## Department of Information Technology

**A.Y.:** 2023-24                 Sem: III                 **Sub:** Operating Systems Laboratory

**Name: Anish Sharma**                                      **Div: I-1**
**Roll no: I011**                                          **SAP ID: 60003220045**

### Experiment 1
### (Linux Commands)

**Aim**: Explore the internal commands of Linux and write shell scripts.

**Theory**:



Anish Sharma          60003220045          I011

Exp: 1                          OS

Aim: Explore the internal commands of linux and write shell scripts.

Theory:

1 Linux Operating System

Linux is an open-source Unix-like operating system - based family on the Linux kernel and the OS kernel was published by Linux Torvalds.

Bootloader          Init Program

Software libraries ← Design of Linux OS → User Interface

Video Input Infrastructure

2. Architecture of Linux System

The Linux operating system's architecture mainly contains some of components:

a) Kernel
b) System Library
c) Hardware layer
d) System
e) Shell utility
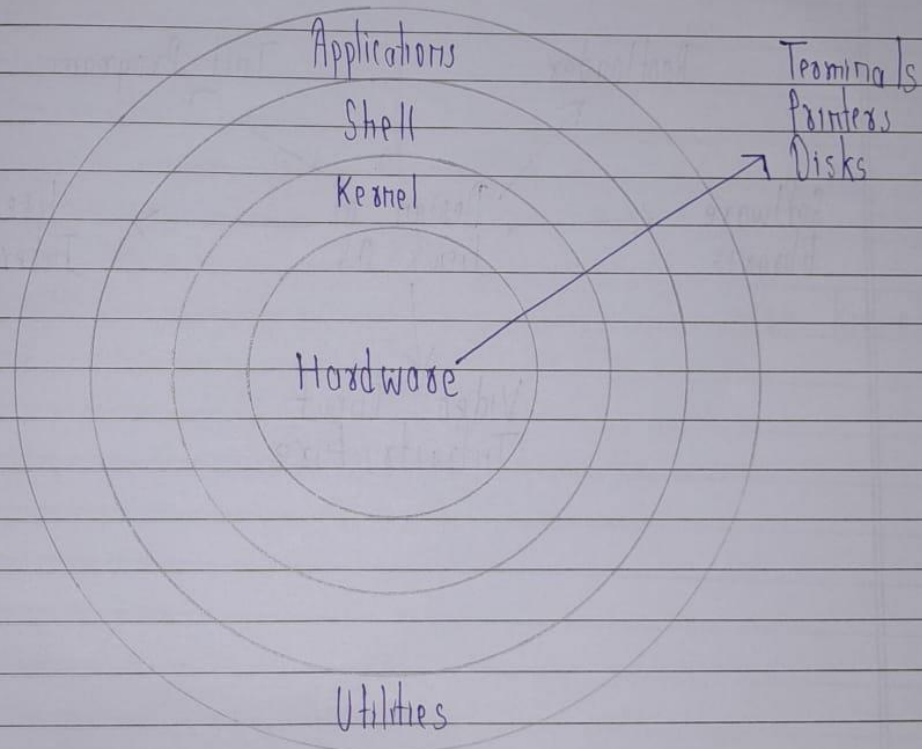
Applications
Shell
Kernel

Hardware

Terminals
Printers
Disks

Utilities

3 Kernel and its types

The kernel is one of the core section of an operating system. It is responsible for each of the major aition of the linux OS. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system.

Monolithic Kernel

Types of Kernel

Micro Kernels

Exo Kernels

Hybrid Kernels
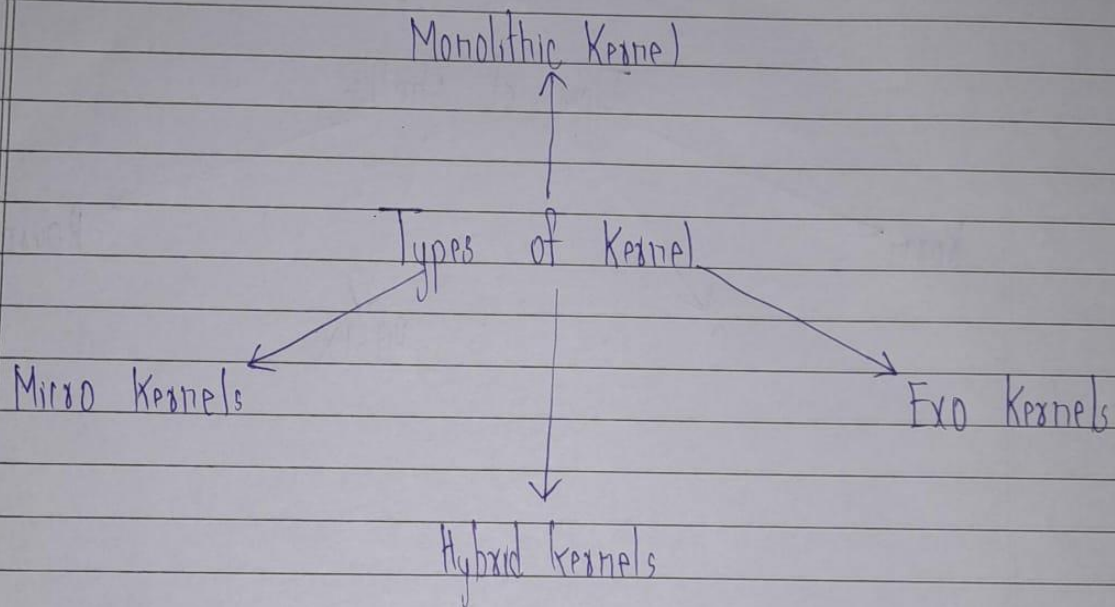
4  Shell

It is an interface among the kernel and user.
It can afford the services of kernel. It can take
commands through the user and runs the functions
of the kernel. These operating systems are
categorized into two different types
(i) graphical shells
(ii) command-line shells
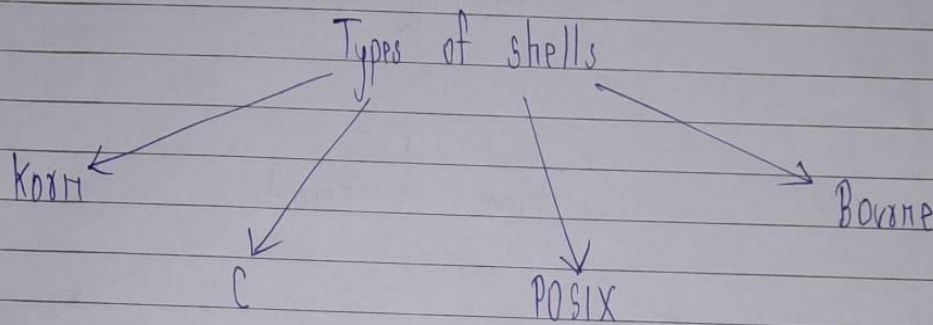
Types of shells

Korn

C

POSIX

Bourne

**Department of Information Technology**

**A.Y.:** 2023-24                    Sem: III                    **Sub:** Operating Systems Laboratory

**Write theory on following points:**

- **Linux operating system**

  Linux (/ˈlɪnʊks/ *LIN-uuks*) is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged as a Linux distribution (distro), which includes the kernel and supporting system software and libraries, many of which are provided by the GNU Project.

- **Architecture of Linux system**

The Linux operating system's architecture mainly contains some of the components: the Kernel, System Library, Hardware layer, System, and Shell utility.

1. Kernel:- The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS.

2. System Libraries:- These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

3. System Utility Programs:- It is responsible for doing specialized level and individual activities.

4. Hardware layer:- Linux operating system contains a hardware layer that consists of several peripheral devices like CPU, HDD, and RAM.

5. Shell:- It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel.

- **Kernel and its types**

The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS. This operating system contains distinct types of modules and cooperates with underlying hardware directly. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

  o Monolithic Kernel

**Department of Information Technology**

- o   Micro kernels
- o   Exo kernels
- o   Hybrid kernels

- **Shell**

 It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are the graphical shells and command-line shells.

The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.

**Procedure:**
**Execute following commands on terminal and include the description and snapshot of each command**

**Linux Commands**

**(I) File and Directory Related commands**
    **1) pwd**
        This command prints the current working directory

```
$ pwd
pwd

/home/cg/root/650b0f7f9c384
```

    **2) ls**
        This command displays the list of files in the current working directory.

        $ls –l Lists the files in the long format

## Department of Information Technology

```
$ ls -l
total 0
-rw-r--r-- 1 webmaster webmaster 0 Sep 20 20:23 file1
-rw-r--r-- 1 webmaster webmaster 0 Sep 20 20:23 file2
-rw-r--r-- 1 webmaster webmaster 0 Sep 20 20:24 file3
```

$ls –t Lists in the order of last modification time

```
$ cd folder1
$ ls -t
file3  file2  file1
```

$ls –d Lists directory instead of contents

```
$ cd folder1
cd folder1

export "PS1=$ "

$ ls -d
.
```

$ls -u Lists in order of last access time

```
$ cd folder1
cd folder1

export "PS1=$ "

$ ls -u
file3  file2  file1
```

**3) cd**

This command is used to change from the working directory to any other directory specified.

$cd directoryname

```
$ cd folder1
cd folder1


export "PS1=$ "


$ touch file1
$ touch file2
$ touch file3
```

### 4) cd ..

This command is used to come out of the current working directory.

$cd ..

```
$ cd folder2
cd folder2


export "PS1=$ "


$ touch file1
$ touch file2
$ rm file2
$ ls
file1
$ cd ..
cd ..
```

*used cd command to access the directory then cd .. command to exit it

### 5) mkdir

This command helps us to make a directory.

$mkdir directoryname

```
$ mkdir folder1
$ ls
folder1
```

### 6) rmdir

This command is used to remove a directory specified in the command line. It requires the
specified directory to be empty before removing it.

$rmdir directoryname

Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Department of Information Technology**

**A.Y.:** 2023-24                    Sem: III                    **Sub:** Operating Systems Laboratory

```
$ mkdir folder1
$ mkdir folder2
$ rmdir folder1
$ ls
folder2
```

**7) cat**

This command helps us to list the contents of a file we specify.
$cat [option][file]
cat > filename – This is used to create a new file.
cat >>filename – This is used to append the contents of the file
Eg:
cat file1
cat file1 file2 > all
cat file1 >> file2

```
$ touch file1
$ touch file2
$ touch file3
$ ls
file1  file2  file3
```

**8) cp**

This command helps us to create duplicate copies of ordinary files.
$cp source destination

**9) mv**

ss
This command is to establish an additional filename for the same ordinary file.
$ln firstname secondname

```
$ cd folder1
$ touch file1
$ ln file1 file2
$ ls
file1  file2
```

**11) rm**

This command is used to delete one or more files from the directory.
$rm [option] filename
$rm –i Asks the user if he wants to delete the file mentioned.
$rm –r Recursively delete the entire contents of the directory as well as the directory itself.

```
$ touch file1
$ touch file2
$ rm file2
$ ls
file1
```

## (II) Process and status information commands

### 1) who
This command gives the details of who all have logged in to the UNIX system currently.
$ who
*command doesn't work in the online environment

### 2) who am i
This command tells us as to when we had logged in and the system's name for the
connection being used.
$who am i
8
*command doesn't work in the online environment

### 3) date
This command displays the current date in different formats.
+%D mm/dd/yy +%w Day of the week
+%H Hr-00 to 23 +%a Abbr.Weekday
+%M Min-00 to 59 +%h Abbr.Month
+%S Sec-00 to 59 +%r Time in AM/PM
+%T HH:MM:SS +%y Last two digits of the year

```
$ date
date

Wed Sep 20 20:36:24 IST 2023
```

### 4) echo
This command will display the text typed from the keyboard.
$echo
Eg: $echo Have a nice day
O/p Have a nice day

```
$ echo hello world
echo hello world

hello world
```

**Department of Information Technology**

**A.Y.:** 2023-24                    Sem: III                    **Sub:** Operating Systems Laboratory

## (III) Text related commands

### 1. head

This command displays the initial part of the file. By default it displays first ten lines of the
file.
$head [-count] [filename]

```
$ head file1
head file1


this is file1 content
```

### 2. tail

This command displays the later part of the file. By default it displays last ten lines of the
file.
$tail [-count] [filename]

```
$ tail file1
this is file1 content
```

### 3. wc

This command is used to count the number of lines, words or characters in a file.
wc -l <filename> print the line count

```
$ echo "my name is atharva" >> file1
echo "my name is atharva" >> file1


export "PS1=$ "


$ wc -l file1
1 file1
```

wc -c <filename> print the byte count

```
$ echo "my name is atharva" >> file1
$ wc -c file1
19 file1
```

wc -m <filename> print the character count


wc -L <filename> print the length of longest line

```
$ wc -L file1
18 file1
```

wc -w <filename> print the word count

## Department of Information Technology

```
$ wc -w file1
4 file1
```

### 4. find

The find command is used to locate files in a directory and in a subdirectory.

```
$ find file2
file2
$ find file3
find: 'file3': No such file or directory
```

## (IV) File Permission commands

### 1) chmod

Changes the file/directory permission mode:

$chmod [option] mode files

**options:**

-R Descend directory arguments recursively while setting modes.

-f Suppress error messages if command fails.

**mode:**

Who u=user, g=group, o=other, a=all (default)

**Opcode**

+ means add permission

— means remove permission

= means assign permission and remove the permission of unspecified fields

**Permission** r=Read, w=write, x=Execute

Eg. $ chmod 777 file1

Gives full permission to owner, group and others

$ chmod o-w file1

Removes write permission for others.

```
$ chmod o-w file1
chmod o-w file1
```

### 2) chgrp

chgrp user file      Makes file belong to the group user.

### 3) chown

chown cliff file     Makes cliff the owner of file.

**Department of Information Technology**

chown -R cliff dir   Makes cliff the owner of dir and everything in its directory tree.

**(V) Other Useful Commands:**

      **1) exit** - Ends your work on the UNIX system.

```
$ exit
exit
exit
```

      **2) Ctrl-l or clear**

            Clears the screen.

            Before the command

```
$ mkdir folder1
$ ls
folder1
$ clear
```

            After the command

      **3) Ctrl-c**

            Stops the program currently running.

      **4) Ctrl-z**

            Pauses the currently running program.

      **5) man COMMAND**

            Looks up the UNIX command COMMAND in the online manual pages.

```
$ man COMMAND
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, including manpages, you can run the 'unminimize'
command. You will still need to ensure the 'man-db' package is installed.
```

**6) history**

            List all commands typed so far.

**Department of Information Technology**

```
$ history
history


1002  python3 main.py
1003  cd /home/cg/root/650b0b4958f71
1004  python3 main.py
1005  cd /home/cg/root/650b09c33407a
1006  python3 main.py
1007  cd /home/cg/root/650b09c33407a
1008  python3 main.py
1009  cd /home/cg/root/650b0b9ac5cac
1010  lua main.lua
1011  cd /home/cg/root/650b07a65193d
1012  tclsh main.tcl
1013  cd /home/cg/root/650b07a65193d
1014  tclsh main.tcl
1015  cd /home/cg/root/650b0aa7930d5
1016  kotlinc main.kt -include-runtime -nowarn -d main.jar && java -Xmx1024M -Xms256M -jar main.jar
1017  cd /home/cg/root/650b0aa7930d5
1018  kotlinc main.kt -include-runtime -nowarn -d main.jar && java -Xmx1024M -Xms256M -jar main.jar
```

**7) more FILE**

Display the contents of FILE, pausing after each screenful.

There are several keys which control the output once a screenful has been printed.

<enter> Will advance the output one line at a time.

<space bar> Will advance the output by another full screenful.

"q" Will quit and return you to the UNIX prompt.

```
$ more file1
this is file1 content
$ more file2
this is file2 content
```

**8) less FILE**

"less" is a program similar to "more", but which allows backward movement in the file as well as forward movement.

```
$ less file2
less file2
this is file2 content
file2 (END)·
(END)·
(END)
...skipping...
this is file2 content
~
~
```

**9) lpr FILE**

## Department of Information Technology

**A.Y.:** 2023-24                     Sem: III                     **Sub:** Operating Systems Laboratory

The lpr command submits files for printing. Files supplied at the command-line are sent to the specified printer or to the print queue if the printer is busy.

### Shell Scripting

A shell is a special user program that provides an interface for the user to use operating system services. Shell accepts human-readable commands from users and converts them into something which the kernel can understand. It is a command language interpreter that executes commands read from input devices such as keyboards or from files. The shell gets started when the user logs in or starts the terminal.

**Shell is broadly classified into two categories –**

**Command Line Shell**
**Graphical shell**

### Hello.sh

```
echo "What is your name?"
read PERSON
echo "Hello, $PERSON"
```

Here is a sample run of the script −

$./Hello.sh

### Add.sh

```
val=`expr 2 + 2`
echo "Total value : $val"
```

### Loop in shell

```
read x
read y


if [ $x -gt $y ]

then

echo X is greater than Y

elif [ $x -lt $y ]

then

echo X is less than Y

elif [ $x -eq $y ]

then

echo X is equal to Y

fi
```

**Conclusion:** I have learned basic Linux command.