**[Linux Unix Filesystems Directories Structure](http://computernetworkingnotes.com/linux-system-administrations/inittab.html)**

### Overview of the FHS

Everything in Linux can be reduced to a file. Partitions are associated with files such as /dev/hda1. Hardware components are associated with files such as /dev/modem. Detected devices are documented as files in the /proc directory. The Filesystem Hierarchy Standard (FHS) is the official way to organize files in Unix and Linux directories.

### Linux/Unix Filesystems and Directories

Several major directories are associated with all modern Unix/Linux operating systems. These directories organize user files, drivers, kernels, logs, programs, utilities, and more into different categories. The standardization of the FHS makes it easier for users of other Unix-based operating systems to understand the basics of Linux. Every FHS starts with the root directory, also known by its label, the single forward slash (/). All of the other directories shown in Table are subdirectories of the root directory. Unless they are mounted separately, you can also find their files on the same partition as the root directory.

|  |  |
| --- | --- |
| / | The root directory, the top-level directory in the FHS. All other directories are subdirectories of root, which is always mounted on some partition. All directories that are not mounted on a separate partition are included in the root directory?s partition. |
| /bin | Essential command line utilities. Should not be mounted separately; otherwise, it could be difficult to get to these utilities when using a rescue disk. |
| /boot | Includes Linux startup files, including the Linux kernel. Can be small; 16MB is usually adequate for a typical modular kernel. If you use multiple kernels, such as for testing a kernel upgrade, increase the size of this partition accordingly. |
| /etc | Most basic configuration files. |
| /dev | Hardware and software device drivers for everything from floppy drives to terminals. Do not mount this directory on a separate partition. |
| /home | Home directories for almost every user. |
| /lib | Program libraries for the kernel and various command line utilities. Do not mount this directory on a separate partition. |
| /mnt | The mount point for removable media, including floppy drives, CD-ROMs, and Zip disks. |
| /opt | Applications such as WordPerfect or StarOffice. |
| /proc | Currently running kernel-related processes, including device assignments such as IRQ ports, I/O addresses, and DMA channels. |
| /root | The home directory of the root user. |
| /sbin | System administration commands. Don't mount this directory separately. |
| /tmp | Temporary files. By default, Red Hat Linux deletes all files in this directory periodically. |
| /usr | Small programs accessible to all users. Includes many system administration commands and utilities. |
| /var | Variable data, including log files and printer spools. |

# Types of Files Used by Linux

When working with Linux, you need to be aware of the fact that there are a number of different file types used by the file system. This is another area where the Linux file system differs significantly from the Windows file system. With a Windows file system you basically have two entry types in the file system:

* Directories
* Files

Granted, you can have normal files, hidden files, shortcut files, word processing files, executable files, and so on. However, these are all simple variations of the basic file when working with Windows.  
With Linux, however, there are a variety of different file types used by the file system. These include the file types shown in Table

|  |  |
| --- | --- |
| **File Type** | **Description** |
| Regular files | These files are similar to those used by the file systems of other operating systems—for example, executable files, OpenOffice.org files, images, text configuration files, etc. |
| Links | These files are pointers that point to other files in the file system. |
| FIFOs | FIFO stands for First In First Out. These are special files used to move data from one running process on the system to another. A FIFO file is basically a queue where the first chunk of data added to the queue is the first chunk of data removed from the queue. Data can only move in one direction through a FIFO. |
| Sockets | Sockets are similar to FIFOs in that they are used to transfer information between sockets. With a socket, however, data can move bi-directionally. |

# Some of the Configuration Files in /etc Directory that you should remember

|  |  |
| --- | --- |
| **File** | **Function** |
| /etc/fstab | Lists the partitions and file systems that will be automatically mounted when the system boots. |
| /etc/group | Contains local group definitions. |
| /etc/grub.conf | Contains configuration parameters for the GRUB bootloader (assuming it's being used on the system). |
| /etc/hosts | Contains a list of hostname-to-IP address mappings the system can use to resolve hostnames. |
| /etc/inittab | Contains configuration parameters for the init process. |
| /etc/init.d/ | A subdirectory that contains startup scripts for services installed on the system. On a Fedora or Red Hat system, these are located in /etc/rc.d/init.d. |
| /etc/modules.conf | Contains configuration parameters for your kernel modules. |
| /etc/passwd | Contains your system user accounts. |
| /etc/shadow | Contains encrypted passwords for your user accounts. |
| /etc/X11/ | Contains configuration files for X Windows. |

#### Share this post

## [Login linux terminal pwd ls useradd commands accessing virtual terminals](http://computernetworkingnotes.com/linux-file-system-basic-commands/login-linux-terminal.html)

In this assignment I will instruct you about some basic commands of Linux. You will get seven virtual terminal when you perform full installations. Although you can use graphics for daily task but here we are preparing for RHCE exam so you must use command line interface. Because all question are based on command line in RHCE Exam.

### Virtual Consoles

A virtual console is a command line where you can log into and control Linux. As RHEL is a multi terminal operating system, you can log into Linux, even with the same user ID, several times. It's easy to open a new virtual console. Just use the appropriate ALT-function key combination. For example, pressing ALT-F2 brings you to the second virtual console. You can switch between adjacent virtual consoles by pressing ALT-RIGHT ARROW or ALT-LEFT ARROW. For example, to move from virtual console 2 to virtual console 3, press ALT-RIGHT ARROW.

You can switch between virtual terminals by just press the **ALT+CTRL+Funcation key** combinations.

ALT + CTRL + F1 for terminal 1

ALT + CTRL + F2 for terminal 2

ALT + CTRL + F3 for terminal 3

ALT + CTRL + F4 for terminal 4

ALT + CTRL + F5 for terminal 5

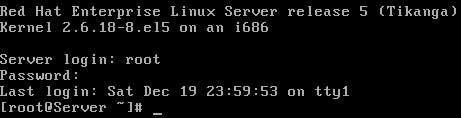
ALT + CTRL + F6 for terminal 6

ALT + CTRL + F7 for terminal 7

Terminal 7 is by default graphic mode beside it all six terminal are CLI based. Open any terminial by press ALT+CTRL+F1 key combinations. root account is automatically created when we install Linux.



Type **root** on login name and press enter key, now give password ( no asterisk character like window to guess the password length) When you login from root account you will get **#** sign at command prompt , and when you login from normal user you will get **$** prompt.



### #clear

This command is used to clear the screen.You have three options to logout .

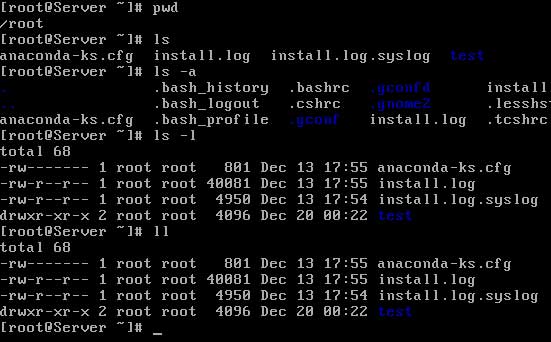
### Press CTRL+D

### #exit

### #logout

All three commands perform same task.

### #pwd /root



Print working directory command will tell you about current location from / partition.

### #ls

ls command will list the object in directory. All directory are listed in blue color while files are shown white color.

### #ls –a

Normal ls command will not list the hidden files. If you want to list the hidden file use –a switch with ls command to list the hidden files.

### #ls –l

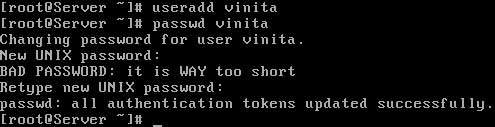
Ls command with –l switch will list the objects in long formats . we will discuss more about –l switch in coming sections.

### #ll

Same as ls –l . First and major task for any system administrator is user managements. For testing purpose you can perform all task with root account but in real life root account is used for administrative purpose only. Lest create a normal user account for further practical.

### #useradd [user name ]

Useradd command is used to create user. Several advance options are used with useradd command but you will learn about them in coming article.



### #passwd [user name]

In linux no user can be login without password. passwd command is used to assign password for any user. Do not execute this command without user name otherwise it will change root password.

## [Basic Linux commands cp mv rm mkdir cat cd command example](http://computernetworkingnotes.com/linux-file-system-basic-commands/basic-linux-commands.html)

In this assignment I will demonstrate some basic commands which are required to perform day to day task by user. In our last assignment we created a normal user named Vinita. Now login from Vinita, and try to find out what are the difference you noticed when you login from normal user.



In bracket right most side is showing **user name Vinita** and beside **@Server is the hostname of computer** and further **~** sign is showing that user is presently logged in her home directory. But first, every Linux user has a home directory. You can use the tilde (~) to represent the home directory of any currently active user. For example, if your username is Vinita, your home directory is /home/Vinita. If you have logged in as the root user, your home directory is /root. Thus, the effect of the **cd ~** command depends on your username. For example, if you have logged in as user Vickey, the **cd ~** command brings you to the /home/Vickey directory. If you have logged in as the root user, this command brings you to the /root directory. You can list the contents of your home directory from anywhere in the directory tree with the **ls ~** command. After bracket you can see the command prompt of normal user is $ sign.

### $mkdir [ directory name ]

mkdir command is used to create new directory. Let’s create a example directory.

$mkdir example $ls example

now create a file. Syntax for creating file is

$cat > [file name]

This command can be used in three way to see the matter of file, to create a new file or to append the matter of file.

$cat [file name] ------------------------ To see the matter of file

$cat > [file name]---------------------- To create a file

$cat >> [file name ]-------------------- To append the matter of file

Be little bit careful while using cat > command to create new files. If you accidently used this command with existing file it will overwrite the matter of file. Use CTRL+D to save the matter of file.  
Different use of cat command

$cat > test This is test of file

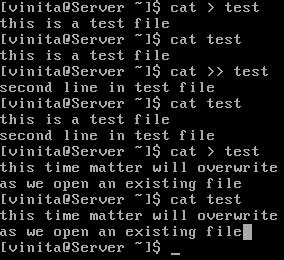
$cat test This is test of file

$cat >> test This is second line

$cat example This is test of file This is second line in test file

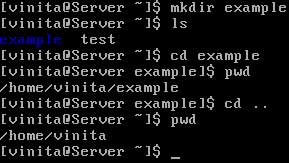
$cat > test Now file will over write

$cat test Now file will overwrite



### $cd [ destination directory path]

It is easy to change directories in Linux. Just use cd and name the absolute path of the desired directory. If you use the relative path, just remember that your final destination depends on the present working directory.



as you can see in the output of ls command **file** is in white color and **directory** is in blue color.

There are two path concepts associated with Linux directories: absolute paths and relative paths.  
**An absolute path** describes the complete directory structure based on the top level directory, root (/).  
**A relative path** is based on the current directory. Relative paths do not include the slash in front.  
The difference between an absolute path and a relative one is important To know more about path and directory structure

### pwd

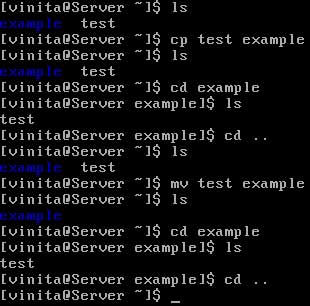
In many configurations, you may not know where you are relative to the root (/) directory. The pwd command, which is short for print working directory, can tell you, relative to root (/). Once you know where you are, you can determine whether you need to move to a different directory.

### $cd ..

this command is used to exit from current directory.

### cp

The cp (copy) command allows you to take the contents of one file and place a copy with the same or different name in the directory of your choice. For example, the **cp file1 file2** command takes the contents of file1 and saves the contents in file2. One of the dangers of cp is that it can easily overwrite files in different directories, without prompting you to make sure that's what you really wanted to do.



### mv

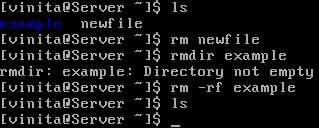
While you can't rename a file in Linux, you can move it. The mv command essentially puts a different label on a file. For example, the **mv file1 file2** command changes the name of file1 to file2. Unless you're moving the file to a different partition, everything about the file, including the inode number, remains the same.

### rm

rm command is used to remove the regular files. It will ask before removing files.

### rmdir

it will remove the empty directory. If directory is full then use **rm –rf [directory name]**



## [RHCE Linux System Administrations commands descriptions and examples](http://computernetworkingnotes.com/linux-file-system-basic-commands/basic-linux-commands-part1.html)

### halt

This command shuts down the operating system, but can only be run by the root user.

#halt

### reboot

This command shuts down and restarts the operating system. It also can only be run by root.

#reboot [will perform simple reboot]

#reboot -f [will perform fast reboot ]

### init 0

This command also shuts down the operating system, and can only be run by your root user.

#init 0

### init 6

This command also shuts down and restarts the operating system. It also can only be run by root

#init 6

### man

This command opens the manual page for the command or utility specified. The man utility is a very useful tool. If you are unsure how to use any command, use man to access its manual page. For example, you could enter **man ls** at the shell prompt to learn how to use the ls utility.

#man ls

### info

The info utility also displays a help page for the indicated command or utility. The information displayed with info command will be in-depth than that displayed in the man page for the same command.

info ls

### su

This command switches the current user to a new user account. For example, if you’re logged in as **vickey** and need to change to user account to **vinita,** you can enter su **vinita** at the shell prompt. This command is most frequently used to switch to the superuser root account.   
In fact, if you don’t supply a username, this utility assumes that you want to change to the root account. If you enter **su -**, then you will switch to the root user account and have all of root’s environment variables applied.  
This command require password of the user you want switch.

Looking for Files

There are two basic commands used for file searches: **find** and **locate**

### find

The find command searches through directories and subdirectories for a desired file. For example, if you wanted to find the directory with the **grub.conf** linux boot loader file, you could use the following command, which would start the search in the top-level root (/) directory:

# find / -name grub.conf

But this search took several minutes to get it task done. Alternatively, if you know that this file is located in the /etc subdirectory tree, or /boot/grub/grub.conf you could start in that directory with the following command:

# find /etc -name grub.conf

### locate

If this is all too time-consuming, RHEL 5 includes a default database of all files and directories. Searches with the locate command are almost instantaneous. And locate searches don't require the full file name. The drawback is that the locate command database is normally updated only once each day, as documented in the /etc/cron.daily/mlocate.cron script.

### Getting into the Files

Now that you see how to find and get around different files, it's time to start reading, copying, and moving the files around. Most Linux configuration files are text files. Linux editors are text editors. Linux commands are designed to read text files. If in doubt, you can check the file types in the current directory with the   
**file \* command.**

### cat

The most basic command for reading files is cat. The **cat filename command** scrolls the text within the filename file. It also works with multiple file names; it concatenates the file names that you might list as one continuous output to your screen. You can redirect the output to the file name of your choice.

### more and less

Larger files demand a command that can help you scroll through the file text at your leisure. Linux has two of these commands:   
more and less.  
With the more filename command, you can scroll through the text of a file, from start to finish, one screen at a time. With the less filename command, you can scroll in both directions through the same text with the PAGE UP and PAGE DOWN keys. Both commands support vi-style searches.

### head and tail

The head and tail commands are separate commands that work in essentially the same way. By default, the head filename command looks at the first 10 lines of a file; the tail filename command looks at the **last 10 lines** of a file. You can specify the number of lines shown with the **-nx** switch. Just remember to avoid the space when specifying the number of lines; for example, the

# tail -n15 /etc/passwd

command lists the last 15 lines of the **/etc/passwd** file.

### cp

The cp (copy) command allows you to take the contents of one file and place a copy with the same or different name in the directory of your choice. For example, the cp file1 file2 command takes the contents of file1 and saves the contents in file2. One of the dangers of cp is that it can easily overwrite files in different directories, without prompting you to make sure that's what you really wanted to do.

### mv

While you can't rename a file in Linux, you can move it. The mv command essentially puts a different label on a file. For example, the mv file1 file2 command changes the name of file1 to file2. Unless you're moving the file to a different partition, everything about the file, including the inode number, remains the same.

### ln

You can create a linked file.   
linked files are common with device files such as /dev/dvdwriter and /dev/par0. They're also useful for making sure that multiple users have a copy of the same file in their directories. Hard links include a copy of the file. As long as the hard link is made within the same partition, the inode numbers are identical. You could delete a hard-linked file in one directory, and it would still exist in the other directory. For example, the following command creates a hard link from the actual Samba configuration file to **smb.conf** in the local directory:

# ln smb.conf /etc/samba/smb.conf

On the other hand, a soft link serves as a redirect; when you open up a file created with a soft link, you're directed to the original file. If you delete the original file, the file is lost. While the soft link is still there, it has nowhere to go. The following command is an example of how you can create a soft link:

# ln -s smb.conf /etc/samba/smb.conf

### sort

You can sort the contents of a file in a number of ways. By default, the sort command sorts the contents in alphabetical order depending on the first letter in each line. For example, the sort /etc/passwd command would sort all users (including those associated with specific services and such) by username.

### grep and egrep

The **grep** command uses a search term to look through a file. It returns the full line that contains the search term. For example, grep 'vickey' /etc/passwd looks for my name in the /etc/passwd file.

The **egrep** command is more forgiving; it allows you to use some unusual characters in your search, including +, ?, |, (, and). While it's possible to set up grep to search for these characters with the help of the backslash, the command can be awkward to use.

### wc

The wc command, short for word count, can return the number of lines, words, and characters in a file. The wc options are straightforward: for example, wc -w filename returns the number of words in that file.

### sed

The sed command, short for stream editor, allows you to search for and change specified words or even text streams in a file. For example, the following command changes the first instance of the word Windows to the word Linux in each line of the file data, and writes the result to the file newdata:

# sed 's/Windows/Linux/' data > newdata

However, this may not be enough. If a line contains more than one instance of Windows, the above sed command does not change the second instance of that word. But you can make it change every appearance of Windows by adding a **"global"** suffix:

# sed 's/Windows/Linux/g' data > newdata

### awk

The awk command, named for its developers (Aho, Weinberger, and Kernighan), is more of a database manipulation utility. It can identify lines with a keyword and read out the text from a specified column in that line. Again, using the /etc/passwd file, for example, the following command will read out the username of every user with a vickey in the comment column:

# awk '/vickey/ {print $1}' /etc/passwd

### ps

It's important to know what's running on your Linux computer. The ps command has a number of critical switches. When trying to diagnose a problem, it's common to get the fullest possible list of running processes, and then look for a specific program. For example, if the Firefox Web browser were to suddenly crash, you'd want to kill any associated processes. The ps aux | grep firefox command could then help you identify the process(es) that you need to kill.

### who and w

If you want to know what users are currently logged into your system, use the who command or the w command. This can help you identify the usernames of those who are logged in, their terminal connections, their times of login, and the processes that they are running.

### Wildcards

Sometimes you may not know the exact name of the file or the exact search term. This is when a wildcard is handy. The basic wildcards are shown

|  |  |
| --- | --- |
| Wildcard | Description |
| \* | Any number of alphanumeric characters (or no characters at all). For example, the ls ab\* command would return the following file names, assuming they exist in the current directory: ab, abc, abcd. |
| ? | One single alphanumeric character. For example, the ls ab? command would return the following file names, assuming they exist in the current directory: abc, abd, abe |
| [ ] | A range of options. For example, the ls ab[123] command would return the following file names, assuming they exist in the current directory: ab1, ab2, ab3. Alternatively, the ls ab[X-Z] command would return the following file names, assuming they exist in the current directory: abX, abY, abZ. |

### env

This command displays the environment variables for the currently logged-in user.

### echo

This command is used to echo a line of text on the screen. It’s frequently used to display environment variables. For example, if you wanted to see the current value of the PATH variable, you could enter   
**echo $PATH**

### top

This command is a very useful command that displays a list of all applications and processes currently running on the system. You can sort them by CPU usage, memory usage, process ID number, and which user owns them

### which

This command is used to display the full path to a shell command or utility. For example, if you wanted to know the full path to the **ls command,** you would enter  
**which ls**

### whoami

This command displays the username of the currently logged-in user.

### netstat

This command displays the status of the network, including current connections, routing tables, etc

### route

This command is used to view or manipulate the system’s routing table.

### ifconfig

This command is used to manage network boards installed in the system. It can be used to display or modify your network board configuration parameters. This command can only be run by the root user.

Once you become familiar with these basic command which you need to perform RCHE skill you are ready to move our next series of article focused on RHCE exam.