

# **FACULTY OF ENGINEERING & TECHNOLOGY Operating System (Subject Code)**

## **B. Tech. 2nd Year**

### **PRACTICAL NO: 1**

Definition: Study of Basic commands of Linux/UNIX.

Command shell: A program that interprets commands is Command shell.

Shell Script: Allows a user to execute commands by typing them manually at a terminal, or automatically in programs called shell scripts. A shell is not an operating system. It is a way to interface with the operating system and run Commands.

BASH (Bourne Again Shell)

☐ Bash is a shell written as a free replacement to the standard Bourne Shell (/bin/sh) originally written by Steve Bourne for UNIX systems.

☐ It has all of the features of the original Bourne Shell, plus additions that make it easier to program with and use from the command line.

☐ Since it is Free Software, it has been adopted as the default shell on most Linux systems.

BASIC LINUX COMMANDS:

#### **1. pwd : Print Working Directory**

DESCRIPTION:

pwd prints the full pathname of the current working directory.

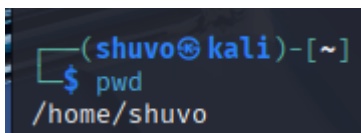
SYNTAX:

pwd

EXAMPLE:

\$ pwd

/home/directory\_name



## 2. cd: Change Directory

### DESCRIPTION:

It allows you to change your working directory. You use it to move around within the hierarchy of your file system.

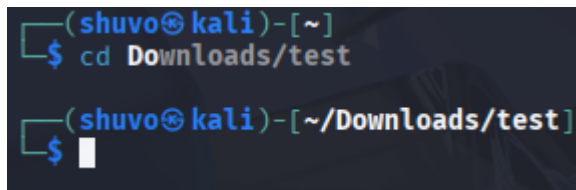
### SYNTAX:

```
cd directory_name
```

### EXAMPLE:

To change into “work directory” in “documents” need to write as follows.

Input: `$ cd /documents/work`



```
(shuvo@kali)-[~]  
$ cd Downloads/test  
  
(shuvo@kali)-[~/Downloads/test]  
$
```

## 3. cd ..

### DESCRIPTION:

Move up one directory.

### SYNTAX:

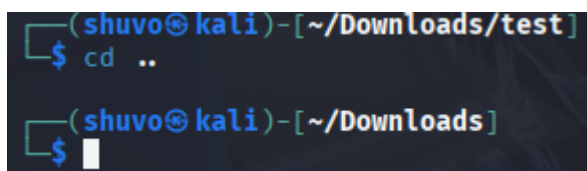
```
cd ..
```

### EXAMPLE:

If you are in work directory and want to go to documents then write

```
cd ..
```

You will end up in /documents.



```
(shuvo@kali)-[~/Downloads/test]  
$ cd ..  
  
(shuvo@kali)-[~/Downloads]  
$
```

## 4. ls : list all the files and directories

### DESCRIPTION:

List all files and folders in the current directory in the column format.

#### SYNTAX:

ls [options]

EXAMPLE: Using various options

▣ Lists the total files in the directory and subdirectories, the names of the files in the current directory, their permissions, the number of subdirectories in directories listed, the size of the file, and the date of last modification.

```
(shuvo@kali)-[~/Downloads]
$ ls
'codetantra-sea_4.0.0_amd64(1).deb'  DEBIAN  opt  test
codetantra-sea_4.0.0_amd64.deb      Dnxd-3U0.html  pic.jpg  usr
```

ls -l

▣ List all files including hidden files

```
(shuvo@kali)-[~/Downloads]
$ ls -l
total 168056
-rw-rw-r-- 1 shuvo shuvo 85722944 Sep 19 19:13 'codetantra-sea_4.0.0_amd64(1).deb'
-rw-rw-r-- 1 shuvo shuvo 85722944 Sep 16 17:44 codetantra-sea_4.0.0_amd64.deb
drwxr-xr-x 2 shuvo shuvo 4096 Sep 2 20:36 DEBIAN
-rw-rw-r-- 1 shuvo shuvo 0 Jul 12 03:18 Dnxd-3U0.html
drwxr-xr-x 3 shuvo shuvo 4096 Sep 2 20:36 opt
-rw-rw-r-- 1 shuvo shuvo 613878 Jul 11 01:20 pic.jpg
drwxrwxr-x 3 shuvo shuvo 4096 Dec 5 09:07 test
drwxr-xr-x 3 shuvo shuvo 4096 Sep 2 20:36 usr
```

ls -a

```
(shuvo@kali)-[~/Downloads]
$ ls -a
.  'codetantra-sea_4.0.0_amd64(1).deb'  DEBIAN  opt  test
.. codetantra-sea_4.0.0_amd64.deb      Dnxd-3U0.html  pic.jpg  usr
```

## 5. cat

#### DESCRIPTION:

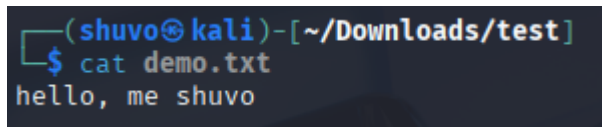
cat stands for "catenate". It reads data from files, and outputs their contents. It is the simplest way to display the contents of a file at the command line.

#### SYNTAX:

cat filename

#### EXAMPLES:

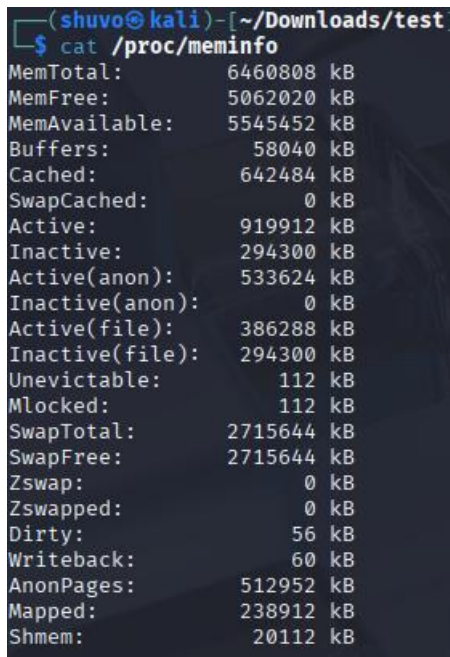
Print the contents of files mytext.txt and yourtext.txt `cat mytext.txt yourtext.txt`



```
(shuvo@kali)-[~/Downloads/test]
$ cat demo.txt
hello, me shuvo
```

Print the cpu information using cat command `cat /proc/cpuinfo` Print the memory information using cat command

`cat /proc/meminfo`



```
(shuvo@kali)-[~/Downloads/test]
$ cat /proc/meminfo
MemTotal:        6460808 kB
MemFree:         5062020 kB
MemAvailable:    5545452 kB
Buffers:         58040 kB
Cached:          642484 kB
SwapCached:      0 kB
Active:          919912 kB
Inactive:        294300 kB
Active(anon):    533624 kB
Inactive(anon):  0 kB
Active(file):    386288 kB
Inactive(file):  294300 kB
Unevictable:     112 kB
Mlocked:         112 kB
SwapTotal:       2715644 kB
SwapFree:        2715644 kB
Zswap:           0 kB
Zswapped:        0 kB
Dirty:           56 kB
Writeback:       60 kB
AnonPages:       512952 kB
Mapped:          238912 kB
Shmem:           20112 kB
```

## 6. head

### DESCRIPTION:

head, by default, prints the first 10 lines of each FILE to standard output. With more than one FILE, it precedes each set of output with a header identifying the file name.

If no FILE is specified, or when FILE is specified as a dash ("-"), head reads from standard input.

### SYNTAX:

`head [option]...[file/directory]`

### EXAMPLE:

Display the first ten lines of myfile.txt.

`head myfile.txt`

```
(shuvo@kali)-[~/Downloads/test]
$ head demo.txt
hello, me shuvo
```

## 7. tail

### DESCRIPTION:

tail is a command which prints the last few number of lines (10 lines by default) of a certain file, then terminates.

### SYNTAX:

tail [option]...[file/directory]

### EXAMPLE:

Output the last 100 lines of the file myfile.txt.

tail myfile.txt -n 100

```
(shuvo@kali)-[~/Downloads/test]
$ tail -n 100 demo.txt
61. Funnel marketing guides users step-by-step.
62. Top-of-funnel focuses on awareness.
63. Middle-of-funnel focuses on engagement.
64. Bottom-of-funnel focuses on conversions.
65. Retargeting helps bring back lost visitors.
66. Offer-based campaigns boost quick sales.
67. Seasonal campaigns improve engagement.
68. Holiday marketing drives high traffic.
69. Market research identifies customer needs.
70. Buyer personas help tailor marketing messages.
71. Digital PR improves online reputation.
72. Press releases can increase brand authority.
```

## 8. mv : Moving (and Renaming) Files

### DESCRIPTION:

The mv command lets you move a file from one directory location to another. It also lets you rename a file (there is no separate rename command).

### SYNTAX:

mv [option] source directory

### EXAMPLE:

☐ Moves the file myfile.txt to the directory destination-directory.

mv myfile.txt destination\_directory

```
(shuvo@kali)-[~/Downloads/test]
$ mv demo.txt /home/shuvo/Downloads

(shuvo@kali)-[~/Downloads/test]
$ ls

(shuvo@kali)-[~/Downloads/test]
$ cd ..

(shuvo@kali)-[~/Downloads]
$ ls
'codetantra-sea_4.0.0_amd64(1).deb'  demo.txt  pic.jpg
codetantra-sea_4.0.0_amd64.deb      Dnxd-3U0.html  test
DEBIAN                             opt         usr
```

☐ Move the file myfile.txt into the parent directory.

```
mv myfile.txt ../
```

```
(shuvo@kali)-[~/Downloads]
$ mv demo.txt ../

(shuvo@kali)-[~/Downloads]
$ ls
'codetantra-sea_4.0.0_amd64(1).deb'  DEBIAN  opt  test
codetantra-sea_4.0.0_amd64.deb      Dnxd-3U0.html  pic.jpg  usr
```

☐ In this case, if JOE1\_expenses does not exist, it will be created with the exact content ofjoe\_expenses, and joe\_expenses will disappear.

If JOE1\_expenses already exists, its content will be replaced with that of joe\_expenses (and joe\_expenses will still disappear).

```
mv joe_expenses JOE1_expenses
```

## 9. mkdir : Make Directory

DESCRIPTION: If the specified directory does not already exist, mkdir creates it. More than one directory may be specified when calling mkdir.

SYNTAX:

```
mkdir [option] directory
```

EXAMPLE:

Create a directory named work.

```
mkdir work
```

```
(shuvo@kali)-[~/Downloads/test]
$ mkdir work folder

(shuvo@kali)-[~/Downloads/test]
$ ls
demo.txt  folder  work

(shuvo@kali)-[~/Downloads/test]
$
```

## 10. cp : Copy Files

### DESCRIPTION:

The cp command is used to make copy of files and directories.

### SYNTAX:

cp [option] source directory

### EXAMPLE:

Creates a copy of the file in the currently working directory named origfile. The copy will be named newfile, and will be located in the working directory.

cp origfile newfile

```
(shuvo@kali)-[~/Downloads/test]
$ cp demo.txt demo_cooopy.txt

(shuvo@kali)-[~/Downloads/test]
$ ls
demo_cooopy.txt  demo.txt  folder  work
```

## 11. rmdir : Remove Directory

### DESCRIPTION:

The rmdir command is used to remove a directory that contains other files or directories.

### SYNTAX:

rm directory\_name

### EXAMPLE:

Delete mydir directory along with all files and directories within that directory. Here, -r is for recursive and -f is for forcefully.

rmdir -rf mydir



```
(shuvo@kali)-[~/Downloads/test]
$ rm -rf work

(shuvo@kali)-[~/Downloads/test]
$ ls
demo.txt
```

## 12. gedit

### DESCRIPTION:

The gedit command is used to create and open a file.

### SYNTAX:

gedit filename.txt

### EXAMPLE:

To create a file named abc.sh

gedit abc.sh

```
(shuvo@kali)-[~/Downloads/test]
$ sudo apt install gedit
You might want to run 'apt --fix-broken install' to correct these.
Unsatisfied dependencies:
anydesk : Depends: libgtk2.0-0 (≥ 2.20.1)
          Depends: libgtkglext1 but it is not installable
gedit : Depends: gedit-common (< 49~) but it is not going to be
         installed
         Depends: gedit-common (≥ 48~) but it is not going to b
```

## 13. man

### DESCRIPTION:

Displays on online manual page or manpage.

### SYNTAX:

man command

### EXAMPLE:

To learn about listing files

man ls



```
File Actions Edit View Help
RM(1)                                User Commands                                RM(1)

NAME
  rm - remove files or directories

SYNOPSIS
  rm [OPTION] ... [FILE] ...

DESCRIPTION
  This manual page documents the GNU version of rm.  rm
  removes each specified file.  By default, it does not
  remove directories.

  If the -I or --interactive=once option is given, and
  there are more than three files or the -r, -R, or --re-
  cursive are given, then rm prompts the user for whether
  to proceed with the entire operation.  If the response
  is not affirmative, the entire command is aborted.

  Otherwise, if a file is unwritable, standard input is a
  terminal, and the -f or --force option is not given, or
  the -i or --interactive=always option is given, rm
  prompts the user for whether to remove the file.  If the
  response is not affirmative, the file is skipped.

OPTIONS
  Manual page rm(1) line 1 (press h for help or q to quit)
```

#### 14. echo

##### DESCRIPTION:

Display text on the screen.

##### SYNTAX:

echo yourtext

##### EXAMPLE:

Print Hello World on the screen

echo "Hello World"

```
(shuvo@kali)-[~/Downloads/test]
$ echo "Hello, Me shuvo."
Hello, Me shuvo.
```

#### 15. clear

##### DESCRIPTION:

Used to clear the screen

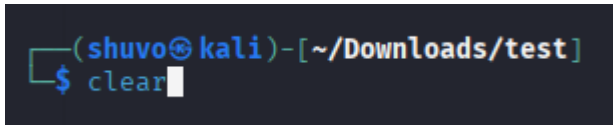
SYNTAX:

clear

EXAMPLE:

Clear the entire screen

clear



```
(shuvo@kali) - [~/Downloads/test]  
$ clear
```

## 16. whoami

DESCRIPTION:

whoami prints the effective user ID. This command prints the username associated with the current effective user ID.

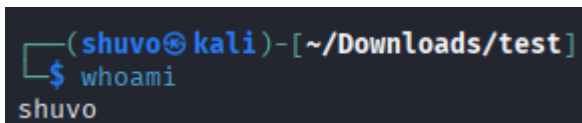
SYNTAX:

whoami [option]

EXAMPLE:

Display the name of the user who runs the command.

whoami



```
(shuvo@kali) - [~/Downloads/test]  
$ whoami  
shuvo
```

## 17. wc

DESCRIPTION:

wc (word count) command, can return the number of lines, words, and characters in a file.

SYNTAX:

wc [option]... [file]...

EXAMPLE:

Print the byte counts of file myfile.txt

wc -c myfile.txt

```
(shuvo@kali)-[~/Downloads/test]
$ wc -c test.txt
158 test.txt
```

Print the line counts of file myfile.txt

```
wc -l myfile.txt
```

```
(shuvo@kali)-[~/Downloads/test]
$ wc -l test.txt
5 test.txt
```

Print the word counts of file myfile.txt

```
wc -w myfile.txt
```

```
(shuvo@kali)-[~/Downloads/test]
$ wc -w test.txt
22 test.txt
```

## 18. grep

DESCRIPTION:

grep command uses a search term to look through a file.

SYNTAX:

```
grep [option]... Pattern [file]...
```

EXAMPLE:

Search the word Hello in file named myfile.txt

```
grep "Hello" myfile.txt
```

```
(shuvo@kali)-[~/Downloads/test]
$ grep "voice" demo.txt
Marketers must optimize content for voice queries.
```

## 19. free

DESCRIPTION:

Display RAM details in Linux machine.

SYNTAX:

```
free
```

EXAMPLE:

To display the RAM details in Linux machine need to write following command.

free



```
(shuvo@kali)-[~/Downloads/test]
$ free
              total        used        free      shared  buff/
cache  available
Mem:    6460816    1260928    4609088     23316      8
50504   5199888
Swap:   2715644         0    2715644
```

## 20. pipe ( | )

### DESCRIPTION:

Pipe command is used to send output of one program as a input to another. Pipes “|” help combine 2 or more commands.


### SYNTAX:

Command 1 | command 2

### EXAMPLE:

Display lines of input files containing “Aug” and send to standard output

ls -l | grep “Aug”



```
(shuvo@kali)-[~/Downloads/test]
$ grep "Voice" demo.txt | wc -w
6
```