Table of Contents

Shutdown command 2
Switch to root user 2
Add new user- useradd or adduser 2
passwd-Set password 3
Switch between different users: 3
Usermod Command- 3
userdel- 3
Terminal commands 4
Linux Basic Commands 4
1) pwd command(print working directory) 4
2) cal command(cal) 4
3) echo command 4
4) date command 5
5) tty command 5
6) whoami command 5
7) id command 5
8) clear command 5
9) help option 5
10) whatis command 6
11) w command 6
12) history command 6
13) Manual Pages 6
14) bc command 6
Command with advanced optional 7
1. who command 7
2. how to open multiple terminal:- 8
3. Pipes 8
4. tee command 9
5. input output redirection 9
6. SORT Command 10
7. PASTE Command 11
8. **GREP Command** 12
Process commands 14

Lab Manual "Working With Linux"

B.Sc 6th Sem Computer Application

Shutdown command

1. $ shutdown 8:00

Schedule the system to shut down at 8 A.M.

2. $ shutdown 20:00

Schedule the system to shut down at 8 P.M.

3. $ shutdown +15 "Upgrading hardware, downtime should be minimal"

Schedule the system to shut down in fifteen minutes. Along with the normal message alerting users that the system is shutting down, they will be given the descriptive message about a hardware upgrade.

4. $ shutdown now

Bring down the system immediately.

5. $shutdown -r now

Bring down the system immediately, and automatically reboot it.

6. $ shutdown -P now

Bring down the system immediately, and automatically power off the system.

7. $ shutdown -t45

tells Linux to wait 45 seconds before starting the shutdown process. The time argument often requires the -t option ahead of it, but some versions assume any number is the number of seconds. If you want to start the shutdown process immediately, you can set the number of seconds to zero or on some systems type the word now:

Switch to root user

[Bsingh@localhost ~]$ su
Password:
[root@localhost Bsingh]#

Notice the $ sign changes to # and we also switch from Bsingh user account to root account.

Add new user- useradd or adduser

1. [root@localhost Bsingh]# useradd ghg

2.[root@localhost Bsingh]# adduser -u 100 -g 100 -c "new linux user" -e 2020-01-22 -d /college ghg

passwd-Set password

[root@localhost Bsingh]# passwd ghg

Changing password for user ghg.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@localhost Bsingh]#

Switch between different users:

[Bsingh@localhost ~]$ su ghg

Password:

[ghg@localhost Bsingh]$su

Password:

[root@localhost Bsingh]#su ghg

[ghg@localhost Bsingh]$su Bsingh

Password:

[Bsingh@localhost ~]$

Usermod Command-

usermod -c "This is ghg" ghg // to change to description

usermod -e 2014-11-01 tecmint // to change the expiry date of account

usermod -l ghg_admin ghg // to change the login name

userdel-

to delete the user account

Syntax:-

userdel [options] userName
userdel -r userName

The following is recommend procedure to delete a user from the Linux server. First, lock user account, enter:

passwd -l ghg or **passwd --lock tecmint**

killall -KILL -u ghg
userdel -r ghg

Terminal commands

Linux Basic Commands

Let's start with some simple commands.

1) pwd command(print working directory)

'pwd' command prints the absolute path to current working directory.

```
$ pwd
/home/Bsingh
```

2) cal command(cal)

Displays the calendar of the current month.

```
$ cal
Jan 2020
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
```

'cal ' will display calendar for the specified month and year.

```
$ cal 08 1991
August 1991
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
```

3) echo command

This command will echo whatever you provide it.

```
$ echo "Bsingh"
Bsingh
```

The 'echo' command is used to display the values of a variable. One such variable is 'HOME'. To check the value of a variable precede the variable with a $ sign.

$ echo $HOME
/home/Bsingh

4) date command

Displays current time and date.

$ date
Fri Jan 24 09:07:09 IST 2020

If you are interested only in time, you can use 'date +%T' (in hh:mm:ss):

$ date +%T
01:13:14

5) tty command

Displays current terminal.

$ tty
/dev/pts/1

6) whoami command

This command reveals the user who is currently logged in.

$ whoami
Bsingh

[root@localhost Bsingh]# who am i
Bsingh pts/1 2020-01-24 11:19 (:0)

7) id command

This command prints user and groups (UID and GID) of the current user.

$ id
uid=0(root) gid=0(root) groups=0(root)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

By default, information about the current user is displayed. If another username is provided as an argument, information about that user will be printed:

$ id root
uid=0(root) gid=0(root) groups=0(root)

8) clear command

This command clears the screen.

9) help option

Nobody can remember all the commands. We can use help option from command like

With almost every command, '--help' option shows usage summary for that command.

$ date --help
Usage: date [OPTION]... [+FORMAT] or: date [-u|--utc|--universal]

[MMDDhhmm[[CC]YY][.ss]] Display the current time in the given FORMAT, or set the system date.

10) whatis command

This command gives a one line description about the command. It can be used as a quick reference for any command.

$ **whatis** date
date (1) - print or set the system date and time

[root@localhost Bsingh]# whatis who
who (1) - show who is logged on

11) w command

w command is used to check which users are logged in to the system, and what command they are executing at that particular time:

[Bsingh@localhost ~]$ w

```
20:52:02 up 1 day, 10:55,  3 users,  load average: 0.32, 0.35, 0.25
USER     TTY      LOGIN@   IDLE   JCPU   PCPU WHAT
Bsingh   tty1     Thu09   34:54m  3:31   0.07s /usr/bin/startplasma-x11
Bsingh   pts/0    Thu09   34:54m  0.00s  6.00s kded5 [kdeinit5]
Bsingh   pts/1    11:19    1.00s  0.19s  0.06s /bin/bash
```

12) history command

History command shows the commands you have entered on your terminal so far.

Bsingh@localhost ~]$ **history**

```
 1  systemctl disable dnf -makecache.service
 2  su
 3  sudo dnf install ./rpmfusion-
 4  sudo dnf install ./rpmfusion-free-release-31.noarch.rpm
 5  su
 6  hostname
 7  hostname fedora31
 8  sudo hostname fedora31
 9  sudoedit
10  sudo hostname fedora31
```

13) Manual Pages

'--help' option and 'whatis' command do not provide thorough information about the command. For more detailed information, Linux provides man pages and info pages. To see a command's manual page, man command is used.

[Bsingh@localhost ~]$ **man** date

14) bc command

This command is used to perform basic arithmetic operation.

[Bsingh@localhost ~]$ bc
bc 1.07.1

Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5+6
11
5*6
30
30/2
15
quit
[Bsingh@localhost ~]$

to quit from calculator type command quit.

Command with advanced optional

1. who command

[root@localhost home]# who

Bsingh tty1 2020-01-21 18:51 (:0)

Bsingh pts/0 2020-01-21 18:51 (:0)

Bsingh pts/1 2020-01-21 21:35 (:0)

[root@localhost home]# who -a

system boot 2020-01-22 00:20

run-level 5 2020-01-21 18:51

Bsingh + tty1 2020-01-21 18:51 old 1205 (:0)

Bsingh + pts/0 2020-01-21 18:51 old 1265 (:0)

Bsingh - pts/1 2020-01-21 21:35 old 4495 (:0)

[root@localhost home]# who -aH

NAME LINE TIME IDLE PID COMMENT EXIT

system boot 2020-01-22 00:20

run-level 5 2020-01-21 18:51

Bsingh + tty1 2020-01-21 18:51 old 1205 (:0)

Bsingh + pts/0 2020-01-21 18:51 old 1265 (:0)

Bsingh - pts/1 2020-01-21 21:35 old 4495 (:0)

[root@localhost home]# who -q

Bsingh Bsingh Bsingh

users=3

[root@localhost home]# who -uH

NAME LINE TIME IDLE PID COMMENT

Bsingh tty1 2020-01-21 18:51 old 1205 (:0)

Bsingh pts/0 2020-01-21 18:51 old 1265 (:0)

Bsingh pts/1 2020-01-21 21:35 old 4495 (:0)

2. how to open multiple terminal:-

user can open multiple terminal by pressing alt+ctlr+f1..f6 that is hit alt+ctlr+f2 and alt+ctlr+f5 to open terminal 2 & 4 and login with username and password. Now open terminal 1 and execute who command to check user currently logged in.

- This continues upto tty6 i.e. default number of allowed tty consoles are 6
- One can switch from tty1 to tty6 using Ctrl+Alt+F[1-6] on the console

[Bsingh@localhost ~]$ who
Bsingh tty1 2020-01-22 12:29 (:0)
Bsingh pts/0 2020-01-22 12:29 (:0)
Bsingh pts/1 2020-01-22 12:48 (:0)
root tty2 2020-01-22 12:51
Bsingh tty5 2020-01-22 12:52

note

if you are on terminal 2 and then use **logout** command to logout the current user.

3. Pipes

An example of a command using a pipe:

$ls -l|cat >file.txt

This takes the output of ls, which displays the contents of your current directory, and *pipes* it to the *text file with name file*.

[Bsingh@localhost ~]$ ls -l|cat >file.txt

[Bsingh@localhost ~]$ cat file.txt

total 40

drwxr-xr-x. 2 Bsingh Bsingh 4096 Jan 10 14:00 Desktop

drwxr-xr-x. 4 Bsingh Bsingh 4096 Jan 21 22:13 Documents

drwxr-xr-x. 6 Bsingh Bsingh 4096 Jan 11 11:51 Downloads

-rw-rw-r--. 1 Bsingh Bsingh 4564 Jan 19 18:57 file2.txt

-rw-rw-r--. 1 Bsingh Bsingh 0 Jan 21 22:30 file.txt

drwxr-xr-x. 2 Bsingh Bsingh 4096 Nov 23 00:32 Music

4. tee command

How to write to standard output and a file

To write to standard output and a file use tee after a pipe and specify the file or files to write to.

```
echo 'Bsingh' | tee linux.txt
Bsingh
cat linux.txt
Bsingh
```

This writes the output of the first command to standard output and to a file called linux.txt

How to write to a file and append output

To use tee and append to a file rather use the -a option.

```
cat linux.txt
Bsingh
echo 'ghg' | tee -a linux.txt
ghg
cat linux.txt
Bsingh
ghg
```

Example 1: Write output to stdout, and also to a file

The following command displays output only on the screen (stdout).

```
$ ls
```

The following command writes the output only to the file and not to the screen.

```
$ ls > file
```

The following command (with the help of tee command) writes the output both to the screen (stdout) and to the file.

```
$ ls | tee file
```

5. input output redirection

Output Redirection

```
[Bsingh@localhost ~]$ who > userinfo.txt
[Bsingh@localhost ~]$ cat userinfo.txt
Bsingh   tty1        2020-01-22 12:29 (:0)
Bsingh   pts/0       2020-01-22 12:29 (:0)
Bsingh   pts/1       2020-01-22 21:30 (:0)
root     tty2        2020-01-22 14:00
[Bsingh@localhost ~]$
```

Input redirection

[Bsingh@localhost ~]$ wc -l < userinfo.txt

4

Append

Commands with a double bracket '>>' **do not overwrite** the existing file content.

For example..

[Bsingh@localhost ~]$ cat > city.txt
sadhar
ludhiana
delhi
[Bsingh@localhost ~]$ cat city.txt
sadhar
ludhiana
delhi
[Bsingh@localhost ~]$ cat >>city.txt
moga
barnala
chandigarh
[Bsingh@localhost ~]$ cat city.txt
sadhar
ludhiana
delhi
moga
barnala
chandigarh

Note:- if user use > symbol instead of >> then data will be overwrite to existing file.

6. SORT Command

[Bsingh@localhost ~]$ sort city.txt
barnala
chandigarh
delhi
ludhiana
moga
sadhar

[Bsingh@localhost ~]$ sort -r city.txt
sadhar
moga
ludhiana
delhi
chandigarh
barnala
[Bsingh@localhost ~]$ sort city.txt > sorted.txt
[Bsingh@localhost ~]$ cat sorted.txt
barnala
chandigarh
delhi
ludhiana
moga
sadhar
[Bsingh@localhost ~]$ cat city.txt | sort -r
sadhar
moga
ludhiana
delhi
chandigarh
barnala

7. PASTE Command

[Bsingh@localhost ~]$ cat city.txt
sadhar
ludhiana
delhi
moga
barnala
chandigarh
[Bsingh@localhost ~]$ cat >city_population.txt
10000
50000
80000
25000
30000
45000
[Bsingh@localhost ~]$ cat city_population.txt
10000
50000
80000
25000

30000
45000

[Bsingh@localhost ~]$ paste city.txt city_population.txt
sadhar 10000
ludhiana   50000
delhi   80000
moga   25000
barnala 30000
chandigarh   45000
[Bsingh@localhost ~]$ paste -d " " city.txt city_population.txt
sadhar 10000
ludhiana 50000
delhi 80000
moga 25000
barnala 30000
chandigarh 45000

[Bsingh@localhost ~]$ paste -s city.txt city_population.txt
sadhar ludhiana delhi   moga   barnala chandigarh
10000   50000   80000   25000   30000   45000

8. GREP Command

[Bsingh@localhost ~]$ grep linux data.txt
linux Training
linux Support
linux Jobs
linux India Events

[Bsingh@localhost ~]$ grep -c linux data.txt
4

[Bsingh@localhost ~]$ grep -i linux data.txt
Linux India
What is Linux
Linux Distributions
linux Training
linux Support
linux Jobs
linux India Events
Important features of Linux Operating System
Following are some of the important features of Linux Operating System.
Portable – Portability means softwares can works on different types of hardwares in same way.Linux kernel and application programs supports their installation on any kind of hardware platform.

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.

[Bsingh@localhost ~]$ grep -n linux data.txt
5:linux Training
6:linux Support
7:linux Jobs
8:linux India Events

[Bsingh@localhost ~]$ grep -v linux data.txt
Linux India
What is Linux
Linux Distributions
Articles
Important features of Linux Operating System
Basic Features
Following are some of the important features of Linux Operating System.
Portable – Portability means softwares can works on different types of hardwares in same way.Linux kernel and application programs supports their installation on any kind of hardware platform.
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.

[Bsingh@localhost ~]$ grep -o linux data.txt
linux
linux
linux
linux
[Bsingh@localhost ~]$ grep linu[xs] data.txt
linus torvalds
linux Training
linux Support
linux Jobs
linux India Events
[Bsingh@localhost ~]$ grep linu[a-x] data.txt
linus torvalds
linux Training
linux Support
linux Jobs
linux India Events
[Bsingh@localhost ~]$ grep ^lin data.txt
linus torvalds

linux Training
linux Support
linux Jobs
linux India Events

[Bsingh@localhost ~]$ grep li..x data.txt

linux Training
linux Support
linux Jobs
linux India Events

[Bsingh@localhost ~]$ grep B\.Singh data.txt

B.Singh

Process commands

1. If you run ps command without any arguments, it displays processes for the current shell.

$ ps

[Bsingh@localhost ~]$ ps

```
  PID TTY          TIME CMD
 2739 pts/1    00:00:00 bash
 4244 pts/1    00:00:00 ps
```

2. **Display every active process on a Linux system in generic (Unix/Linux) format**.

$ ps -A
OR
$ ps -e

3. **To perform a full-format listing, add the -f or -F flag.**

$ ps -ef
OR
$ ps -eF

4. [Bsingh@localhost ~]$ ps -ef

```
UID        PID  PPID  C STIME TTY          TIME CMD

root         2     0  0 19:04 ?        00:00:00 [kthreadd]
root         3     2  0 19:04 ?        00:00:00 [rcu_gp]
root         4     2  0 19:04 ?        00:00:00 [rcu_par_gp]
root         9     2  0 19:04 ?        00:00:00 [mm_percpu_wq]
root        10     2  0 19:04 ?        00:00:00 [ksoftirqd/0]
root        11     2  0 19:04 ?        00:00:01 [rcu_sched]
```

```
root        12     2     0 19:04      ?        00:00:00      [migration/0]
root        13     2     0 19:04      ?        00:00:00      [cpuhp/0]
```

4. **See process run by username**

Select by process by effective user ID (EUID) or name by passing username such as Bsingh:

```
# ps -u Bsingh
```

5. **Linux running processes with top command**

The top program provides a dynamic real-time view of a running system. Type the top at command prompt:

```
# top
```

6. **to display a tree of processes**

The pstree command shows running processes as a tree. The tree is rooted at either pid or init if pid is omitted. If a user name is specified, all process trees rooted at processes owned by that user are shown.

```
$ pstree
```

7. **Running a Job in the Background and foreground**

To run a job in the background, you need to enter the command that you want to run, followed by an **ampersand (&)** symbol at the end of the command line. For example, run the sleep command in the background.

```
$ sleep 100 &
[1] 1302
$
```

The shell returns the job ID, in brackets, that it assigns to the command and the associated PID. With the job ID, you can use the job control commands to manage the job whereas the kernel uses PIDs to manage jobs.

When a background job is complete and you press Return, the shell displays a message indicating the job is done.

```
[1] + Done       sleep 100 &
$
```

Managing the background jobs

You can use the **jobs** command to list the jobs that are currently running or suspended in the background.

```
$ jobs
[1]+  Running              sleep 100 &
```

You can use the fg command to bring a background job to the foreground.

```
$ fg % 1
sleep 100
```

8.Kill Command

Kill command send a signal, a specified signal to be more perfect to a process. The kill command can be executed in a number of ways, directly or from a shell script.

The common syntax for **kill command** is:

```
# kill [signal or option] PID(s)
```

The most common kill signals are:

Signal Name	Single Value	Effect
SIGHUP	1	Hangup
SIGINT	2	Interrupt from keyboard
SIGKILL	9	Kill signal
SIGTERM	15	Termination signal
SIGSTOP	17, 19, 23	Stop the process

```
$kill -9 3827

$kill -9 3919

$kill -9 10764
```

Another method to kill process by name

```
$killall -9 chrome
```