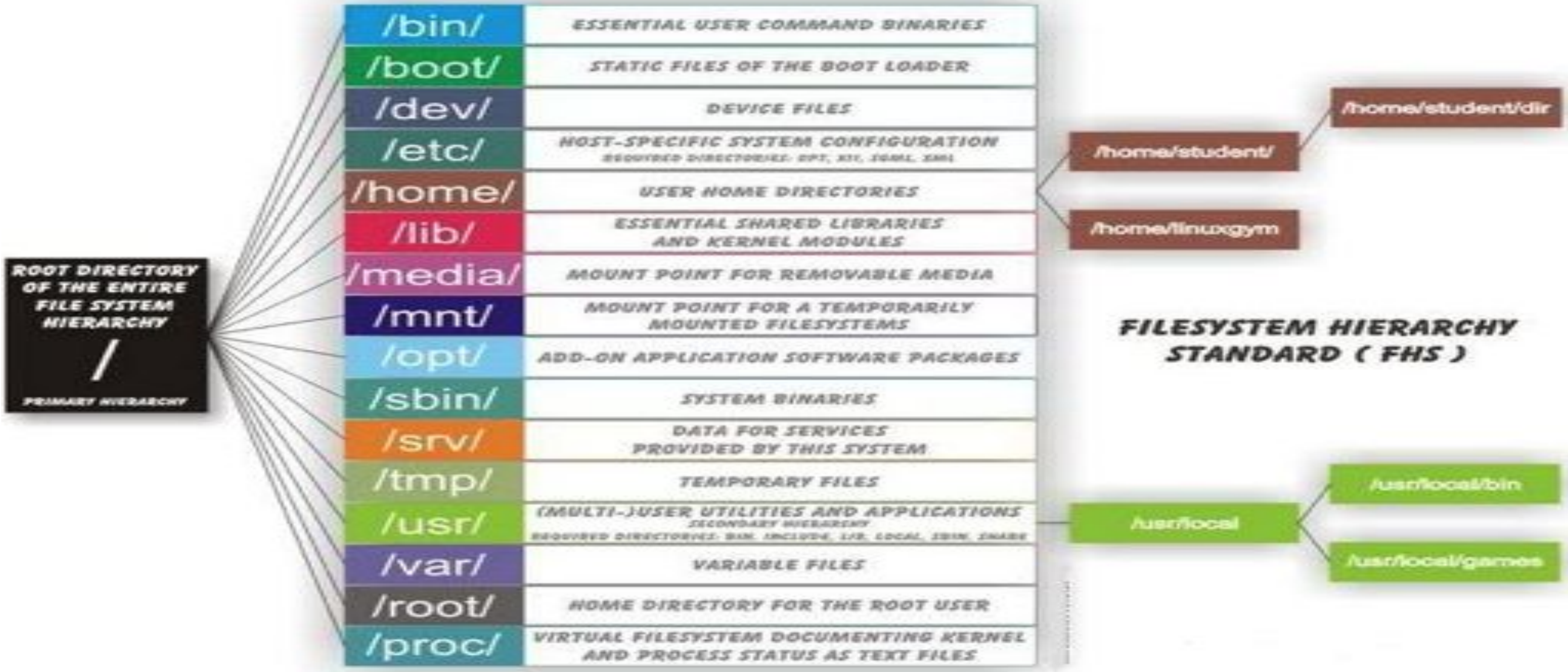
Introduction To Linux

Basic features of Linux

- **Open Source** - Linux source code is freely available and it is community based development project. Multiple teams works in collaboration to enhance the capability of Linux operating system and it is continuously evolving
- **Multi-User** - Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** - Linux is a multiprogramming system means multiple applications can run at same time.



ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY

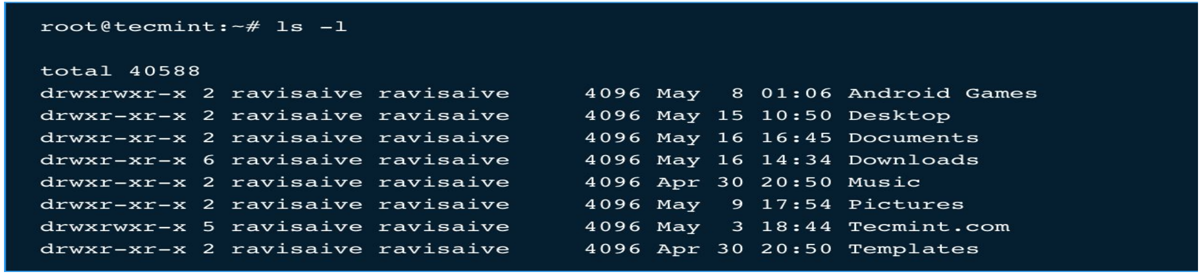
/

PRIMARY HIERARCHY

/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

FILESYSTEM HIERARCHY STANDARD (FHS)

Basic Commands



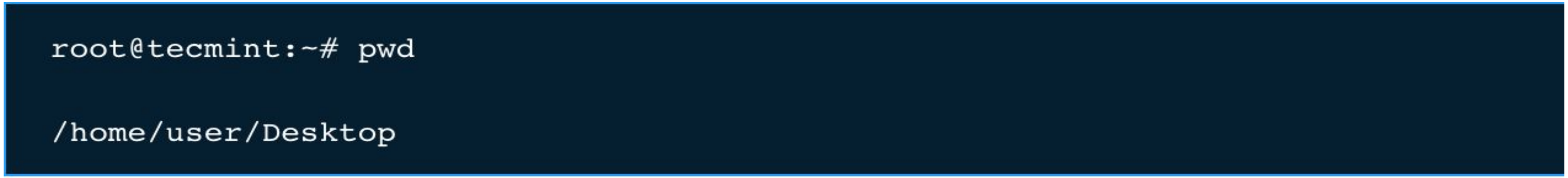
```
root@tecmint:~# ls -l

total 40588
drwxrwxr-x 2 ravisaive ravisaive     4096 May  8 01:06 Android Games
drwxr-xr-x 2 ravisaive ravisaive     4096 May 15 10:50 Desktop
drwxr-xr-x 2 ravisaive ravisaive     4096 May 16 16:45 Documents
drwxr-xr-x 6 ravisaive ravisaive     4096 May 16 14:34 Downloads
drwxr-xr-x 2 ravisaive ravisaive     4096 Apr 30 20:50 Music
drwxr-xr-x 2 ravisaive ravisaive     4096 May  9 17:54 Pictures
drwxrwxr-x 5 ravisaive ravisaive     4096 May  3 18:44 Tecmint.com
drwxr-xr-x 2 ravisaive ravisaive     4096 Apr 30 20:50 Templates
```

- d (stands for directory).
- rwxr-xr-x is the file permission of the file/folder for owner, group and world.
- May 8 01:06 is the date and ti me of last modification.
- And at the end is the name of the File/Folder.

Continued..

- **pwd :** The command “pwd” (print working directory), prints the current working directory with full path name from terminal.
- **cd** : the frequently used “cd” command stands for (change directory), it change the current working directory.
- **cd ~** : “cd ~” will change the working directory to user’s home directory, and is very useful if a user finds himself lost in terminal.
-
- **cd ..** : “cd ..” will change the working directory to parent directory (of current working directory).



```
root@tecmint:~# pwd

/home/user/Desktop
```

Continued..

- **touch:** The “touch” command creates the file, only if it doesn’t exist. If the file already exists it will update the timestamp and not the contents of the file.
  $ touch abc.txt

- **cat**: The “cat” stands for (Concatenation). Concatenate (join) two or more plain file and/or print contents of a file on standard output.
  $ cat abc.txt

- **cp**: The “copy” stands for (Copy), it copies a file from one location to another location.
  $ cp /home/user/Downloads abc.tar.gz /home/user/Desktop

- **mv**: The “mv” command move/rename a file from one location to another location.
  $ mv /home/user/Downloads abc.tar.gz /home/user/Desktop

- **mkdir**: The “mkdir” (Make directory) command create a new directory with name path. However is the directory already exists, it will return an error message “cannot create folder, folder already exists”.
  $ mkdir -p /a/b/c

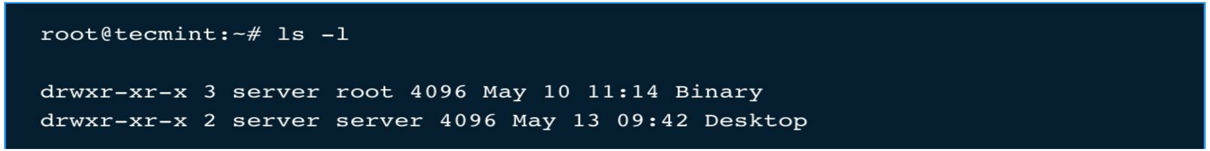
- **rm:** The “rm” command removes each specified FILE. By default, it does not remove directories.
  $ rm -i filename : Prompt before every removal.
  $ rm -f: forcefully remove files and never ask
  $ rm -r: Remove directories and their contents recursively.

Continued..

- **chown:** The Linux “chown” command stands for (change file owner and group). Every file belongs to a group of user and a owner.

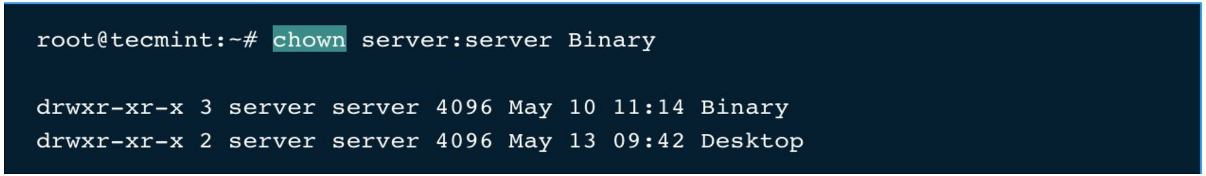
  This “chown” command is used to change the file ownership and thus is useful in managing and providing file to authorised user and usergroup only.

  lets Do ‘ls -l‘ into your directory and you will see something like this.



```
root@tecmint:~# ls -l

drwxr-xr-x 3 server root 4096 May 10 11:14 Binary
drwxr-xr-x 2 server server 4096 May 13 09:42 Desktop
```

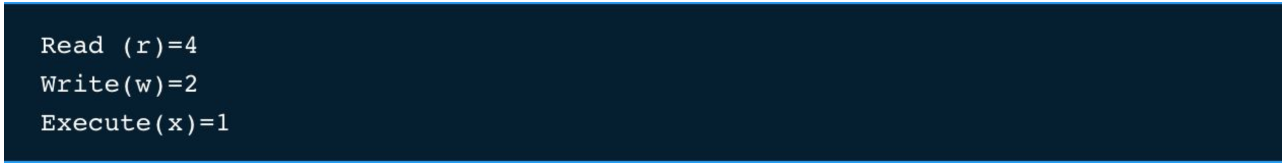


```
root@tecmint:~# chown server:server Binary

drwxr-xr-x 3 server server 4096 May 10 11:14 Binary
drwxr-xr-x 2 server server 4096 May 13 09:42 Desktop
```

Continued..

- **chmod**: The Linux “chmod” command stands for (change file mode bits). chmod changes the file mode (permission) of each given file, folder, script, etc.. according to mode asked for.
- There exist 3 types of permission on a file (folder or anything but to keep things simple we will be using file).



```
Read (r)=4
Write(w)=2
Execute(x)=1
```

lets try some combinations

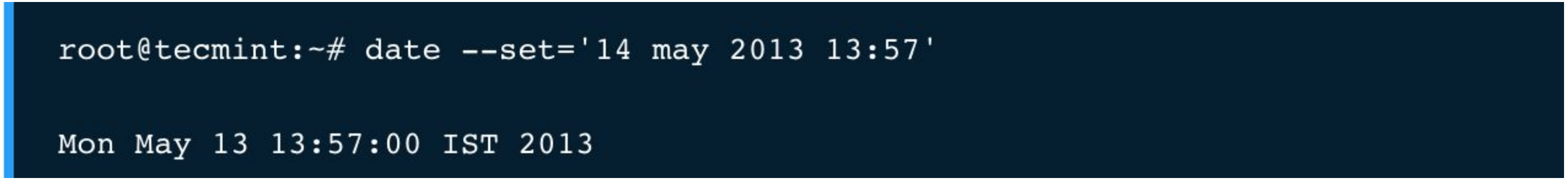
```
$ chmod 666 abc.txt
 -rw-rw-rw-+ 15 staff  staff   510 Jan 11 22:08
$ chmod 777 abc.txt
-rwxrwxrwx+ 15 staff  staff   510 Jan 11 22:08
$ chmod 555 abc.txt
-r-xr-xr-x+ 15 staff  staff   510 Jan 11 22:08
$ chmod 333 abc.txt
-wx-wx-wx-+ 15 staff  staff   510 Jan 11 22:08
```

Continued..

- **cal**: The “cal” (Calendar), it is used to displays calendar of the present month or any other month of any year that is advancing or passed.

- **echo**: The echo command is used to print content on the screen
  $echo “hello world”

- **date**: The “date” (Date) command print the current date and time on the standard output, and can further be set.



```
root@tecmint:~# date --set='14 may 2013 13:57'

Mon May 13 13:57:00 IST 2013
```

Continued..

- **cal**: The “cal” (Calendar), it is used to displays calendar of the present month or any other month of any year that is advancing or passed.

- **echo**: The echo command is used to print content on the screen
  $echo “hello world”

- **date**: The “date” (Date) command print the current date and time on the standard output, and can further be set.

Standard Redirections

- By default, standard output directs its contents to the display. To redirect standard output to a file, the ">" character is used like this:

  ```
  $ ls > abc.txt
  ```

- Each time the command above is repeated, file_list.txt is overwritten (from the beginning) with the output of the command **ls**. If you want the new results to be *appended* to the file instead, use ">>" like this:

  ```
  $ ls >> abc.txt
  ```

- The most useful and powerful thing you can do with I/O redirection is to connect multiple commands together with what are called *pipes*. With pipes, the standard output of one command is fed into the standard input of another.

  ```
  $ ls -l | less
  ```

Basic Commands

- **head** - The *head* command reads the first 10 lines of any file given to it. If it is desired to obtain some number of lines other than the default ten, the -*n* option can be used followed by an integer indicating the number of lines desired

  ```
  $ head -n 15 abc.txt
  ```

- **tail** - The “*tail*” command reads the last 10 lines of any file given to it. If it is desired to obtain some number of lines other than the default ten, the -*n* option can be used followed by an integer indicating the number of lines desired

  ```
  $ tail -n 15 abc.txt
  ```

Continued..

- **diff**:- The “diff” analyzes two files and prints the lines that are different. Essentially, it outputs a set of instructions for how to change one file in order to make it identical to the second file.

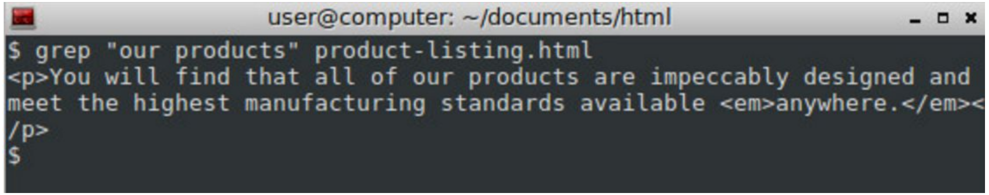
  ```
  $ diff abc.txt xyz.txt
  ```

- **wc**: The wc (word count) command in Unix/Linux operating systems is used to find out number of newline count, word count, byte and characters count in a files specified by the file arguments.

  ```
  $ wc -l : Prints the number of lines in a file.
  $ wc -w : prints the number of words in a file.
  $ wc -c : Displays the count of bytes in a file.
  $ wc -m : prints the count of characters from a file.
  $ wc -L : prints only the length of the longest line in a file.
  ```

Continued..

grep: It stands for "global regular expression print," processes text line by line and prints any lines which match a specified pattern.



```
$ grep "our products" product-listing.html
<p>You will find that all of our products are impeccably designed and
meet the highest manufacturing standards available <em>anywhere.</em><
/p>
$
```

$ grep -i "boo" /etc/passwd #You can force grep to **ignore** word case
$ grep -r "192.168.1.5" /etc/ #You can search **recursively**
$ grep -v bar /path/to/file #You can use -v option to print **inverts** the match
$ grep --color vivek /etc/passwd #You can force grep to display output in **colors**

VIM

Installing Software via command line

- Install dpkg package (local) - dpkg -i openssh-server
- sudo apt-get install <*packagename*>
- sudo apt-get install zip

process management

- **top**: The “top” command is one of the most frequently used commands in our daily system administrative jobs.top command displays processor activity of your Linux box and also displays tasks managed by kernel in real-time. It’ll show processor and memory are being used and other information like running processes. This may help you to take correct action

- **ps:** The “ps” command on linux is one of the most basic commands for viewing the processes running on the system. It provides a snapshot of the current processes along with detailed information like user id, cpu usage, memory usage, command name etc.