**1. Command: ls**

The command “**ls**” stands for (**List Directory Contents**), List the contents of the folder, be it file or folder, from which it runs.

root@tecmint:~# ls

Android-Games Music

Pictures Public

Desktop Tecmint.com

Documents TecMint-Sync

Downloads Templates

The command “**ls -l**” list the content of folder, in **long listing** fashion.

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

Command “**ls -a**“, list the content of folder, including **hidden** files starting with ‘.’.

root@tecmint:~# ls -a

. .gnupg .dbus .goutputstream-PI5VVW .mission-control

.adobe deja-dup .grsync .mozilla .themes

.gstreamer-0.10 .mtpaint .thumbnails .gtk-bookmarks .thunderbird

.HotShots .mysql\_history .htaccess .apport-ignore.xml .ICEauthority

.profile .bash\_history .icons .bash\_logout .fbmessenger

.jedit .pulse .bashrc .liferea\_1.8 .pulse-cookie

.Xauthority .gconf .local .Xauthority.HGHVWW .cache

.gftp .macromedia .remmina .cinnamon .gimp-2.8

.ssh .xsession-errors .compiz .gnome teamviewer\_linux.deb

.xsession-errors.old .config .gnome2 .zoncolor

Note: In **Linux** file name starting with ‘**.**‘ is hidden. In Linux every file/folder/device/command is a file. The output of **ls -l** is:

1. **d** (stands for directory).
2. **rwxr-xr-x** is the file permission of the file/folder for owner, group and world.
3. The 1st **ravisaive** in the above example means that file is owned by user ravisaive.
4. The 2nd **ravisaive** in the above example means file belongs to user group ravisaive.
5. **4096** means file size is 4096 Bytes.
6. **May 8 01:06** is the date and time of last modification.
7. And at the end is the name of the **File/Folder**.

For more “**ls**” command examples read [15 ‘ls’ Command Examples in Linux](http://www.tecmint.com/15-basic-ls-command-examples-in-linux/).

**2. Command: lsblk**

The “**lsblk**” stands for (**List Block Devices**), print block devices by their assigned name (but not **RAM**) on the standard output in a tree-like fashion.

root@tecmint:~# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 232.9G 0 disk

├─sda1 8:1 0 46.6G 0 part /

├─sda2 8:2 0 1K 0 part

├─sda5 8:5 0 190M 0 part /boot

├─sda6 8:6 0 3.7G 0 part [SWAP]

├─sda7 8:7 0 93.1G 0 part /data

└─sda8 8:8 0 89.2G 0 part /personal

sr0 11:0 1 1024M 0 rom

The “**lsblk -l**” command list block devices in ‘**list**‘ structure (not tree like fashion).

root@tecmint:~# lsblk -l

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 232.9G 0 disk

sda1 8:1 0 46.6G 0 part /

sda2 8:2 0 1K 0 part

sda5 8:5 0 190M 0 part /boot

sda6 8:6 0 3.7G 0 part [SWAP]

sda7 8:7 0 93.1G 0 part /data

sda8 8:8 0 89.2G 0 part /personal

sr0 11:0 1 1024M 0 rom

Note: **lsblk** is very useful and easiest way to know the name of **New Usb Device** you just plugged in, especially when you have to deal with disk/blocks in terminal.

**3. Command: md5sum**

The “**md5sum**” stands for (**Compute** and **Check MD5 Message Digest**), md5 checksum (commonly called **hash**) is used to match or verify integrity of files that may have changed as a result of a faulty file transfer, a disk error or non-malicious interference.

root@tecmint:~# md5sum teamviewer\_linux.deb

47790ed345a7b7970fc1f2ac50c97002 teamviewer\_linux.deb

Note: The user can match the generated md5sum with the one provided officially. Md5sum is considered less secure than sha1sum, which we will discuss later.

**4. Command: dd**

Command “**dd**” stands for (**Convert** and **Copy a file**), Can be used to convert and copy a file and most of the times is used to copy a iso file (or any other file) to a usb device (or any other location), thus can be used to make a ‘**Bootlable**‘ Usb Stick.

root@tecmint:~# dd if=/home/user/Downloads/debian.iso of=/dev/sdb1 bs=512M; sync

Note: In the above example the usb device is supposed to be sdb1 (You should Verify it using command **lsblk**, otherwise you will overwrite your disk and OS), use name of disk very Cautiously!!!.

**dd** command takes some time ranging from a few seconds to several minutes in execution, depending on the size and type of file and read and write speed of Usb stick.

**5. Command: uname**

The “**uname**” command stands for (**Unix Name**), print detailed information about the machine name, Operating System and Kernel.

root@tecmint:~# uname -a

Linux tecmint 3.8.0-19-generic #30-Ubuntu SMP Wed May 1 16:36:13 UTC 2013 i686 i686 i686 GNU/Linux

Note: uname shows type of kernel. **uname -a** output detailed information. Elaborating the above output of **uname -a**.

1. “**Linux**“: The machine’s kernel name.
2. “**tecmint**“: The machine’s node name.
3. “**3.8.0-19-generic**“: The kernel release.
4. “**#30-Ubuntu SMP**“: The kernel version.
5. “**i686**“: The architecture of the processor.
6. “**GNU/Linux**“: The operating system name.

**6. Command: history**

The “**history**” command stands for **History (Event) Record**, it prints the history of long list of executed commands in terminal.

root@tecmint:~# history

1 sudo add-apt-repository ppa:tualatrix/ppa

2 sudo apt-get update

3 sudo apt-get install ubuntu-tweak

4 sudo add-apt-repository ppa:diesch/testing

5 sudo apt-get update

6 sudo apt-get install indicator-privacy

7 sudo add-apt-repository ppa:atareao/atareao

8 sudo apt-get update

9 sudo apt-get install my-weather-indicator

10 pwd

11 cd && sudo cp -r unity/6 /usr/share/unity/

12 cd /usr/share/unity/icons/

13 cd /usr/share/unity

Note: Pressing “**Ctrl + R**” and then search for already executed commands which lets your command to be completed with auto completion feature.

(reverse-i-search)`if': ifconfig

**7. Command: sudo**

The “**sudo**” (**super user do**) command allows a permitted user to execute a command as the superuser or another user, as specified by the security policy in the sudoers list.

root@tecmint:~# sudo add-apt-repository ppa:tualatrix/ppa

Note: **sudo** allows user to borrow superuser privileged, while a similar command ‘**su**‘ allows user to actually log in as superuser. **Sudo** is safer than **su**.  
It is not advised to use **sudo** or **su** for day-to-day normal use, as it can result in serious error if accidentally you did something wrong, that’s why a very popular saying in Linux community is:

“To err is human, but to really foul up everything, you need root password.”

**8. Command: 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”**.

root@tecmint:~# mkdir tecmint

Note: Directory can only be created inside the folder, in which the user has write permission. **mkdir**: cannot create directory `**tecmint**‘: File exists  
(Don’t confuse with file in the above output, you might remember what i said at the beginning – In Linux every file, folder, drive, command, scripts are treated as file).

**9. Command: touch**

The “**touch**” command stands for (Update the access and modification times of each **FILE** to the current time). **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.

root@tecmint:~# touch tecmintfile

Note: **touch** can be used to create file under directory, on which the user has write permission, only if the file don’t exist there.

**10. Command: 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

So if you want to give only read permission on a file it will be assigned a value of ‘**4**‘, for write permission only, a value of ‘**2**‘ and for execute permission only, a value of ‘**1**‘ is to be given. For read and write permission **4+2** = ‘**6**‘ is to be given, ans so on.

Now permission need to be set for **3** kinds of user and usergroup. The first is owner, then usergroup and finally world.

rwxr-x--x abc.sh

Here the root’s permission is **rwx** (**read**, **write** and **execute**).  
usergroup to which it belongs, is **r-x** (**read** and **execute** only, no write permission) and  
for world is **–x** (only **execute**).

To change its permission and provide **read**, **write** and **execute** permission to owner, group and world.

root@tecmint:~# chmod 777 abc.sh

only **read** and **write** permission to all three.

root@tecmint:~# chmod 666 abc.sh

**read**, **write** and **execute** to **owner** and only execute to **group** and **world.**

root@tecmint:~# chmod 711 abc.sh

Note: one of the most important command useful for sysadmin and user both. On a multi-user environment or on a server, this command comes to rescue, setting wrong permission will either makes a file inaccessible or provide unauthorized access to someone.

**11. Command: chown**

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

Here the directory **Binary** is owned by user “**server**” and it belongs to usergroup “**root**” where as directory “**Desktop**” is owned by user “**server**” and belongs to user group “**server**“.

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.

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

Note: “**chown**” changes the user and group ownership of each given **FILE** to **NEW-OWNER** or to the user and group of an existing reference file.

**12. Command: apt**

The Debian based “**apt**” command stands for (**Advanced Package Tool**). **Apt** is an advanced package manager for **Debian** based system (**Ubuntu**, **Kubuntu**, etc.), that automatically and intelligently **search**, **install**, **update** and **resolves dependency** of packages on **Gnu/Linux** system from command line.

root@tecmint:~# apt-get install mplayer

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following package was automatically installed and is no longer required:

java-wrappers

Use 'apt-get autoremove' to remove it.

The following extra packages will be installed:

esound-common libaudiofile1 libesd0 libopenal-data libopenal1 libsvga1 libvdpau1 libxvidcore4

Suggested packages:

pulseaudio-esound-compat libroar-compat2 nvidia-vdpau-driver vdpau-driver mplayer-doc netselect fping

The following NEW packages will be installed:

esound-common libaudiofile1 libesd0 libopenal-data libopenal1 libsvga1 libvdpau1 libxvidcore4 mplayer

0 upgraded, 9 newly installed, 0 to remove and 8 not upgraded.

Need to get 3,567 kB of archives.

After this operation, 7,772 kB of additional disk space will be used.

Do you want to continue [Y/n]? y

root@tecmint:~# apt-get update

Hit http://ppa.launchpad.net raring Release.gpg

Hit http://ppa.launchpad.net raring Release.gpg

Hit http://ppa.launchpad.net raring Release.gpg

Hit http://ppa.launchpad.net raring Release.gpg

Get:1 http://security.ubuntu.com raring-security Release.gpg [933 B]

Hit http://in.archive.ubuntu.com raring Release.gpg

Hit http://ppa.launchpad.net raring Release.gpg

Get:2 http://security.ubuntu.com raring-security Release [40.8 kB]

Ign http://ppa.launchpad.net raring Release.gpg

Get:3 http://in.archive.ubuntu.com raring-updates Release.gpg [933 B]

Hit http://ppa.launchpad.net raring Release.gpg

Hit http://in.archive.ubuntu.com raring-backports Release.gpg

Note: The above commands results into system-wide changes and hence requires root password (Check ‘**#**‘ and not ‘**$’** as prompt). **Apt** is considered more advanced and intelligent as compared to [yum command](http://www.tecmint.com/20-practical-examples-of-rpm-commands-in-linux/).

As the name suggest, **apt-cache** search for package containing sub package **mpalyer**. **apt-get** install, update all the packages, that are already installed, to the newest one.

Read more about apt-get and apt-cache commands at [25 APT-GET and APT-CACHE Commands](http://www.tecmint.com/useful-basic-commands-of-apt-get-and-apt-cache-for-package-management/)

**13. Command: tar**

The “**tar**” command is a **Tape Archive** is useful in creation of archive, in a number of file format and their extraction.

root@tecmint:~# tar -zxvf abc.tar.gz (Remember '**z**' for **.tar.gz**)

root@tecmint:~# tar -jxvf abc.tar.bz2 (Remember '**j**' for **.tar.bz2**)

root@tecmint:~# tar -cvf archieve.tar.gz(**.bz2**) /path/to/folder/abc

Note: A ‘**tar.gz**‘ means gzipped. ‘**tar.bz2**‘ is compressed with bzip which uses a better but slower compression method.

Read more about “tar command” examples at [18 Tar Command Examples](http://www.tecmint.com/18-tar-command-examples-in-linux/)

**14. Command: 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.

root@tecmint:~# cal

May 2013

Su Mo Tu We Th Fr Sa

1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30 31

Show calendar of year **1835** for month **February**, that already has passed.

root@tecmint:~# cal 02 1835

February 1835

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

Shows calendar of year **2145** for the month of **July**, that will advancing

root@tecmint:~# cal 07 2145

July 2145

Su Mo Tu We Th Fr Sa

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

Note: You need not to turn the calendar of **50** years back, neither you need to make complex mathematical calculation to know what day you were worn or your coming birthday will fall on which day.

**15. Command: date**

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

root@tecmint:~# date

Fri May 17 14:13:29 IST 2013

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

Mon May 13 13:57:00 IST 2013

Note: This Command will be very use-full in scripting, time and date based scripting, to be more perfect. Moreover changing date and time using terminal will make you feel **GEEK**!!!. (Obviously you need to be **root** to perform this operation, as it is a system wide change).

**16. Command: cat**

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

root@tecmint:~# cat a.txt b.txt c.txt d.txt >> abcd.txt

root@tecmint:~# cat abcd.txt

....

contents of file abcd

...

Note: “**>>**” and “**>**” are called append symbol. They are used to append the output to a file and not on standard output. “**>**” symbol will delete a file already existed and create a new file hence for security reason it is advised to use “**>>**” that will write the output without overwriting or deleting the file.

Before Proceeding further, I must let you know about wildcards (you would be aware of wildcard entry, in most of the Television shows) Wildcards are a shell feature that makes the command line much more powerful than any **GUI** file managers. You see, if you want to select a big group of files in a graphical file manager, you usually have to select them with your mouse. This may seem simple, but in some cases it can be very frustrating.

For example, suppose you have a directory with a huge amount of all kinds of files and subdirectories, and you decide to move all the HTML files, that have the word “Linux” somewhere in the middle of their names, from that big directory into another directory. What’s a simple way to do this? If the directory contains a huge amount of differently named HTML files, your task is everything but simple!

In the Linux CLI that task is just as simple to perform as moving only one HTML file, and it’s so easy because of the shell wildcards. These are special characters that allow you to select file names that match certain patterns of characters. This helps you to select even a big group of files with typing just a few characters, and in most cases it’s easier than selecting the files with a mouse.

Here’s a list of the most commonly used wildcards :

**Wildcard** **Matches**

\* zero or more characters

? exactly one character

[abcde] exactly one character listed

[a-e] exactly one character in the given range

[!abcde] any character that is not listed

[!a-e] any character that is not in the given range

{debian,linux} exactly one entire word in the options given

**!** is called not symbol, and the reverse of string attached with ‘!’ is true.

Read more examples of Linux “cat command” at [13 Cat Command Examples in Linux](http://www.tecmint.com/13-basic-cat-command-examples-in-linux/)

**17. Command: cp**

The “**copy**” stands for (**Copy**), it copies a file from one location to another location.

root@tecmint:~# cp /home/user/Downloads abc.tar.gz /home/user/Desktop (Return 0 when sucess)

Note: **cp** is one of the most commonly used command in shell scripting and it can be used with wildcard characters (Describe in the above block), for customised and desired file copying.

**18. Command: mv**

The “**mv**” command moves a file from one location to another location.

root@tecmint:~# mv /home/user/Downloads abc.tar.gz /home/user/Desktop (Return 0 when sucess)

Note: **mv** command can be used with wildcard characters. **mv** should be used with caution, as moving of system/unauthorised file may lead to security as well as breakdown of system.

**19. Command: pwd**

The command “**pwd**” (**print working directory**), prints the current working directory with full path name from terminal.

root@tecmint:~# pwd

/home/user/Desktop

Note: This command won’t be much frequently used in scripting but it is an absolute life saver for newbie who gets lost in terminal in their early connection with nux. (**Linux** is most commonly referred as **nux** or **nix**).

**20. Command: cd**

Finally, the frequently used “**cd**” command stands for (**change directory**), it change the working directory to execute, copy, move write, read, etc. from terminal itself.

root@tecmint:~# cd /home/user/Desktop

server@localhost:~$ pwd

/home/user/Desktop

Note: **cd** comes to rescue when switching between directories from terminal. “**Cd ~**” will change the working directory to user’s home directory, and is very useful if a user finds himself lost in terminal. “**Cd ..**” will change the working directory to parent directory (of current working directory).