i.e., ext2, ext3, ext4, etc.) that is to be created | **mke2fs /dev/sda2**creates an ext2 file system on the second partition on the first hard disk drive. **mke2fs -j /dev/sda1**creates an ext3 file system on the first partition on the first hard disk drive. **mke2fs -t ext4 /dev/sdc3**creates an ext4 file system on the third partition on the third hard disk drive. |

Keep the following in mind when working with file systems:

* Linux cannot format an extended partition; however, it can create logical partitions inside an extended partition for formatting.
* File systems use an *inode*(information node)*table* to store information about files. An inode specifies where a file's data physically exists on a disk. Inodes also contain additional information including:
  + File size
  + Modification, access, and creation times
  + Permissions
  + Ownership
* Each file system has a *superblock*, which contains information about the file system, such as:
  + File system type (e.g., ext2 and ext3)
  + Size (e.g., 10GB and 360GB)
  + Status
* Linux maintains multiple redundant copies of the superblock in every filesystem.

#### File System Maintenance Facts

Use the following commands to maintain file system integrity:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Examples** |
| **df** | Display the free space in the partition holding the specified directory. If directory is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1K blocks by default. Common options include:   * **-h** displays the output in get human readable format (bytes, KB, MB, GB). * **-i** Displays inode information. * **-l** limits the list to local file systems. | **df /home** lists the free space on the partition that holds the**/home** directory. |
| **du** | Display files and file sizes in and below a specified directory. Common options include:   * **-c** lists a total amount of space for the directory. * **-h** display the output in human readable format (bytes, KB, MB, GB). * **-s** lists only the total, not each file. | **du -c /home/badams** lists all files and directories in badams' home directory along with a file size and a total amount of space taken up by the directory. **du -c -s /home/badams** shows the total amount of space taken up in badams' home directory. |
| **lsof** | Display open files in the file system. **lsof** gives the following information by default:   * The command used to access the file * The process ID * The name of the user who is accessing the file * A file descriptor (these are described in the **lsof** man pages) * The file node type * Device numbers * The file size * The inode address * The file path   Common options include:   * **+D***directory\_name*recursivelylists files in a directory. * **-c** *command\_name* lists all files for processes that are executing the specified command. * **-u***user*lists open files owned by the specified user. * **-g** *process\_ID*lists files opened by a specific process. | **lsof -u *user***lists files opened by processes that the specified user owns. |
| **fuser** | Display the Process IDs of processes that are accessing a specified file or file system. Common options include:   * **-a**displays all process IDs. * **-v**displays extended information (verbose mode). * **-u**appends the username to each process ID. * **-m** displays process IDs in a specified directory. | **fuser -v /bin/bash** shows the user name and process ID for all users who have an open bash shell. |
| **fsck** | Check and optionally repair one or more Linux file systems. Common options include:   * **-s** serializes **fsck** when multiple file systems are checked. * **-t** specifies the type(s) of file system to be checked. * **-a** automatically repairs the file system without any questions. * **-r** prompts for confirmation when errors are found and ask permission to fix the errors (only when **-a** is not specified).   Be aware of the following:   * The file system must be unmounted before using **fsck**. * When manually running **fsck**, use runlevel 1 to ensure that other users do not mount the file system. | **fsck -t ext3 /dev/sdb1**checks the first partition on the first partition of the second hard drive. |
| **e2fsck** | Check and optionally repair a second extended file system (ext2) or ext2 files systems containing a journal (ext3). Command options include:   * **-f** forces a file system check even when the file system appears clean. * **-n** opens the file system as read-only and automatically answers all questions as "no". * **-p**automatically repairs the file system without any questions. * **-y** automatically answers all questions as "yes". * **-b**uses an alternative superblock if the primary superblock is corrupt. | **e2fsck -p /dev/sda1**checks and repairs the first partition of the first hard drive so long as it is partitioned using ext2. |
| **debugfs** | Debug the file system. The command examines and changes the state of an ext2, ext3, or ext4 file system. It allows administrators to unlink directories, change inode blocks find all inodes that point to a block, and several other similar functions. Command options include:   * **-w**specifies the file system should open in read-write mode. * **-c** specifies the file system should open in catastrophic mode (this is useful for file system with significant corruption.) * **-f**executes commands in a text file. | **debugfs /dev/sda1**opens the first partition of the first hard drive and displays a prompt that administrators can use to execute commands for the file system. |
| **dumpe2fs** | Print super block and block information for an ext2, ext3, or ext4 file system. This includes information for each sector on the partition about sector type, block ranges, inode information, free blocks, and similar information. Command options include:   * **-b** prints blocks reserved as bad in the file system. * **-h** prints only super block information. * **-x** prints group information block numbers in hexadecimal format. | **dumpe2fs /dev/sda1** lists information for the first partition of the first hard drive. |
| **tune2fs** | Adjust tunable file system parameters on ext2, ext3, and ext4 file systems. Some of the adjustable parameters include volume label, reserved blocks, inode sizes, and journaling. Tune2fs can also implement access control lists for individual users. Command options include:   * **-c** adjust the number of mounts after which the file system will be checked. * **-e remount-ro** remounts the file system as read-only. * -**l**lists the contents of the file system super block. * **-o acl**enables Posix access control lists. * **-j** converts ext2 file systems to ext3 file systems. | **tune2fs -o acl /dev/sdb1** enables access control lists on the first partition of the second hard drive. The drive needs to be remounted. |
| **xfs\_info** | Display the XFS file system parameters, such as the block size and inode data structures. This is the same functionality as the **xfs\_growfs -n** command. | **xfs\_info /dev/sdb1**displays file system parameters for the first partition of the second hard drive. |
| **xfs\_metadump** | Copy (dump) the metadata from an XFS file system to a file. It does not alter the file system. By default, the file names and extended attribute names are obfuscated before they are dumped. Command options include:   * **-e** stops dumping the file system if there is a read error. * **-g** displays the dump process. * **-o** disables obfuscation of file names and extended attributes.   **Note**: Only use **xfs\_metadump** to dump unmounted, read-only mounted, or frozen file systems. | **xfs\_metadump -o** **/dev/sdb1 /dump**copies the file system metadata for the first partition on the second hard drive to the**/dump**file. |

#### Mounting Facts

*Mounting* is the process of making a device accessible to users through the directory tree. The directory to which the device or partition is attached is called the *mount point*.

* Volumes are represented by files located in the **/dev** directory; however, volumes must be mounted before use.
* A volume is mounted to a directory. When accessing the directory in the file system, you are actually accessing the volume mounted to that directory.
* Always mount volumes and other storage devices to empty directories. Mounting a volume to a directory that contains data makes the data inaccessible.
* The **/mnt** and **/media** directories (depending on the system configuration) are directories that contain mount points specifically for external storage devices (e.g., CD-ROM drives, floppy drives, magnetic tape drives).

The following files manage and monitor the mounting of file systems:

|  |  |
| --- | --- |
| **File** | **Description** |
| **/etc/fstab** | The **/etc/fstab** file identifies volumes to mount each time the system boots. When the system boots, it automatically mounts the volumes identified in the file. Thefile contains entries with six fields that control how a device is mounted. A common entry is shown below:  /dev/hda3 /mnt/disk1 ext3 auto,ro,nosuid,users 0 1  The fields in entry are as follows:   * *Device to mount* is the path to the device file or the label that describes the volume. * *Mount point* specifies where to mount the device. This is the directory to which the device is attached. * *file system type* specifies the device's file system type. * *Options* specifies the additional options accepted when mounting the device. Options can be strung together in a comma-separated list. Be aware of the following options:   + **sync**reads all I/O files synchronously.(**async** disables this function.)   + **atime**updates the timestamp on file's inode. (**noatime**disables this function.)   + **auto** allows the volume to be mounted automatically. Use the auto parameter typically with floppy devices.   + **noauto**prevents the volume from being mounted automatically. Use this option for removable media.   + **dev** allows block files to be read from the volume. (**nodev** disables this function.)   + **exec** allows programs and script files to run from the volume. (**noexec** disables this function.)   + **owner** identifies that only the device owner can mount the volume.   + **ro** mounts the volume read only.   + **rw** mounts the volume read/write.   + **suid** allows the SUID bit to be set on files in the volume. (**nosuid** disables this function.)   + **user** identifies a specific user who can mount the volume.   + **nouser** allows only the root user to mount the volume.   + **users**allows any user to mount the volume.   + **defaults** uses the following default settings: rw, suid, dev, exec, auto, nouser, and async. * *Dump##* is used by the dump command when backing up the file system. * *Fsck#*indicates when to run **fsck** (file system check) during boot up.   + 0 = never   + 1 = only if not unmounted cleanly (Always set for the root partition.)   + 2 = always |
| **/etc/mtab** | The **/etc/mtab** file tracks the currently-mounted volumes on the system. |
| **/procs/mounts** | The **/procs/mounts** file contains entries for all currently-mounted volumes on the system. The **/proc** file system is a virtual file system that contains current system information, including the mounted file systems. |

Use the following commands to manage the mounting of file systems:

|  |  |  |
| --- | --- | --- |
| **Use...** | **To...** | **Example** |
| **mount /dev/***device/mountpoint* | Mount a volume or device. Common mount options:   * **-a** mounts all file systems listed in the **/etc/fstab**file * **-r, ro** mounts the volume as read only * **-w, rw** mounts the volume as read/write * **-t** specifies the volume type (If you mount an ext3 file system without the **-t**, the system recognizes it as an ext2 file system) * **-o loop** mounts an ISO file. | **mount -a**reads the **/etc/fstab**file and mounts all volumes listed (except those with the **noauto**parameter) **mount -rt reiserfs /dev/hdc1 /mnt/reis** mounts the hdc1 device with the reiser file system as read only to the **/mnt/reis** mount point. **mount -t iso9660 /dev/cdrom /media/cdrom**mounts the CD-ROM device to the**/media/cdrom** mount point. **mount -wt vfat /dev/fd0 /mnt/floppy** mounts the fd0 device with the vfat file system as read/write to the floppy mount point. |
| **mount** | View the currently-mounted volumes on the system. This will display the contents of the **/etc/mtab** file. |  |
| **df** | View which file systems are mounted to specific mount points. |  |
| **umount** */device* **umount***/mountpoint* | Unmount a volume or device from the system. If *disk is busy* is displayed when unmounting a device:   * Make sure the current working directory is not in that file system. * Close any open files located on that file system. | **umount /dev/hdc1** unmounts the hdc1 device. **umount /mnt/reis** unmounts the device on the /mnt/reis mount point. **umount /dev/cdrom** unmounts the CD-ROM device. **umount /mnt/cdrom** unmounts the device on the /mnt/cdrom mount point (most likely a CD-ROM device). |