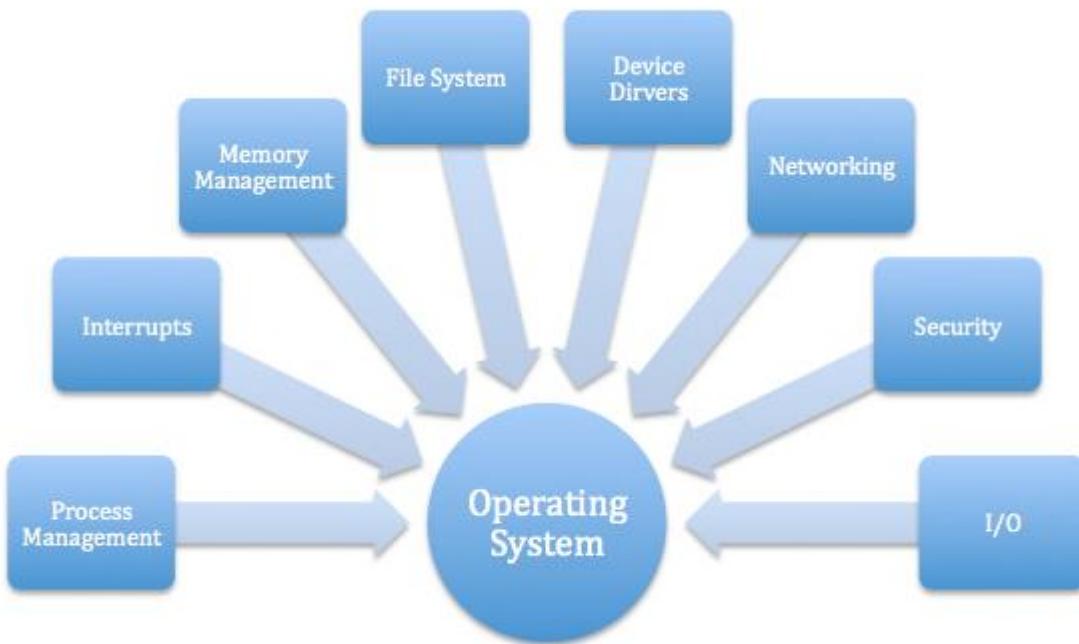


PRACTICAL - 1

AIM: Introduction to Operating System, Types of Operating System, Difference between Linux, UNIX and Windows and Installation of Ubuntu

1.1 INTRODUCTION TO OPERATING SYSTEM

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers. Some popular Operating Systems include Linux, Windows, OS X, VMS, OS/400, AIX, z/OS, etc.



1.2 TYPES OF OPERATING SYSTEM

Batch OS

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

Distributed OS

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

Networked OS

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

Time Sharing OS

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

1.3 DIFFERENCE BETWEEN LINUX, UNIX AND WINDOWS

OS → PARAMETERS			
Cost	Mostly free, but some are priced.	Paid	Mostly free, but some paid
Development and Distribution	Open Source	Owned by Microsoft.	Mostly by AT&T Labs
Manufacturer	Linux kernel is developed by the community. Linus Torvalds oversees things.	Microsoft	Solaris (Oracle), AIX (IBM) & HP-UX Hewlett Packard. OSX, an UNIX based OS.
User	Programmers and Developers	Home, Personal and General use	Internet servers and infrastructure backend support
Usage	Mainframes and supercomputers	On PC's desktops, laptops, servers and some phones.	Workstations
File system support	Ext2, Ext3, Ext4, Btrfs	FAT, FAT32, NTFS, exFAT	jfs, gpfs, hfs, hfs+, ufs, xfs, zfs format
Text mode interface	BASH (Bourne Again Shell)	Command Prompt, PowerShell	Bourne, Bash, KORN, CSH

1.4 INSTALLATION OF UBUNTU 18.04

1. Download Ubuntu

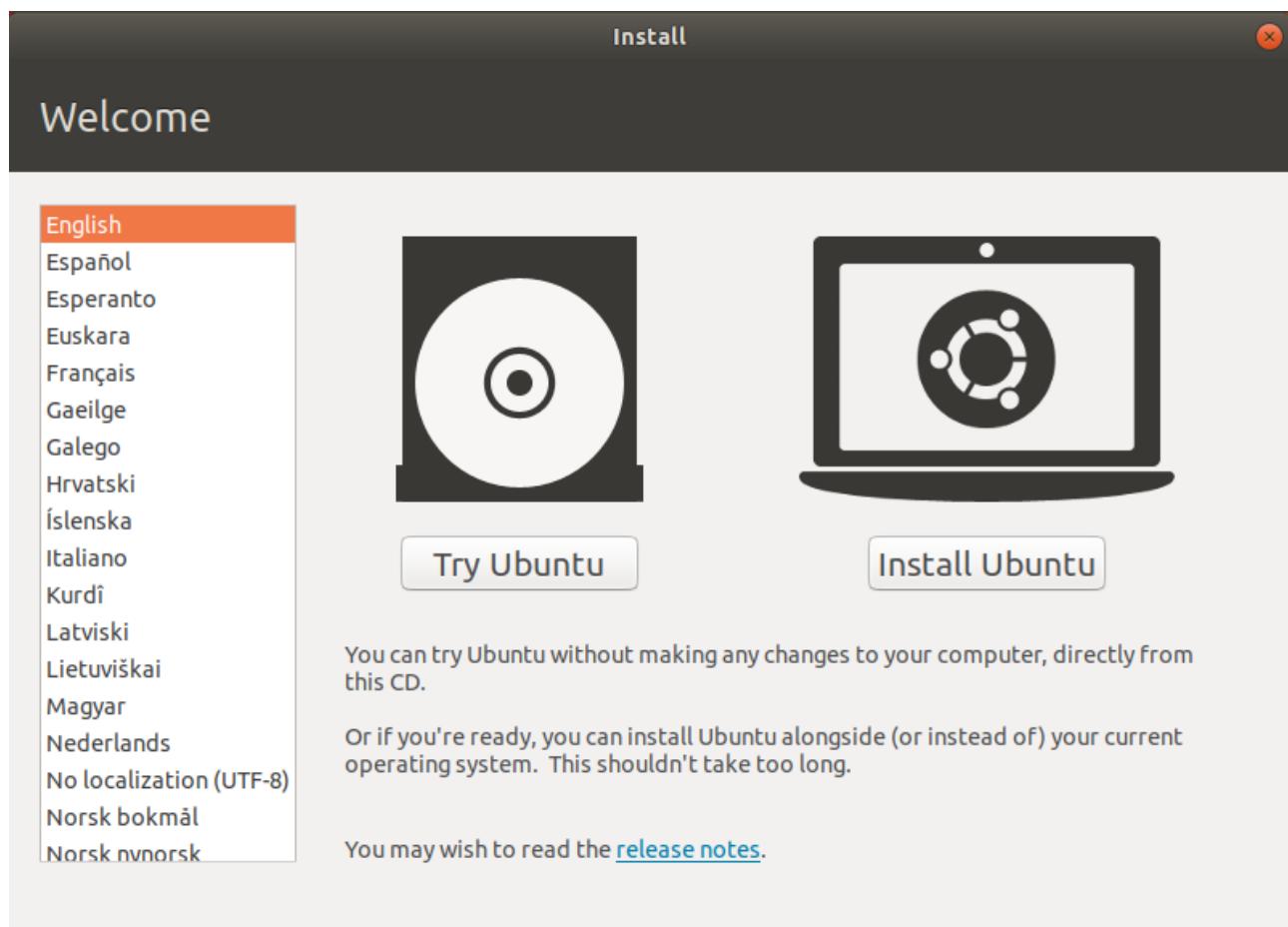
For you first need to download an Ubuntu .ISO CD image file. In this example we install Ubuntu version 18.04 LTS.

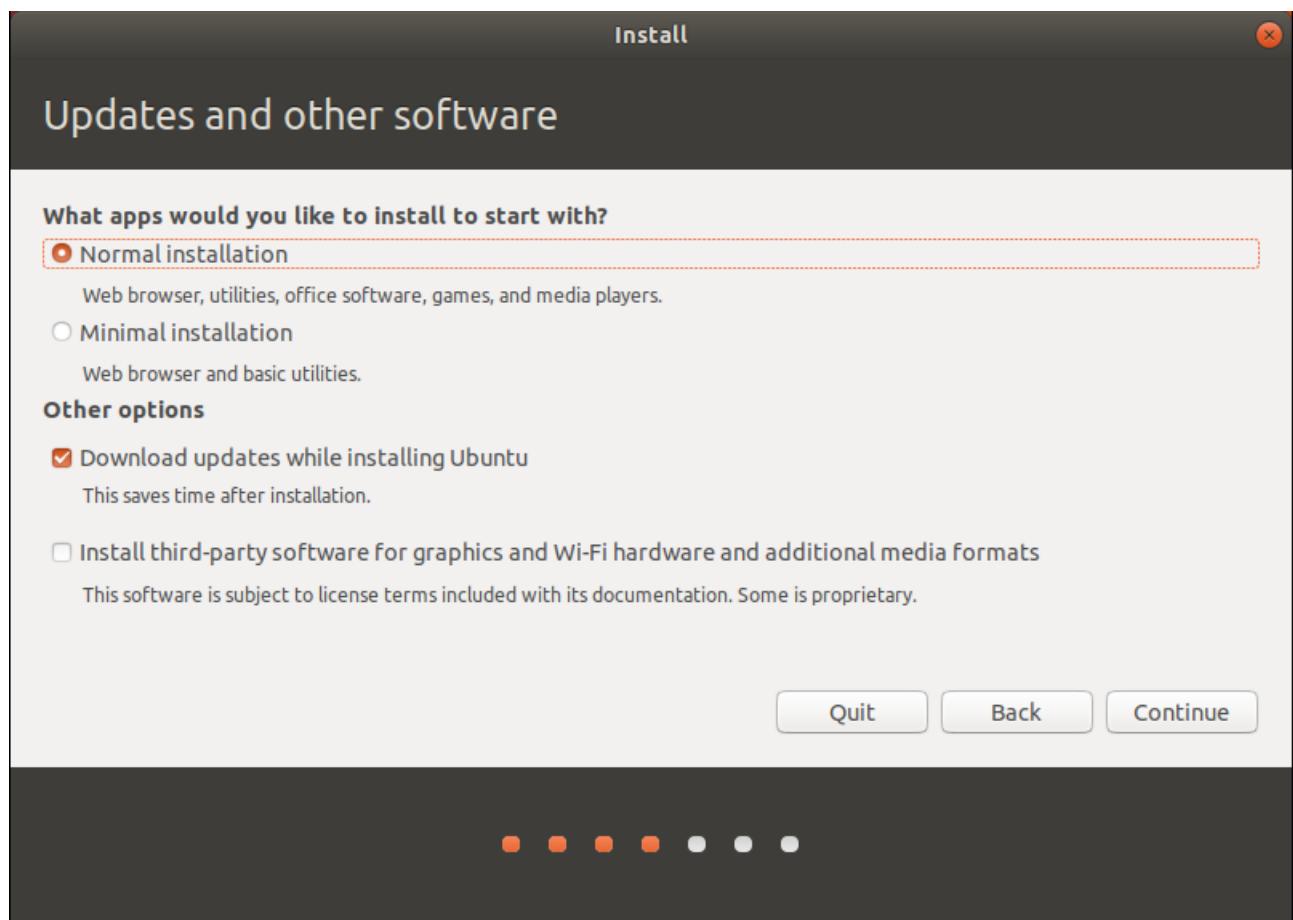
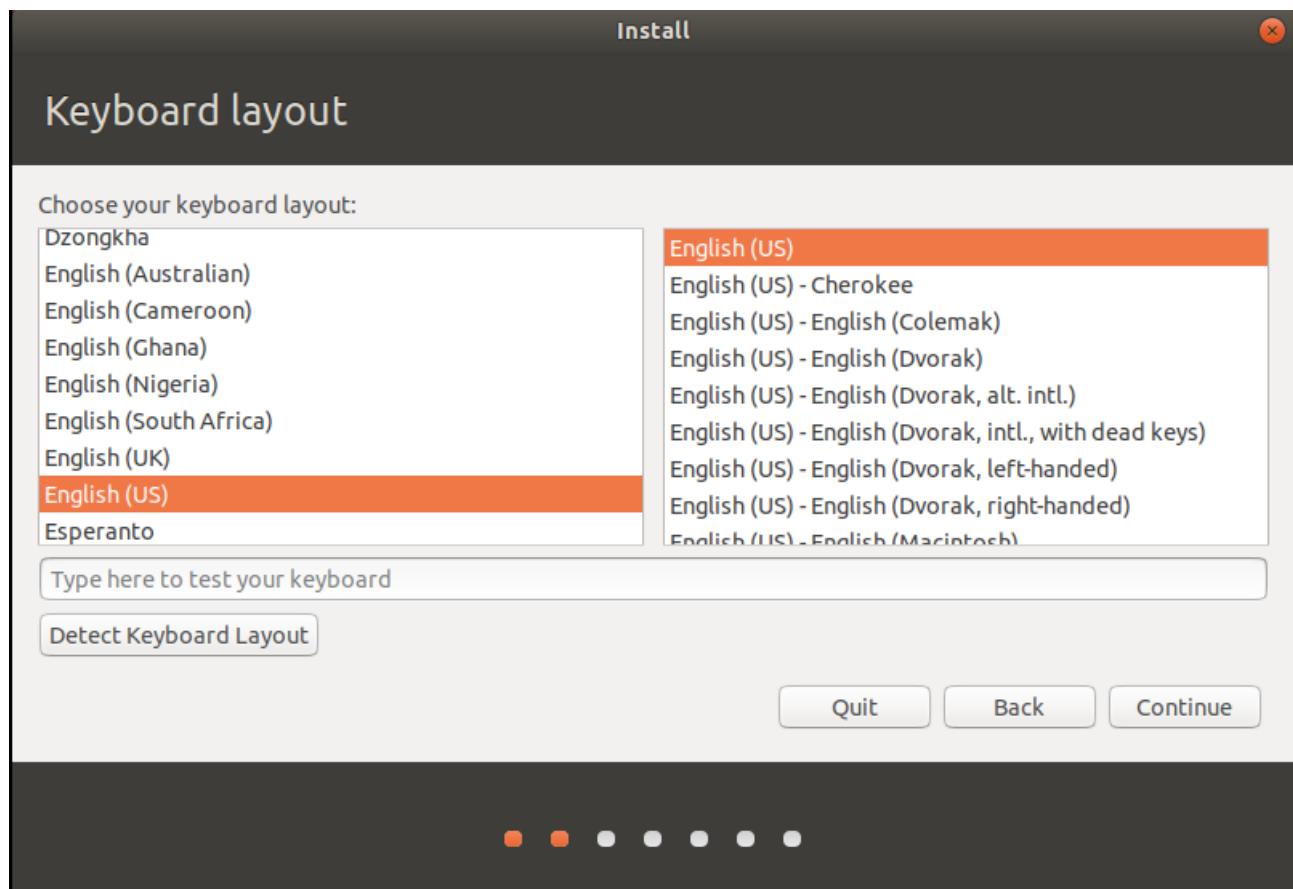
2. Create Bootable USB

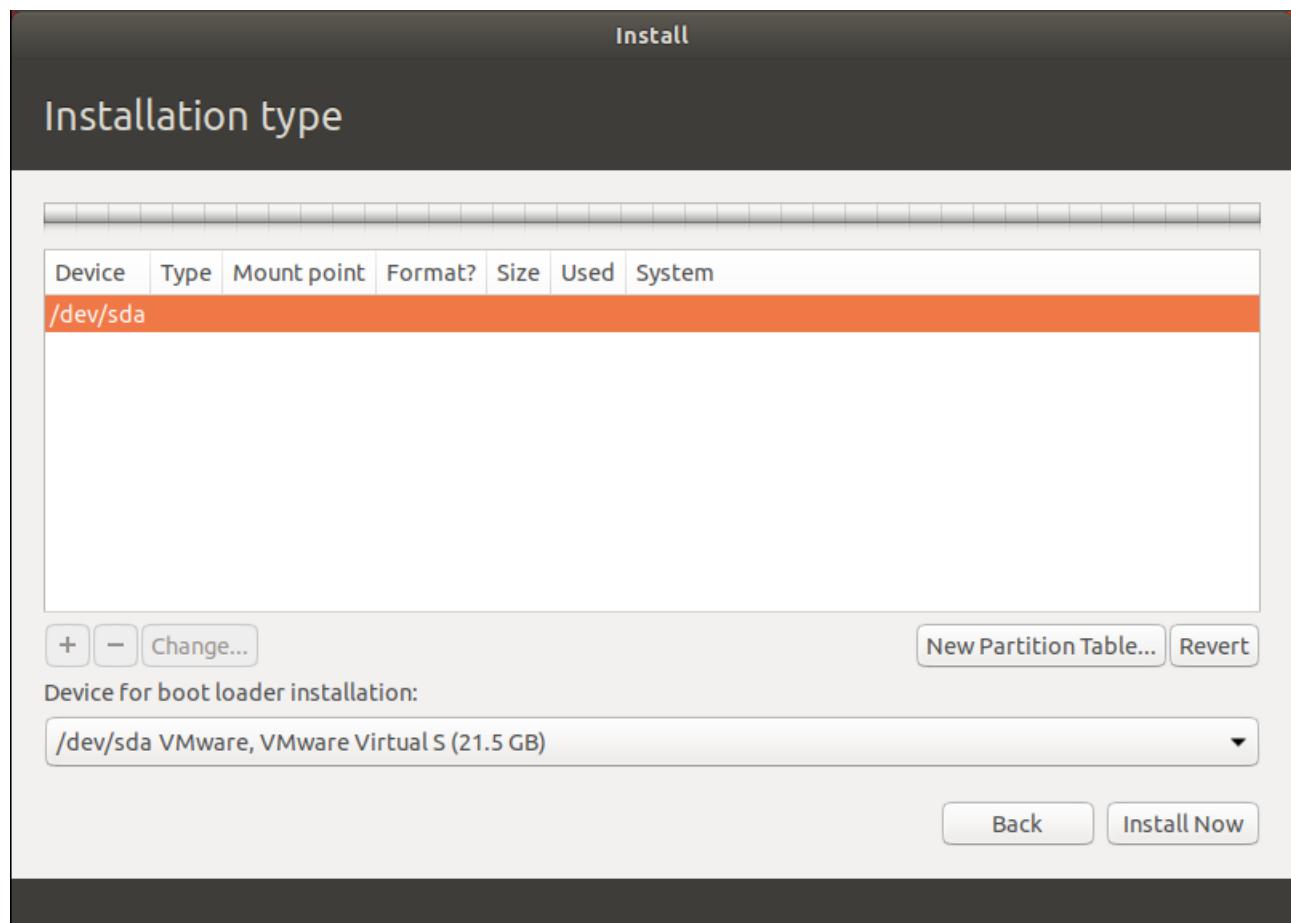
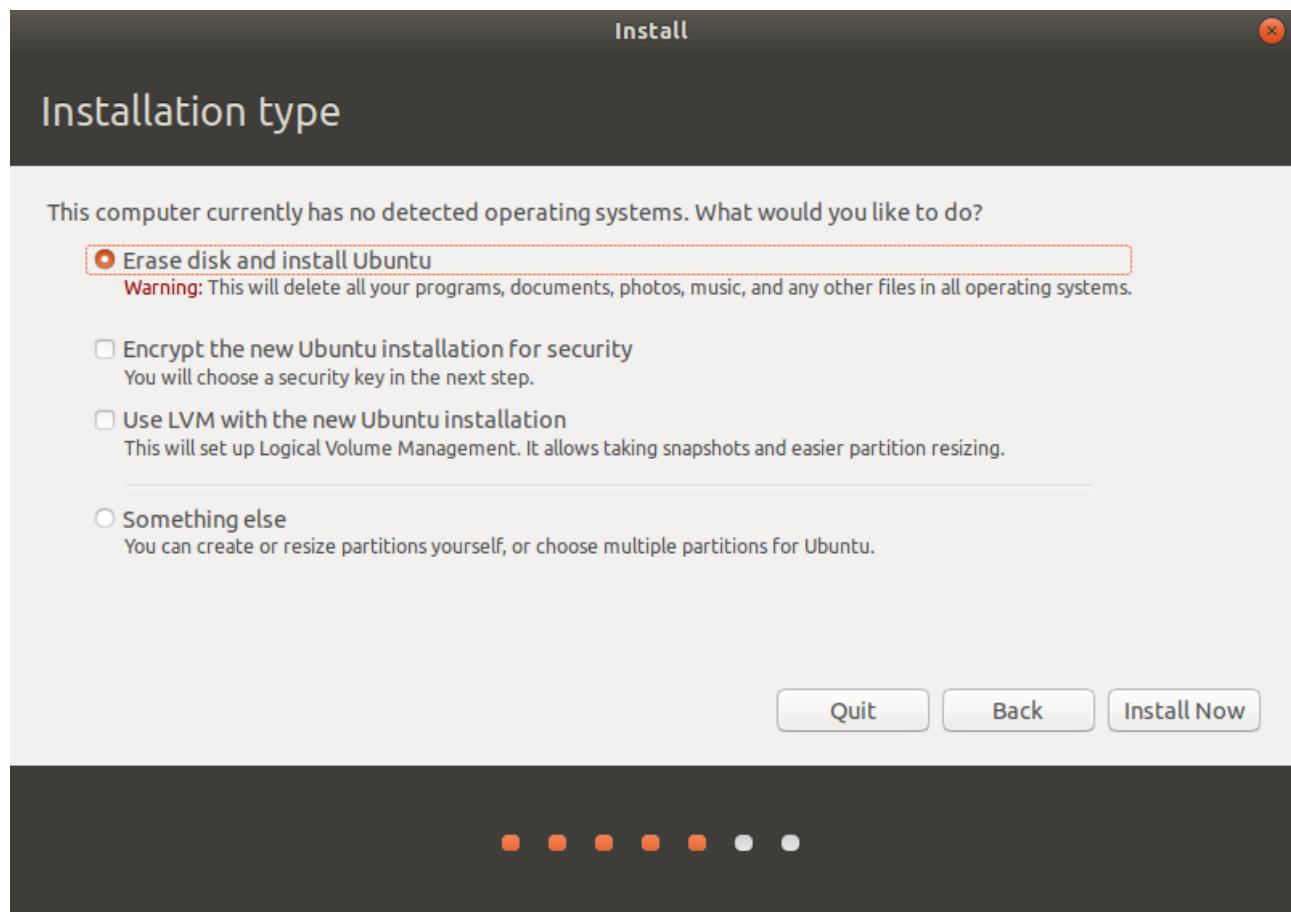
Create a bootable USB from the .ISO image file. It usually boots and installs faster than a CD image.

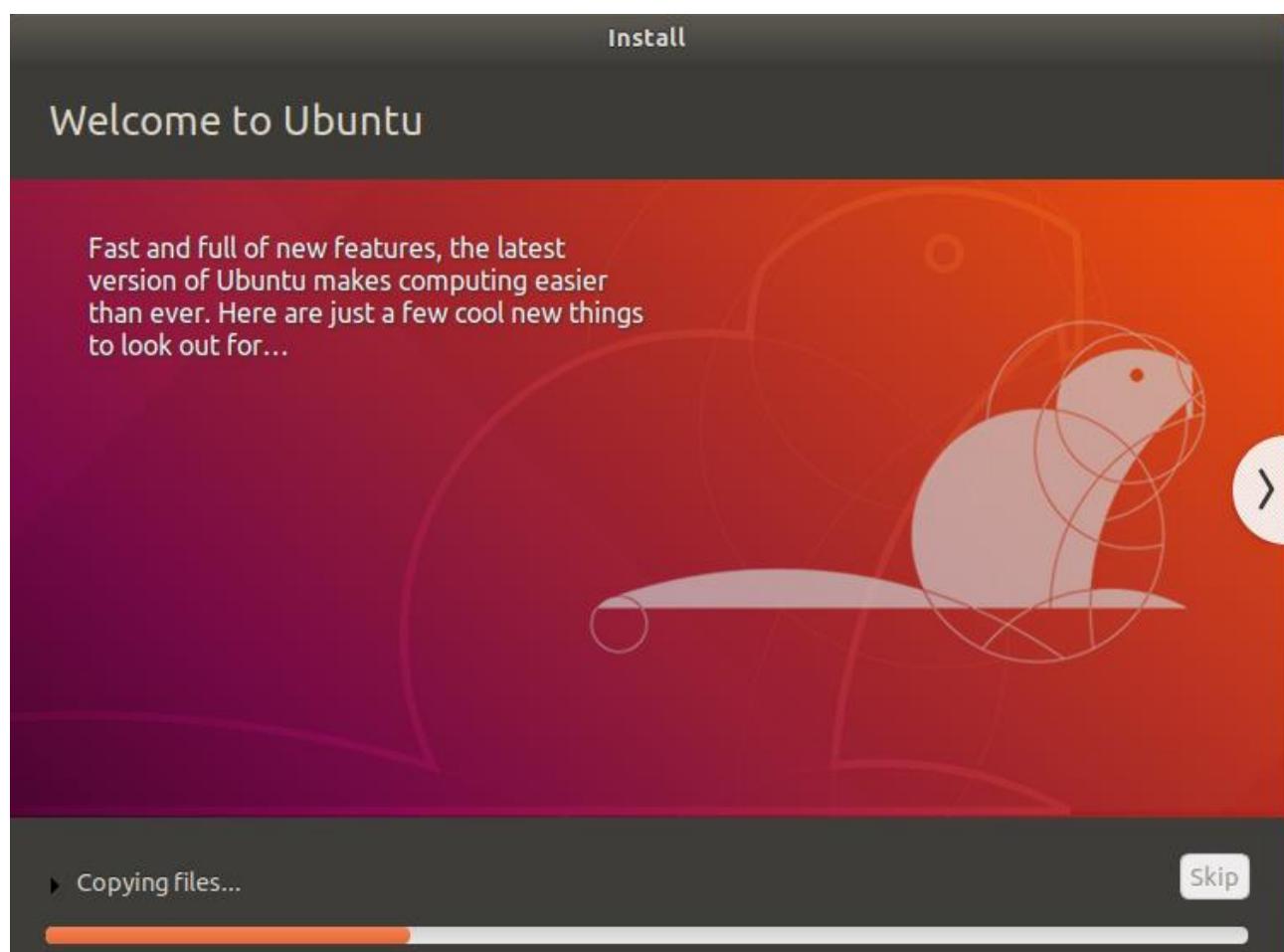
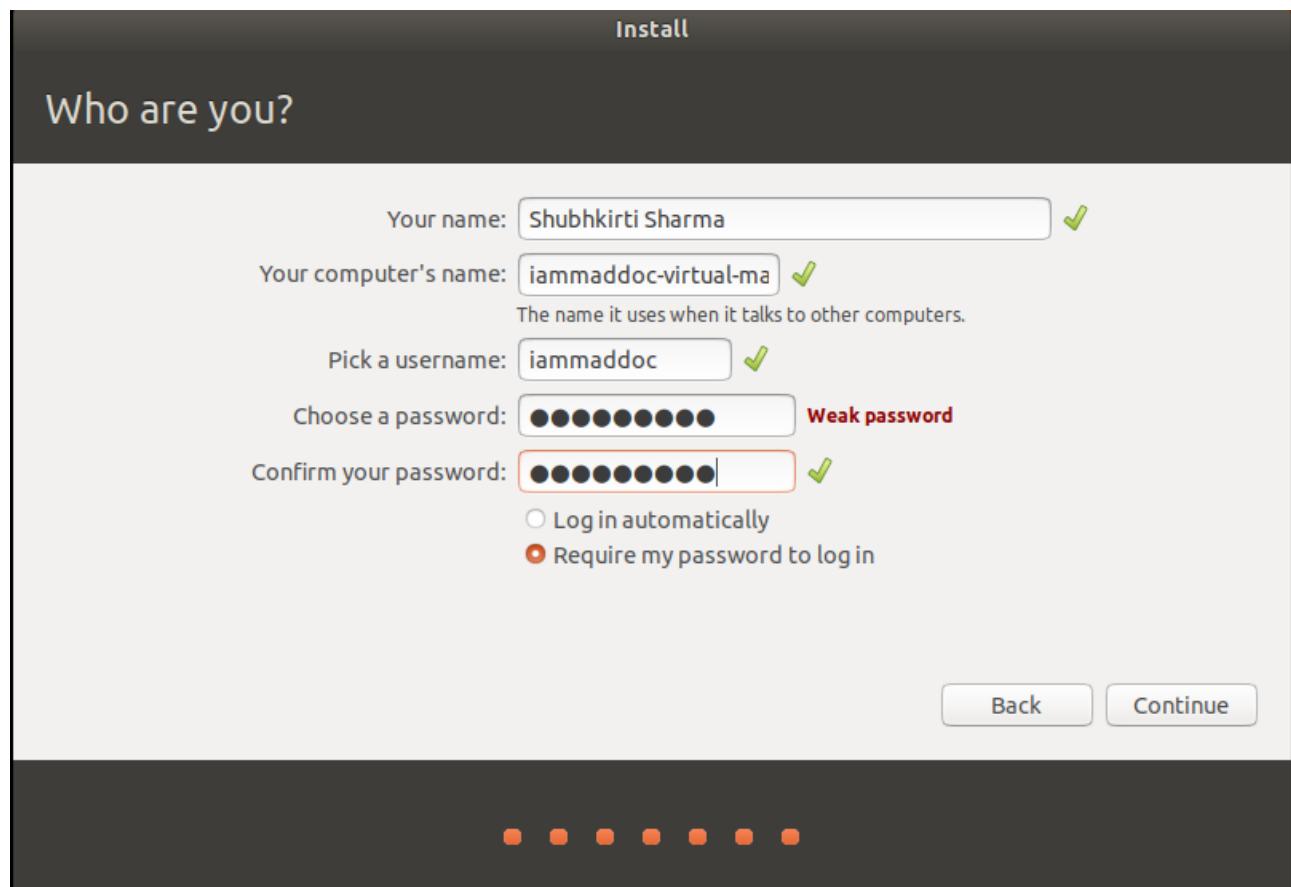
3. Install Ubuntu

Once you get the bootable USB working follow the screens below to install USB:









1.5 SHELL

In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. It is named a shell because it is the outermost layer around the operating system kernel.

Bourne Shell

The original Bourne shell is named after its developer at Bell Labs, Steve Bourne. It was the first shell used for the Unix operating system, and it has been largely surpassed in functionality by many of the more recent shells. However, all Unix and many Linux versions allow users to switch to the original Bourne Shell, known simply as "sh," if they choose to forgo features such as file name completion and command histories that later shells have added.

C Shell

The C shell, as its name might imply, was designed to allow users to write shell script programs using a syntax very similar to that of the C programming language. It is known as "csh."

TC Shell

TC shell is an expansion upon the C shell. It has all the same features, but adds the ability to use keystrokes from the Emacs word processor program to edit text on the command line. For example, users can press Esc-D to delete the rest of the highlighted word. It is also known as "tcsh."

Korn Shell

Korn Shell was also written by a developer at Bell Labs, David Korn. It attempts to merge the features of the C shell, TC shell and Bourne shell under one package. It also includes the ability for developers to create new shell commands as the need arises.

It is known as "ksh."

Bash

The popularity of sh motivated programmers to develop a shell that was compatible with it, but with several enhancements. Linux systems still offer the sh shell, but "bash" -- the "Bourne-again Shell," based on sh -- has become the new default standard. One attractive feature of bash is its ability to run sh shell scripts unchanged.

PRACTICAL - 2

AIM: To implement the following commands in linux
MAN, CAT, TOUCH, LS, SCRIPT, CD, MKDIR, RM, RMDIR, PWD

2.1 MAN

man is the system's manual viewer; it can be used to display manual pages, scroll up and down, search for occurrences of specific text, and other useful functions.

Each argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section number, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections, following a pre-defined order and to show only the first page found, even if page exists in several sections.

SYNTAX:

```
man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale]
[-m system[,...]] [-M path] [-S list] [-e extension] [-i|-I]
[--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P pager]
[-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justification]
[-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
[[section] page ...] ...
```

OPTIONS:

Tag	Description
-C file, --config-file=file	Use this user configuration file rather than the default of ~/.manpath
-d, --debug	Print debugging information.
-D, --default	This option is normally issued as the very first option and resets man's behaviour to its default.
-c, --catman	This option is not for general use and should only be used by the catman program.
-m system[,...], --systems=system[,...]	If this system has access to other operating system's manual pages, they can be accessed using this option.
-S list, -s list, --sections=list	List is a colon- or comma-separated list of 'order specific' manual sections to search.
-i, --ignore-case	Ignore case when searching for manual pages. This is the default.
-I, --match-case	Search for manual pages case-sensitively.

USING MAN WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ man cat
CAT(1)                               User Commands                         CAT(1)
NAME
    cat - concatenate files and print on the standard output
SYNOPSIS
    cat [OPTION]... [FILE]...
DESCRIPTION
    Concatenate FILE(s) to standard output.
    With no FILE, or when FILE is -, read standard input.
    -A, --show-all
```

```

equivalent to -vET
-b, --number-nonblank
    number nonempty output lines, overrides -n
.....
.....

```

2.2 CAT

Cat command concatenate FILE(s), or standard input, to standard output. With no FILE, or when FILE is -, it reads standard input.

SYNTAX:

```
cat [Options]... [File]...
```

OPTIONS:

Tag	Description
-A, --show-all	equivalent to -vET
-b, --number-nonblank	number nonblank output lines
-e	equivalent to -vE
-E, --show-ends	display \$ at end of each line
-n, --number	number all output lines
-s, --squeeze-blank	never more than one single blank line
-t	equivalent to -vT
-T, --show-tabs	display TAB characters as ^I
-u	(ignored)
-v, --show-nonprinting	use ^ and M- notation, except for LFD and TAB. display this help and exit
--help	display this help and exit
--version	output version information and exit

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat> file1
This is the first file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file1 file2 > file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file
This is the first file
Sample text in file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file1>>file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file2
Sample text in file2
This is the first file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat <file1 >>file5
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file5
Kindof file5
This is the first file
Sample text in file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file6 1>f3 2>f4
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat f3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat f4
cat: file6: No such file or directory
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat >file6
YES it is the sixth file.
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file6 1>f3 2>f4
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat f3
YES it is the sixth file.
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file1 file2 >>file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file3
cat: file5: No such file or directory
This is the first file
Second file it is

```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -A file6
```

```

YES it is the sixth file.$
Second file it is$
cat: file5: No such file or directory$
This is the first file$  

Second file it is$  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -b file4
  1 It is the fourth file.

  2 Appended by 3rd file.
  3 It has the contents of third file too.

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -e file4
It is the fourth file.$
$  

Appended by 3rd file.$
It has the contents of third file too.$  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -E file4
It is the fourth file.$
$  

Appended by 3rd file.$
It has the contents of third file too.$  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -n file4
  1 It is the fourth file.
  2
  3 Appended by 3rd file.
  4 It has the contents of third file too.

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -s file8
Empty
file  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -s file8
Empty
fil  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file7
This is the 7th file with          2 tabs.  

The end.  

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat -t file7
This is the 7th file with      tabs.

```

2.3 TOUCH

Update the access and modification times of each FILE to the current time. Basically changes the timestamp of the file.

SYNTAX:

`touch [OPTION]...FILE...`

OPTIONS:

Tag	Description
<code>-a</code>	change only the access time
<code>-c, --no-create</code>	do not create any files
<code>-d, --date=STRING</code>	parse STRING and use it instead of current time
<code>-f</code>	(ignored)
<code>-m</code>	change only the modification time
<code>r, --reference=FILE</code>	use this file's times instead of current time
<code>-t STAMP</code>	use [[CC]YY]MMDDhhmm[.ss] instead of current time
<code>--time=WORD</code>	change the specified time: WORD is access, atime, or use: equivalent to <code>-a</code> WORD is modify or mtime: equivalent to <code>-m</code>
<code>--help</code>	display this help and exit
<code>--version</code>	output version information and exit

USING TOUCH WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch file1 file2 file3
```

USING TOUCH WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -r file5 file4
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -r file6 -B 30 file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -d '1 Jan 2018 10:22' file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -d '14 Jan' file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -d '14:24' file9
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -c file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ touch -c -t 01101730 file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -l
-rw-r--r-- 1 iammaddoc 44 Jan 10 17:30 file7
```

2.4 LS

Lists information about the FILES (the current directory by default). Sort entries alphabetically by default

SYNTAX:

```
ls [OPTION]... [FILE]...
```

OPTIONS:

Option	Description
ls -a	list all files including hidden file starting with '.'
ls --color	colored list [=always/never/auto]
ls -d	list directories - with '*/'
ls -F	add one char of */=>@ to entries
ls -i	list file's inode index number
ls -l	list with long format - show permissions
ls -la	list long format including hidden files
ls -lh	list long format with readable file size
ls -ls	list with long format with file size
ls -r	list in reverse order
ls -R	list recursively directory tree
ls -s	list file size
ls -S	sort by file size
ls -t	sort by time & date
ls -X	sort by extension name

USING LS WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
```

USING LS WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -a
. Dir1 Dir3 f3 file file1 file3 file5 file7
.. Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -A
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls --author
```

```

Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -b
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -B
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -c
Dir4 Dir2 file7 file1 file3 file8 file6 f4
Dir3 Dir1 file4 file2 file0 file f3 file5
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls --color
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -d
.
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -D
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -f
file2 .. file5 f3 file0 Dir3 file6 Dir2 f4
file Dir4 file7 file8 file1 . file3 file4 Dir1

```

2.5 SCRIPT

script makes a typescript of everything printed on your terminal. It is useful for users who need a hardcopy record of an interactive session as proof of work done, as the typescript file can be printed out later with `lpr`. If the `file` argument is given, script saves all dialogue in `file`. If no file name is given, the typescript is saved in a file named typescript.

SYNTAX:

```
script [FILENAME]...
```

OPTIONS:

TAG	DESCRIPTION
<code>-a, --append</code>	Append the output to file or typescript, retaining the prior contents.
<code>-c, --command command</code>	Run the <code>command</code> rather than an interactive shell.
<code>-e, --return</code>	Return the exit code of the child process.
<code>-f, --flush</code>	Flush output after each write.
<code>--force</code>	Allow the default output destination, i.e. the typescript file, to be a hard or symbolic link.
<code>-q, --quiet</code>	Operate without displaying any output.
<code>-t, --timing[=file]</code>	Output timing data to standard error, or to <code>file</code> when given. This data contains two fields, separated by a space.
<code>-V, --version</code>	Display version information, and exit.
<code>-h, --help</code>	Display a help message, and exit.

USING SCRIPT WITHOUT OPTIONS:

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ script myfile.txt
Script started, file is myfile.txt
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4 FILE file3 file6 file.odt Practical 5.pdf
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cd Scripting/
Iammaddoc@iammaddoc-virtual-machine:~/OS/Scripting$ cd ..

```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ exit
exit
Script done, file is myfile.txt
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat myfile.txt
Script started on Friday 27 April 2018 11:38:06 AM IST
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4 FILE file3 file6 file.odt Practical 5.pdf
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cd Scripting/
Iammaddoc@iammaddoc-virtual-machine:~/OS/Scripting$ cd ..
Iammaddoc@iammaddoc-virtual-machine:~/OS$ exit
exit
```

Script done on Friday 27 April 2018 11:38:38 AM IST

USING SCRIPT WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ script -c "ps ax" /tmp/processes.txt
Script started, file is /tmp/processes.txt
  PID TTY      STAT    TIME COMMAND
   1 ?        Ss      0:02 /sbin/init splash
   2 ?        S      0:00 [kthreadd]
   4 ?        S<     0:00 [kworker/0:0H]
   6 ?        S<     0:00 [mm_percpu_wq]
   7 ?        S      0:00 [ksoftirqd/0]
.....
.....
....
```

2.6 CD

The **cd** command, which stands for "change directory". Changes current working directory to the one specified into it.

SYNTAX:

`cd [-L | -P [-e]] directory`

OPTIONS:

TAG	DESCRIPTION
-L	Force symbolic links to be followed. In other words, if you tell cd to move into a "directory", which is actually a symbolic link to a directory.
-P	Use the physical directory structure without following symbolic links. In other words, only change into the specified directory if it actually exists as named.
-e	If the -P option is specified, and the current working directory cannot be determined, this option tells cd to exit with an error.

USING CD WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ cd Desktop/
Iammaddoc@iammaddoc-virtual-machine:~/Desktop$
Iammaddoc@iammaddoc-virtual-machine:~/Desktop$ cd
Iammaddoc@iammaddoc-virtual-machine:~$
Iammaddoc@iammaddoc-virtual-machine:~/Desktop/WebTech$ cd ~
Iammaddoc@iammaddoc-virtual-machine:~$
Iammaddoc@iammaddoc-virtual-machine:~/Desktop/WebTech/project$ cd ..
Iammaddoc@iammaddoc-virtual-machine:~/Desktop/WebTech$ 
Iammaddoc@iammaddoc-virtual-machine:~$ cd /
Iammaddoc@iammaddoc-virtual-machine:/$
```

2.7 MKDIR

Create the DIRECTORY(ies), if they do not already exist.

SYNTAX:

`mkdir [OPTION]... DIRECTORY...`

OPTIONS:

TAG	DESCRIPTION
<code>-m, --mode=MODE</code>	set file mode (as in chmod), not a=rwx – umask
<code>-p, --parents</code>	no error if existing, make parent directories as needed
<code>-v, --verbose</code>	print a message for each created directory
<code>-Z, --context=CTX</code>	set the SELinux security context of each created directory to CTX
<code>--help</code>	display this help and exit
<code>--version</code>	output version information and exit
<code>--kernelargs args</code>	Adds args to the arguments appended on the kernel command line.
<code>--size size</code>	Uses size (in kilobytes) as the size of the image to use for the boot disk.

USING MKDIR WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ mkdir Dir1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir1  f4  file0  file2  file4  file6  file8
      f3  file  file1  file3  file5  file7
```

USING MKDIR WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ mkdir -p Dir2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir1  f3  file  file1  file3  file5  file7
Dir2  f4  file0  file2  file4  file6  file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ mkdir -v Dir3
mkdir: created directory 'Dir3'
Iammaddoc@iammaddoc-virtual-machine:~/OS$ mkdir --context Dir4
```

2.8 PWD

Print the full filename of the current working directory.

SYNTAX:

`pwd [OPTION]...`

OPTIONS:

TAG	DESCRIPTION
<code>-L, --logical</code>	If the environment variable PWD contains an absolute name of the current directory with no "." or ".." components, then output those contents, even if they contain symbolic links. Otherwise, fall back to default (-P) behavior.
<code>-P, --physical</code>	Print a fully resolved name for the current directory, in which all components of the name are actual directory names, and not symbolic links.
<code>--help</code>	Display a help message, and exit.
<code>--version</code>	Display version information, and exit.

USING PWD WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/Desktop/OS$ pwd
/home/iammaddoc/Desktop/OS
```

USING PWD WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/Desktop/OS $ pwd -L
/home/iammaddoc/Desktop/OS
Iammaddoc@iammaddoc-virtual-machine:~/OS$ pwd -P
/home/iammaddoc/Desktop/OS
```

2.9 RMDIR

Remove the DIRECTORY(ies), if they are empty.

SYNTAX:

```
rmdir [OPTION]... DIRECTORY...
```

OPTIONS:

Tag	Description
-p	Each directory argument is treated as a pathname of which all components will be removed.
-v, --verbose	Display verbose information for every directory processed.
--ignore-fail-on-non-empty	Do not report a failure which occurs solely because a directory is non-empty.
--help	Display a help message, and exit.
--version	Output version information, and exit.

USING RMDIR WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir1 Dir3 f3 file file1 file3 file5 file7
Dir2 Dir4 f4 file0 file2 file4 file6 file8
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ rmdir Dir1
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Dir3 f3 file file1 file3 file5 file7
```

USING RMDIR WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ rmdir --ignore-fail-on-non-empty Dir2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir2 Dir4 f4 file0 file2 file4 file6 file8
Dir3 f3 file file1 file3 file5 file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ rmdir -p Dir2/hi
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir3 f3 file file1 file3 file5 file7
Dir4 f4 file0 file2 file4 file6 file8
Iammaddoc@iammaddoc-virtual-machine:~/OS$ rmdir -v Dir3
rmdir: removing directory, 'Dir3'
```

2.10 RM

rm removes each file specified on the command line. By default, it does not remove directories; see removing directories below for details.

SYNTAX:

```
rm [OPTION...] FILE...
```

OPTIONS:

Tag	Description
-f, --force	Ignore nonexistent files, and never prompt before removing.
-i	Prompt before every removal.
-I	Prompt once before removing more than three files, or when removing recursively.
--no-preserve-root	Do not treat "/" (the root directory) in any special way.
--preserve-root	Do not remove "/" (the root directory), which is the default behavior.
-r, -R, --recursive	Remove directories and their contents recursively.
-d, --dir	Remove empty directories.
-v, --verbose	Verbose mode; explain at all times what is being done.

USING RM WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ rm file1
Iammaddoc@iammaddoc-virtual-machine:~$ rm file1 file2 file3
```

USING RM WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ rm -i file4
rm: remove regular file 'file4'? Y
Iammaddoc@iammaddoc-virtual-machine:~$ rm -r mydir
```

PRACTICAL - 3

AIM: To implement the following commands in linux:
ECHO, WHO, WHOAMI, CP, MV, TTY, WC, TR, GREP, TEE

3.1 ECHO

Displays message on screen, writes each given STRING to standard output, with a space between each word. It is frequently used in scripts, batch files, and as part of individual commands; anywhere you may need to output text.

SYNTAX:

```
echo [SHORT-OPTION]... [STRING]...
echo LONG-OPTION
```

OPTIONS:

Tag	Description
-n	Do not output a trailing newline.
-e	Enable interpretation of backslash escape sequences (see below for a list of these).
-E	Disable interpretation of backslash escape sequences (this is the default).
--help	Display a help message and exit.
--version	Output version information and exit.
\\\	A literal backslash character ("\").
\a	An alert (The BELL character).
\b	Backspace
\c	Produce no further output after this.
\e	The escape character; equivalent to pressing the escape key.
\f	A form feed.
\n	A newline.
\r	A carriage return
\t	A horizontal tab.
\v	A vertical tab.
\0NNN	byte with octal value NNN (which can be 1 to 3 digits).
\xHH	byte with hexadecimal value HH (which can be either 1 or 2 digits)

USING ECHO WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ echo 'hey'
hey
Iammaddoc@iammaddoc-virtual-machine:~$ echo "HELLO"
HELLO
Iammaddoc@iammaddoc-virtual-machine:~$ echo `Have a good day`
No command 'Have' found, did you mean:
  Command 'save' from package 'atfs' (universe)
  Command 'dave' from package 'libhttp-dav-perl' (universe)
Have: command not found
```

DIFFERENCE BETWEEN " and ``

```
Iammaddoc@iammaddoc-virtual-machine:~$ echo 'echo "Facts are based on statistics"' '
echo "Facts are based on statistics"
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ echo `echo "Facts are based on statistics"`
Facts are based on statistic
```

USING ECHO WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ echo -e 'Remove \bThe \bSpaces'
RemoveTheSpaces
Iammaddoc@iammaddoc-virtual-machine:~$ echo -e 'New line \nLine 2'
New line
Line 2
Iammaddoc@iammaddoc-virtual-machine:~$ echo -n 'Have a good day'
Have a good dayIammaddoc@iammaddoc-virtual-machine:~$ 
Iammaddoc@iammaddoc-virtual-machine:~$ echo -E 'No new line \n'
No new line \n
Iammaddoc@iammaddoc-virtual-machine:~$ echo 'No new line \n'
No new line \n
Iammaddoc@iammaddoc-virtual-machine:~$ echo -Ee 'Good morning.\nHave a great day ahead.'
Good morning.
Have a great day ahead.
```

3.2 WHO

Print information about users who are currently logged in.

SYNTAX:

```
who [OPTION]... [ FILE | ARG1 ARG2 ]
```

OPTIONS:

-a, --all	Same as using the options -b -d --login -p -r -t -T -u.
-b, --boot	Display the time of the last system boot.
-d, --dead	Display dead processes.
-H, --heading	Print a line of column headings.
--ips	Print IP addresses instead of hostnames. With --lookup, canonicalizes based on stored IP, if available, rather than stored hostname.
-l, --login	Print system login processes.
--lookup	Attempt to canonicalize hostnames via DNS.
-m	Only print information about the user and host associated with standard input (the terminal where the command was issued). This method adheres to the POSIX standard.
-p, --process	Print active processes spawned by init.
-q, --count	Displays all login names, and a count of all logged-on users.
-r, --runlevel	Print the current runlevel.
-s, --short	Print only name, line, and time fields, which is the default.
-t, --time	Print the last time the system clock was changed, if the information is available.
-T, -w, --mesg	Add a character which indicates the state of the terminal line: "+" if the terminal is writable, "-" if it is not, or "?" if a bad line is encountered.
-u, --users	Print the idle time for each user, and the process ID.
--message	Same as -T.
--writable	Same as -T.

USING WHO WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who
Iammaddoc  tty7          2018-02-18 08:53 (:0)
```

USING WHO WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -a
    system boot 2018-02-18 04:41
    run-level 5 2018-02-18 04:41
LOGIN    tty1          2018-02-18 04:41           1018 id=tty1
Iammaddoc + tty7      2018-02-18 08:53 07:34           11012 (:0)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -b
    system boot 2018-02-18 04:41
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -d
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -H
NAME     LINE          TIME          COMMENT
Iammaddoc  tty7        2018-02-18 08:53 (:0)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who --ips
Iammaddoc  tty7        2018-02-18 08:53 (:0)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -l
LOGIN    tty1          2018-02-18 04:41           1018 id=tty1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who --lookup
Iammaddoc  tty7        2018-02-18 08:53 (:0)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -m
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -p
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who -q
Iammaddoc
# users=1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ who --writable
Iammaddoc + tty7      2018-02-18 08:53 (:0)
```

3.3 WHOAMI

Prints the user name associated with the current effective user ID.

SYNTAX:

`whoami [OPTION]...`

OPTIONS:

Tag	Description
--help	Display a help message, and exit.
--version	Display version information, and exit.

USING WHOAMI WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whoami
Iammaddoc
```

3.4 CP

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

SYNTAX:

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

OPTIONS:

Tag	Description
--backup[=control]	Make a backup of each existing destination file that would otherwise be overwritten or removed.
-b	Like --backup, but does not accept a control argument; the default control method is always used.
-i, --interactive	Prompt before overwrite (overrides a previous -n option).
--strip-trailing-slashes	Remove any trailing slashes from each source argument.
-s, --symbolic-link	Make symbolic links instead of copying the files themselves. All source files must be absolute pathnames starting with a slash, unless the destination files are in the current directory.
-S, --suffix=suffix	Override the usual backup suffix.
-t, --target-directory=directory	Copy all source arguments into directory
--help	Display a help message, and exit.
--version	Output version information, and exit.

USING CP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4 f4  file0  file2  file4  file6  file8
f3  file  file1  file3  file5  file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp file0 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls Dir4/
file0
```

USING CP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -a file0 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4 f4  file0  file2  file4  file6  file8
f3  file  file1  file3  file5  file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls Dir4/
file0
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp --backup file1 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls Dir
file0  file1  file1~
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp --copy-contents file1 file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp --copy-contents file2 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls Dir4/
file0  file1  file1~  file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -f file1 file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -i file1 Dir4/
cp: overwrite 'Dir4/file1'? y
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -H file1 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -l file1 Dir4/
cp: cannot create hard link 'Dir4/file1' to 'file1': File exists
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -lf file1 Dir4/
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cp -v file4 Dir4/
'file4' -> 'Dir4/file4'
```

3.5 MV

The mv command moves, or renames, files and directories on your file-system.

SYNTAX:

```
mv [options] [-T] source destination
```

OPTIONS:

Tag	Description
--backup[=vcm]	Make a backup of each existing destination file, using the version control method vcm.
-b	Like --backup, but does not accept a backup method.
-f, --force	Always overwrite existing files without prompting.
-i, --interactive	Prompt before overwriting an existing file, regardless of the file's permissions.
-n, --no-clobber	Never overwrite any existing file.
--strip-trailing-slashes	Remove any trailing slashes from each source argument.
-u, --update	Don't overwrite files if they're newer. A move will only happen if the destination file is older than the source file, or the destination file does not already exist.
-v, --verbose	Provide verbose output. Print the name of every file moved.
--help	Display a help message, and exit.
--version	Display version information, and exit.

USING MV WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ mv myfile.txt myfiles
Iammaddoc@iammaddoc-virtual-machine:~$ mv myfile1 myfile2 myfiles
```

USING MV WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ mv -t myfiles myfile1 myfile2
Iammaddoc@iammaddoc-virtual-machine:~$ mv -n file file2
Iammaddoc@iammaddoc-virtual-machine:~$ mv -f file file2
Iammaddoc@iammaddoc-virtual-machine:~$ mv -i file file2
mv: overwrite 'file2'? 
```

3.6 TTY

Print the file name of the terminal connected to standard input.

SYNTAX:

```
tty [OPTION]...
```

OPTIONS:

Tag	Description
-s, --silent, --quiet	Print nothing, only return an exit status.
--help	display this help and exit.
--version	output version information and exit.

USING TTY WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ tty
/dev/pts/18
```

USING TTY WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ tty -s
Iammaddoc@iammaddoc-virtual-machine:~/OS$
```

3.7 WC

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. A word is a non-zero-length sequence of characters delimited by white space.

SYNTAX:

```
wc [OPTION]... [FILE]...
```

OPTIONS:

-c, --bytes	print the byte counts.
-m, --chars	print the character counts.
-l, --lines	print the newline counts.
--files0-from=F	read input from the files specified by NUL-terminated names in file F
-L, --max-line-length	print the length of the longest line.
-w, --words	print the word counts.
--help	display a help message, and exit.
--version	output version information, and exit.

USING WC WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc
COUNT the words of this statement
      1       6      34
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc file6
      5     26  123 file6
```

USING WC WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc -c file6
123 file6
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc -m file6
123 file6
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc -l file6
5 file6
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc -L file6
37 file6
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wc -w file6
26 file6
```

3.8 TR

It is used to translate (replaces character by character) or delete characters.

SYNTAX:

```
tr [-Ccsu] string1 string2
```

OPTIONS:

Tag	Description
-c	Same as -C but complement the set of values in <i>string1</i> .
-C	Complement the set of characters in <i>string1</i> , that is "-C ab" includes every character except for 'a' and 'b'.
-d	Delete characters in <i>string1</i> from the input.
-s	Squeeze multiple occurrences of the characters listed in the last operand (either <i>string1</i> or <i>string2</i>) in the input into a single instance of the character. This occurs after all deletion and translation is completed.
-u	Guarantee that any output is unbuffered.

USING TR WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ tr 'abc' 'pqr'
echo abc
echo pqr
```

USING TR WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ tr -t 'there' 'hi'
there
hiere
Iammaddoc@iammaddoc-virtual-machine:~$ tr 'there' 'hi'
there
hiiii
Iammaddoc@iammaddoc-virtual-machine:~$ tr -d .
echo hi. this is demo.
echo hi this is demo
Iammaddoc@iammaddoc-virtual-machine:~$ tr -s ,
this is sample text,,,,,, with multiple commas,,,,,, reduced to one.
this is sample text, with multiple commas, reduced to one.
```

3.9 GREP

grep searches the named input FILEs for lines containing a match to the given PATTERN. If no files are specified, or if the file “-” is given, grep searches standard input. By default, grep prints the matching lines.

SYNTAX:

```
grep [OPTIONS] PATTERN [FILE...]
grep [OPTIONS] [-e PATTERN]... [-f FILE]... [FILE...]
```

OPTIONS:

Tag	Description
-E, --extended-regexp	Interpret PATTERN as an extended regular expression (see Basic vs. Extended Regular Expressions).
-F, --fixed-strings	Interpret PATTERN as a list of fixed strings, separated by newlines, that is to be matched.
-G, --basic-regexp	Interpret PATTERN as a basic regular expression (see Basic vs. Extended Regular Expressions). This is the default option when running grep.
-P, --perl-regexp	Interpret PATTERN as a Perl regular expression. This functionality is still experimental, and may produce warning messages.

USING GREP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ grep 'a*'  
This is a sample text to find a.  
This is a sample text to find a.  
Iammaddoc@iammaddoc-virtual-machine:~$ grep 'a_b*'  
Aib aoba aoca abba  
done  
Iammaddoc@iammaddoc-virtual-machine:~$ grep 'a_b'  
aib  
Iammaddoc@iammaddoc-virtual-machine:~$ grep 'ab*'  
ba* ab* abc abcd  
ba* ab* abc abcd
```

USING GREP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ grep -i 'ab*'  
ABcd abcd aBcd  
ABcd abcd aBcd  
Iammaddoc@iammaddoc-virtual-machine:~$ grep -o 'a*'  
apple apple air plane  
a  
a  
a  
a  
Iammaddoc@iammaddoc-virtual-machine:~$ grep -c 'a*'  
apple air plane  
1  
Iammaddoc@iammaddoc-virtual-machine:~$ grep -v 'a*'  
apple air plane
```

3.10 TEE

read from standard input and write to standard output and files.

The tee command is named after the T-splitter in plumbing, which splits water into two directions and is shaped like an uppercase T.

SYNTAX:

```
tee [OPTION]... [FILE]...
```

OPTIONS:

TAG	DESCRIPTION
-a, --append	Append to the given FILEs. Do not overwrite.
-i, --ignore-interrupts	Ignore interrupt signals.

USING TEE WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ ls -1 *.txt | wc -l | tee count.txt  
2
```

USING TEE WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ wc -l *.txt|tee -a count.txt  
1 count.txt  
108560 Points.txt  
0 testing.txt  
108561 total
```

PRACTICAL - 4

AIM: To implement the following commands in LINUX:
 CMP, COMM, DIFF, DF, DU, FREE, WHATIS, WHEREIS, FIND, TYPE

4.1 CMP

Compare two files byte by byte. The optional SKIP1 and SKIP2 specify the number of bytes to skip at the beginning of each file (zero by default).

SYNTAX:

```
cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]
```

OPTIONS:

Tag	Description
-b, --print-bytes	Print differing bytes.
-i, --ignore-initial=SKIP	Skip first SKIP bytes of both files.
-i, --ignore-initial=SKIP1:SKIP2	Skip first SKIP1 bytes of FILE1 and first SKIP2 bytes of FILE2.
-l, --verbose	Output byte numbers and differing byte values.
-n, --bytes=LIMIT	Compare at most LIMIT bytes.
-s, --quiet, --silent	Suppress all normal output.
--help	Display a help message and exit.
-v, --version	Output version information and exit.

USING CMP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp file1 file2
file1 file2 differ: byte 32, line 2
```

USING CMP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp -b file1 file2
file1 file2 differ: byte 32, line 2 is 146 f 163 s
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp -i file1 file2
cmp: invalid --ignore-initial value 'file1'
cmp: Try 'cmp --help' for more information.
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp --ignore-initial=32 file1 file2
file1 file2 differ: byte 1, line
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp -l file1 file2
32 146 163
33 151 145
34 162 143
35 163 157
36 164 156
37 40 144
38 146 40
39 151 146
40 154 151
41 145 154
42 12 145
cmp: EOF on file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp --bytes=64 file1 file2
file1 file2 differ: byte 32, line 2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp --bytes=16 file1 file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp -l --bytes=16 file1 file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cmp -s file1 file2
```

4.2 COMM

Compare sorted files FILE1 and FILE2 line-by-line. With no options, comm produces three-column output. Column one contains lines unique to FILE1, column two contains lines unique to FILE2, and column three contains lines common to both files. Each of these columns can be suppressed individually with options.

SYNTAX:

```
comm [OPTION]... FILE1 FILE2
```

OPTIONS:

TAG	DESCRIPTION
-1	suppress column 1 (lines unique to FILE1)
-2	suppress column 2 (lines unique to FILE2)
-3	suppress column 3 (lines that appear in both files)
--check-order	check that the input is correctly sorted, even if all input lines are pairable
--nocheck-order	do not check that the input is correctly sorted
--output-delimiter=STR	separate columns with string STR
--zero-terminated	line delimiter is NUL, not newline
--version	output version information, and exit.

USING COMM WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat NewContacts
Simran
Asthya
Ajay
Anurag
Virat
Daneem
Srijita
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat > OldContacts
Asthya
Simran
Ajay
Virat
Radhika
Juneet
Yati
^C
Iammaddoc@iammaddoc-virtual-machine:~/OS$ comm NewContacts OldContacts
      Astha
      Simran
comm: file 1 is not in sorted order
comm: file 2 is not in sorted order
      Ajay
      Astha
      Anurag
      Virat
      Daneem
      Radhika
      Juneet
      Srijita
      Yati
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort NewContacts  NewContacts
Ajay
27
```

Anurag
Astha
Daneem
Simran
Srijita
Virat

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ sort OldContacts OldContacts

Ajay
Astha
Juneet
Radhika
Simran
Virat
Yati

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ comm OldContacts NewContacts

Ajay	
Anurag	Astha
	Daneem
Juneet	
Radhika	Simran
	Srijita
	Virat
Yati	

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ comm OldContacts NewContacts > Contacts.csv

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ cat Contacts.csv	
Anurag	Astha
	Daneem
Juneet	
Radhika	Simran
	Srijita
	Virat
Yati	

USING COMM WITH OPTIONS:

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ comm --check-order NewContacts OldContacts

Astha	
	Simran
comm: file 1 is not in sorted order	

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ comm --zero-terminated NewContacts OldContacts

Ajay	
Anurag	
Astha	
Daneem	
Simran	
Srijita	
Virat	
	Ajay

Astha	
Juneet	
Radhika	
Simran	
Virat	
Yati	

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ comm -12 OldContacts NewContacts

Ajay	
Astha	
Simran	
Virat	

4.3 DIFF

Compare files line by line.

SYNTAX:

```
diff [OPTION]... FILES
```

OPTIONS:

TAG	DESCRIPTION
--normal	output a normal diff (the default)
-s, --report-identical-files	report when two files are the same
-c, -C NUM, --context[=NUM]	output NUM (default 3) lines of copied context
-W, --width=NUM	output at most NUM (default 130) print columns
-y, --side-by-side	output in two columns
-t, --expand-tabs	expand tabs to spaces in output
-T, --initial-tab	make tabs line up by prepending a tab
-l, --paginate	pass output through 'pr' to paginate it
-r, --recursive	recursively compare any subdirectories found
N, --new-file	treat absent files as empty
--ignore-file-name-case	ignore case when comparing file names
--no-ignore-file-name-case	consider case when comparing file names
-x, --exclude=PAT	exclude files that match PAT
-X, --exclude-from=FILE	exclude files that match any pattern in FILE

USING DIFF WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff file1 file2
1c1
< This is the first file
---
> Second
file it is
```

USING DIFF WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat >file1
It is a text file.
This is the first file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat >file2
It is a text file.
This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -q file1 file2
Files file1 and file2 differ
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -s file1 file2
2c2
< This is the first file
---
> This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -c file1 file2
*** file1      2018-02-19 00:12:45.203092554 +0530
```

```

--- file2      2018-02-19 00:13:07.279092249 +0530
*****
*** 1,2 ****
It is a text file.
! This is the first file
--- 1,2 ----
It is a text file.
! This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -u file1 file2
--- file1      2018-02-19 00:12:45.203092554 +0530
+++ file2      2018-02-19 00:13:07.279092249 +0530
@@ -1,2 +1,2 @@
It is a text file.
-This is the first file
+This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -e file1 file2
2c
This is the second file
.
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -n file1 file2
d2 1
a2 1
This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -y file1 file2
It is a text file.          It is a text file.
This is the first file      | This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff --width=1 file1 file2
2c2
< This is the first file
---
> This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff --left-column file1 file2
2c2
< This is the first file
---
> This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff --suppress-common-lines file1 file2
2c2
< This is the first file
---
> This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -p file1 file2
*** file1      2018-02-19 00:12:45.203092554 +0530
--- file2      2018-02-19 00:13:07.279092249 +0530
*****
*** 1,2 ****
It is a text file.
! This is the first file
--- 1,2 ----
It is a text file.
! This is the second file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ diff -f file1 file2
c2
This is the second file
.

```

DIFFERENCE BETWEEN CMP, DIFF AND COMM

The cmp command:

- Byte by byte comparison performed for two files comparison and displays the first mismatch byte.
- cmp returns the 1st byte and the line no of the fileone to make the changes to make the fileone identical to filetwo.
- Directory names cannot be used.

The diff Command:

- The diff command provides you with the line numbers of one of the files that need to be changed to make the files identical. The output looks like:
10c10
- Indicates the changes that are to be done to make the files identical.
- Returns the text of filetwo that is different from filetwo.
- Directory names can be used

The comm Command:

- If you have alphabetically sorted files, or files that you can use the "sort" command on, you can find the commonalities and differences using the comm command. The output is comprised of 3 columns, the first column containing the lines unique to the first file, the second column being the lines common to both files, the third column being the lines unique to the second file.
- The comm command also has options that allow you to selectively print each column by itself. By placing the minus sign (-) and the column number after the command, you will "drop" that column. So, if you wanted only the lines that are the same for each file, you would issue the command: comm -13.

4.4 DF

df displays the amount of disk space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

SYNTAX:

```
df [OPTION]... [FILE]...
```

OPTIONS:

TAG	DESCRIPTION
-a, --all	Include dummy file systems.
-B, --block-size=SIZE	Scale sizes by SIZE before printing them. E.g., '-BM' prints sizes in units of 1,048,576 bytes. See "SIZE Format" below for more information.
--total	Display a grand total.
-h, --human-readable	Print sizes in human readable format (e.g., 1K 234M 2G).
-H, --si	Same as -h, but use powers of 1000 instead of 1024.
-i, --inodes	List inode information instead of block usage.
-k	Like --block-size=1K.
-l, --local	Limit listing to local file systems.
--no-sync	Do not invoke a sync before getting usage info, which is the default setting.
-P, --portability	Use the POSIX output format.
--sync	Invoke a sync before getting usage info.
-t, --type=TYPE	Limit listing to file systems of type TYPE.
-T, --print-type	Print file system type.
-x, --exclude-type=TYPE	Limit listing to file systems not of type TYPE.

USING DF WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df
Filesystem 1K-blocks Used Available Use% Mounted on
udev 6071116 0 6071116 0% /dev
tmpfs 1219472 101096 1118376 9% /run
/dev/sda2 202875784 22367028 170180184 12% /
tmpfs 6097356 25312 6072044 1% /dev/shm
tmpfs 5120 4 5116 1% /run/lock
tmpfs 6097356 0 6097356 0% /sys/fs/cgroup
/dev/sda1 523248 30000 493248 6% /boot/efi
tmpfs 1219472 48 1219424 1% /run/user/1000
```

USING DF WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df -a
Filesystem 1K-blocks Used Available Use% Mounted on
sysfs 0 0 0 - /sys
proc 0 0 0 - /proc
udev 6071116 0 6071116 0% /dev
devpts 0 0 0 - /dev/pts
tmpfs 1219472 101096 1118376 9% /run
...
...
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df --block-size=1
Filesystem 1B-blocks Used Available Use% Mounted on
udev 6216822784 0 6216822784 0% /dev
tmpfs 1248739328 103518208 1145221120 9% /run
/dev/sda2 207744802816 22903885824 174264459264 12% /
tmpfs 6243692544 18194432 6225498112 1% /dev/shm
tmpfs 5242880 4096 5238784 1% /run/lock
tmpfs 6243692544 0 6243692544 0% /sys/fs/cgroup
/dev/sda1 535805952 30720000 505085952 6% /boot/efi
tmpfs 1248739328 49152 1248690176 1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df -h
Filesystem Size Used Avail Use% Mounted on
udev 5.8G 0 5.8G 0% /dev
tmpfs 1.2G 99M 1.1G 9% /run
/dev/sda2 194G 22G 163G 12% /
tmpfs 5.9G 19M 5.8G 1% /dev/shm
tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs 5.9G 0 5.9G 0% /sys/fs/cgroup
/dev/sda1 511M 30M 482M 6% /boot/efi
tmpfs 1.2G 48K 1.2G 1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df -H
Filesystem Size Used Avail Use% Mounted on
udev 6.3G 0 6.3G 0% /dev
tmpfs 1.3G 104M 1.2G 9% /run
/dev/sda2 208G 23G 175G 12% /
tmpfs 6.3G 28M 6.3G 1% /dev/shm
tmpfs 5.3M 4.1k 5.3M 1% /run/lock
tmpfs 6.3G 0 6.3G 0% /sys/fs/cgroup
/dev/sda1 536M 31M 506M 6% /boot/efi
tmpfs 1.3G 50k 1.3G 1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df -i
Filesystem Inodes IUsed IFree IUse% Mounted on
udev 1517779 539 1517240 1% /dev
tmpfs 1524339 803 1523536 1% /run
/dev/sda2 12894208 265392 12628816 3% /
tmpfs 1524339 63 1524276 1% /dev/shm
tmpfs 1524339 5 1524334 1% /run/lock
tmpfs 1524339 17 1524322 1% /sys/fs/cgroup
/dev/sda1 0 0 0 - /boot/efi
tmpfs 1524339 27 1524312 1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df -k
Filesystem 1K-blocks Used Available Use% Mounted on
udev 6071116 0 6071116 0% /dev
tmpfs 1219472 101096 1118376 9% /run
/dev/sda2 202875784 22353572 170193640 12% /
```

```

tmpfs      6097356  20444  6076912  1% /dev/shm
tmpfs      5120     4      5116   1% /run/lock
tmpfs      6097356  0      6097356  0% /sys/fs/cgroup
/dev/sda1  523248  30000  493248  6% /boot/efi
tmpfs      1219472  48    1219424  1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df --sync
Filesystem 1K-blocks Used Available Use% Mounted on
udev        6071116  0    6071116  0% /dev
tmpfs       1219472 101096 1118376  9% /run
/dev/sda2  202875784 22363568 170183644 12% /
tmpfs       6097356 24884  6072472  1% /dev/shm
tmpfs       5120     4      5116   1% /run/lock
tmpfs       6097356  0      6097356  0% /sys/fs/cgroup
/dev/sda1  523248  30000  493248  6% /boot/efi
tmpfs       1219472  48    1219424  1% /run/user/1000
Iammaddoc@iammaddoc-virtual-machine:~/OS$ df --total
Filesystem 1K-blocks Used Available Use% Mounted on
udev        6071116  0    6071116  0% /dev
tmpfs       1219472 101096 1118376  9% /run
/dev/sda2  202875784 22364624 170182588 12% /
tmpfs       6097356 25328  6072028  1% /dev/shm
tmpfs       5120     4      5116   1% /run/lock
tmpfs       6097356  0      6097356  0% /sys/fs/cgroup
/dev/sda1  523248  30000  493248  6% /boot/efi
tmpfs       1219472  48    1219424  1% /run/user/1000
total      224108924 22521100 191259252 11% -

```

4.5 DU

Summarize disk usage of the set of FILES, recursively for directories.

SYNTAX:

```

du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F

```

OPTIONS:

Tag	Description
-a, --all	Write counts for all files, not just directories.
--apparent-size	Print apparent sizes, rather than disk usage; although the apparent size is usually smaller, it may be larger due to holes in ('sparse') files, internal fragmentation, and indirect blocks.
-B, --block-size=SIZE	Scale sizes by SIZE before printing them. For example, '-BM' prints sizes in units of 1,048,576 bytes. (See SIZE format below).
-b, --bytes	Equivalent to '--apparent-size --block-size=1'.
-c, --total	Display a grand total.
-D, --dereference-args	Dereference only symlinks that are listed on the command line.
--files0-from=F	Summarize disk usage of the NUL-terminated file names specified in file F; If F is "-" then read names from standard input.
-H	Equivalent to --dereference-args (-D).
-h, --human-readable	Print sizes in human readable format, rounding values and using abbreviations. For example, "1K", "234M", "2G", etc.
--si	Like -h, but use powers of 1000, not 1024.
-k	Like --block-size=1K.
-l, --count-links	Count sizes many times if hard-linked.
-m	Like --block-size=1M.

USING DU WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du
4      ./Dir4
44     .

USING DU WITH OPTIONS:
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du --apparent-size
4      ./Dir4
9     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du --block-size=1
4096   ./Dir4
45056  .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -a
4      ./file2
0      ./file
4      ./Dir4
4      ./file5
4      ./file7
4      ./f3
4      ./file8
0      ./file0
4      ./file1
4      ./file6
4      ./file3
4      ./file4
0      ./f4
44     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -c
4      ./Dir4
44     .
44     total
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -D
4      ./Dir4
44     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -h
4.0K   ./Dir4
44K    .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -k
4      ./Dir4
44     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -L
4      ./Dir4
44     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -l
4      ./Dir4
44     .
Iammaddoc@iammaddoc-virtual-machine:~/OS$ du -s
44     .
```

4.6 FREE

Displays the total amount of free and used **physical** and **swap** **memory** in the **system**, as well as the **buffers** used by the **kernel**.

SYNTAX:

```
free [options]
```

OPTIONS:

TAG	DESCRIPTION
-b, --bytes	Display the amount of memory in bytes.
-k, --kilo	Display the amount of memory in kilobytes. This is the default.
-m, --mega	Display the amount of memory in megabytes.
-g, --giga	Display the amount of memory in gigabytes.
--tera	Display the amount of memory in terabytes.
-h, --human	Show all output fields automatically scaled to shortest three digit unit and display the units of print out. Following units are used. B = bytes, K = kilos, M = mega, G = giga, T = tera
-c, --count <i>count</i>	Display the result <i>count</i> times. Requires the -s option.
-l, --lohi	Show detailed low and high memory statistics.
-o, --old	Display the output in old format; the only difference being this option will disable the display of the "buffer adjusted" line.
-s, --seconds <i>seconds</i>	Continuously display the result <i>delay</i> seconds apart. You may actually specify any floating point number for delay. The usleep system call is used for microsecond resolution delay times.
--si	Use power of 1000 not 1024.
-t, --total	Display a line showing the column totals.

USING FREE WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ free
              total        used         free      shared  buff/cache   available
Mem:       12194708     1840436     3365880        678264    6988392    9477488
Swap:      12480508          0     12480508
```

USING FREE WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ free -t
              total        used         free      shared  buff/cache   available
Mem:       12194708     1871556     3213552        686424    7109600    9443348
Swap:      12480508          0     12480508
Total:     24675216     1871556     15694060
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ free -l
              total        used         free      shared  buff/cache   available
Mem:       12194708     1872044     3222080        669728    7100584    9458096
Low:      12194708     8972628     3222080
High:      0           0           0
Swap:      12480508          0     12480508
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ free -b
              total        used         free      shared  buff/cache   available
Mem:       12487380992  1918726144  3365384192  684810240  7203270656  9683062784
Swap:      12780040192          0  12780040192
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ free -k
              total        used         free      shared  buff/cache   available
Mem:       12194708     1873864     3283636        668836    7037208    9455944
Swap:      12480508          0     12480508
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ free -g
              total        used         free      shared  buff/cache   available
Mem:        11           1           3           0           6           9
Swap:      11           0           11-
```

DIFFERENCE BETWEEN FREE AND DF

df: displays the amount of disk space available on the file system containing each file name argument. If no file name is given, the space available on all currently mounted file systems is shown.

Example:

```
Iammaddoc@iammaddoc-virtual-machine:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/loop0        18761008  15246876  2554440  86% /
none                  4        0        4   0% /sys/fs/cgroup
udev                 493812       4   493808  1% /dev
tmpfs                 100672     1364   99308  2% /run
none                  5120       0    5120  0% /run/lock
none                 503352     1764   501588  1% /run/shm
none                 102400       20   102380  1% /run/user
/dev/sda3        174766076 164417964 10348112 95% /host
```

Free: Display RAM details in Linux machine

Example:

```
Iammaddoc@iammaddoc-virtual-machine:~$ free
total used free shared buffers cached
Mem: 8027952 4377300 3650652 0 103648 1630364
-/+ buffers/cache: 2643288 5384664
Swap: 15624188 608948 15015240
```

Where

Line 1: Indicates Memory details like total available RAM, used RAM, Shared RAM, RAM used for buffers, RAM used of caching content.

Line 2: Indicates total buffers/Cache used and free.

Line 3: Indicates total swap memory available, used swap and free swap memory size available.

So in short, df is for disk and free is for RAM.

4.7 WHATIS

Each manual page has a short description available within it. Whatis searches the manual page names and displays the manual page descriptions of any name matched.

SYNTAX:

```
whatis [-dlv?V] [-rl-w] [-s list] [-m system[,...]] [-M path] [-L locale] [-C file] name ...
```

OPTIONS:

Tag	Description
-d, --debug	Print debugging information.
-v, --verbose	Print verbose warning messages.
-r, --regex	Interpret each <i>name</i> as a regular expression.
-w, --wildcard	Interpret each name as a pattern containing shell style wildcards.
-l, --long	Do not trim output to the terminal width.
-s <i>list</i> , --sections <i>list</i> , --section <i>list</i>	Search only the given manual sections. The <i>list</i> is a colon- or comma-separated list of sections.

USING WHATIS WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis cat
cat (1)           - concatenate files and print on the standard output

Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis man
man (1)          - an interface to the on-line reference manuals
man (7)          - macros to format man pages
```

USING WHATIS WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis -d cat
From the config file /etc/manpath.config:
```

```
Mandatory mandir '/usr/man'.
Mandatory mandir '/usr/share/man'.
...
...
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis -w cat
cat (1)           - concatenate files and print on the standard output
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis -l cat
cat (1)          - concatenate files and print on the standard output
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whatis -C file6 cat
whatis: can't parse directory list 'YES it is the sixth file.
'

whatis: can't make sense of the manpath configuration file /etc/manpath.config
```

4.8 WHEREIS

whereis locates the binary, source and manual files for the specified command names. The supplied names are first stripped of leading path- name components and any (single) trailing extension of the form .ext (for example: .c). Prefixes of s. resulting from use of source code control are also dealt with. whereis then attempts to locate the desired program in the standard Linux places, and in the places specified by \$PATH and \$MANPATH.

SYNTAX:

```
whereis [options] [-BMS directory... -f] name...
```

OPTIONS:

Tag	Description
-b	Search only for binaries.
-m	Search only for manual sections.
-s	Search only for sources.
-u	Search for unusual entries. A file is said to be unusual if it does not have one entry of each requested type. For example, "whereis -m -u *" asks for those files in the current directory which have no documentation.
-B	Change or otherwise limit the places where whereis searches for binaries.
-M	Change or otherwise limit the places where whereis searches for manual sections.
-S	Change or otherwise limit the places where whereis searches for sources.
-f	Terminate the last directory list and signals the start of file names, and must be used when any of the -B, -M, or -S options are used.

USING WHEREIS WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis file
file: /usr/bin/file /usr/share/file /usr/share/man/man1/file.1.gz
```

USING WHEREIS WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -b file
file: /usr/bin/file /usr/share/file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -B file
whereis: option -f is missing
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -m file
file: /usr/share/man/man1/file.1.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -s file
file:
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -f file
file: /usr/bin/file /usr/share/file /usr/share/man/man1/file.1.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -u file
file: /usr/bin/file /usr/share/file /usr/share/man/man1/file.1.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ whereis -l file
bin: /usr/bin
bin: /usr/sbin
bin: /usr/lib
bin: /bin
bin: /sbin
bin: /etc
...
...
```

4.9 FIND

This manual page documents the GNU version of find. GNU find searches the directory tree rooted at each given starting-point by evaluating the given expression from left to right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point find moves on to the next file name. If no starting-point is specified, `.' is assumed.

SYNTAX:

```
find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]
```

OPTIONS:

TAG	DESCRIPTION
-P	Never follow symbolic links. This is the default behavior.
-L	Follow symbolic links. When find examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself (unless it is a broken symbolic link or find is unable to examine the file to which the link points).
-H	Do not follow symbolic links, except while processing the command line arguments. When find examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself.

USING FIND WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ find file
```

USING FIND WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ find -P file
file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ find -L file
file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ find -H file
file
```

4.10 TYPE

Display information about command type.

SYNTAX:

```
type [-afptP] name [name ...]
```

OPTIONS:

Tag	Description
-t	It tells a command or name is an alias, shell reserved word, function, builtin, or disk file(external command)
-p	Command of the disk file that would be executed
-P	Force a PATH search for each command/NAME, even if it is an alias,builtin, or function, and returns the name of the disk file that would be executed.
-a	Displays all locations have command or name.
-f	Suppress shell function lookup.

USING TYPE WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ type cat
cat is /bin/cat
```

USING TYPE WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ type -t ls
alias
Iammaddoc@iammaddoc-virtual-machine:~$ type -t date
file
Iammaddoc@iammaddoc-virtual-machine:~$ type -t declare
builtin
Iammaddoc@iammaddoc-virtual-machine:~$ type -t rvm
function
Iammaddoc@iammaddoc-virtual-machine:~$ type -p date
/bin/date
Iammaddoc@iammaddoc-virtual-machine:~$ type -P date
/bin/date
Iammaddoc@iammaddoc-virtual-machine:~$ echo $PATH
/usr/local/rvm/gems/ruby-2.1.0/bin:/usr/local/rvm/gems/ruby-
2.1.0@global/bin:/usr/local/rvm/rubies/ruby-
2.1.0/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/rvm/bin
Iammaddoc@iammaddoc-virtual-machine:~$ type -a echo
echo is a shell builtin
echo is /usr/sbin/echo
echo is /bin/echo
```

PRACTICAL - 5

AIM: To implement the following commands in LINUX:

SLEEP, SHUTDOWN, SEMICOLON (;), PIPE (|), AMPERSAND (&), SORT, HEAD, TAIL, MORE, LESS, BANNER, USERMOD

5.1 SLEEP

Pause for NUMBER seconds. Given two or more arguments, pause for the amount of time specified by the sum of their values.

SYNTAX:

```
sleep NUMBER[SUFFIX]...
sleep OPTION
```

OPTIONS:

Tag	Description
--help	Display a help message, and exit.
--version	Display version information, and exit.

USING SLEEP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ sleep 5
Iammaddoc@iammaddoc-virtual-machine:~$ sleep 2m
```

5.2 SHUTDOWN

shutdown may be used to halt, power-off or reboot the machine

SYNTAX:

```
shutdown [OPTIONS...]
```

OPTIONS:

Tags	Description
-a	Control access to the shutdown command using the control access file /etc/shutdown.allow.
-k	Do not shut down, but send the warning messages as if it were real.
-r	Reboot after shutdown.
-h	Instructs the system to shut down and then halt.
-f	Skip fsck after reboot.
-F	Force fsck after reboot.
-n	Don't call init to do the shutdown of processes; instruct shutdown to do that itself.
-c	Cancel a pending shutdown.
-t sec	Tell init to wait sec seconds between sending processes the warning and the kill signal, before changing to another runlevel.
Message	A message to be sent to all users, along with the standard shutdown notification.

USING SHUTDOWN WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown 8:00
Iammaddoc@iammaddoc-virtual-machine:~$ 20:00
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown +15 "Upgrading hardware, downtime should be minimal"
//schedule the system to shut down in 15 min and alerts the user with the message in " ".
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown now
```

USING SHUTDOWN WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown -r now
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown -P now
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown -H now
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown -f now
Iammaddoc@iammaddoc-virtual-machine:~$ shutdown -c
```

5.3 SEMICOLON (;

The semi-colon operator makes it possible to run, several commands in a single go and the execution of command occurs sequentially.

SYNTAX:

[command] ; [command]

USING SEMICOLON (;

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat NewContacts; mkdir new; cp NewContacts new/
Ajay
Anurag
Astha
Daneem
Simran
Srijita
Virat
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cd new/
Iammaddoc@iammaddoc-virtual-machine:~/OS/new$ ls
NewContacts
```

5.4 AMPERSAND (&)

The function of '&' is to make the command run in background. Just type the command followed with a white space and '&'. You can execute more than one command in the background, in a single go.

SYNTAX:

[command] &

USING AMPERSAND (&)

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat NewContacts & mkdir new & cp NewContacts new/
[1] 29339
[2] 29340
Ajay
Anurag
Astha
Daneem
Simran
Srijita
Virat
[1]- Done          cat NewContacts
[2]+ Done          mkdir new
```

5.5 PIPE (|)

This | operator is very useful where the output of first command acts as an input to the second command.

SYNTAX:

[command] | [command] | [command]

USING PIPE (|)

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls | cat > file
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file
Contacts.csv
Dir4
f3
f4
file2.gz
file
new1
NewContacts
OldContacts
Page_Faults
PRACTICAL 1.docx
Practical 5.pdf
Scripting
```

5.6 SORT

Sort is a standard command line program that prints the lines of its input or concatenation of all files listed in its argument list in sorted order.

SYNTAX:

```
sort [OPTION]... [FILE]...
sort [OPTION]... --files0-from=F
```

OPTIONS:

Tag	Description
-b, --ignore-leading-blanks	Ignore leading blanks.
-d, --dictionary-order	Consider only blanks and alphanumeric characters.
-f, --ignore-case	Fold lower case to upper case characters.
-g, --general-numeric-sort	Compare according to general numerical value.
-i, --ignore-nonprinting	Consider only printable characters.
-M, --month-sort	Compare (unknown) < `JAN' < ... < `DEC'.
-h, --human-numeric-sort	Compare human readable numbers (e.g., "2K", "1G").
-n, --numeric-sort	Compare according to string numerical value.
-R, --random-sort	Sort by random hash of keys.
--random-source=FILE	Get random bytes from FILE.
-r, --reverse	Reverse the result of comparisons.
--sort=WORD	Sort according to WORD: general-numeric -g, human-numeric -h, month -M, numeric -n, random -R, version -V.
-V, --version-sort	Natural sort of (version) numbers within text.

USING SORT WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

USING SORT WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -b file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -d file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -f file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -g file9
```

```
.
```

End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

7 wonders of the world

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -i file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -M file9
```

```
.
```

7 wonders of the world
End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -h file9
```

End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

7 wonders of the world

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -n file9.
```

End of the file.
on a factual basis, linux is better
We are studying Operating System
Yes it is the 9th file.

7 wonders of the world

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -R file9
```

End of the file.

```
We are studying Operating System
Yes it is the 9th file.
7 wonders of the world
.
on a factual basis, linux is better
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sort -V file9
7 wonders of the world
End of the file.
We are studying Operating System
Yes it is the 9th file.
on a factual basis, linux is better
```

5.7 HEAD

head makes it easy to output the first part of files. Head, by default, prints the first 10 lines of each FILE to standard output.

SYNTAX:

```
head [OPTION]... [FILE]...
```

OPTIONS:

TAG	DESCRIPTION
Head -v	It prints the file header with the first 10 lines of a file
Head -c num	It prints only the first [num] bytes in a file
Head -n num	It prints only the first [num] lines in a file

USING HEAD WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ head file
1
2
3
4
5
6
7
8
9
10
```

USING HEAD WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ head -v file
==> file <==
1
2
3
4
5
6
7
8
9
10
Iammaddoc@iammaddoc-virtual-machine:~/OS$ head -n 5 file
1
2
3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ head -c 5 file
1
2
```

5.8 TAIL

Tail outputs the last part, or "tail", of files. By default, tail prints the last 10 lines of each *file* to standard output. If you specify more than one file, each set of output is prefixed with a header showing the file name.

SYNTAX:

```
tail [OPTION]..[FILE]
```

OPTIONS:

TAG	DESCRIPTION
Tail -v	It prints the file footer with the last 10 lines of a file
Tail-n num	It prints only the last [num] lines in a file
Tail -c num	It prints only the last [num] bytes in a file

USING TAIL WITHOUT OPTIONS:

Tail:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ tail file
5
6
7
8
9
10
11
12
13
14
```

USING TAIL WITH OPTIONS

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ tail -v file
==> file <==
5
6
7
8
9
10
11
12
13
14
```

5.9 MORE

Displays text, one screen at a time.

SYNTAX:

```
more [OPTION] filename
```

OPTIONS:

TAG	DESCRIPTION
more -p	Do not scroll, instead, clear the whole screen and then display the text
more -s	Squeeze multiple blank lines into one blank line

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ more +2 file
This is the second line
This is the third line
This is the fourth line in the file
This is the last line in the file
```

5.10 LESS

Less is the opposite of the more command and is used to view files rather than open them

SYNTAX:

```
less [OPTION...] [FILE...]
```

USING LESS WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ less file
This is the first line
This is the second line
This is the third line
This is the fourth line in the file
This is the last line in the file
```

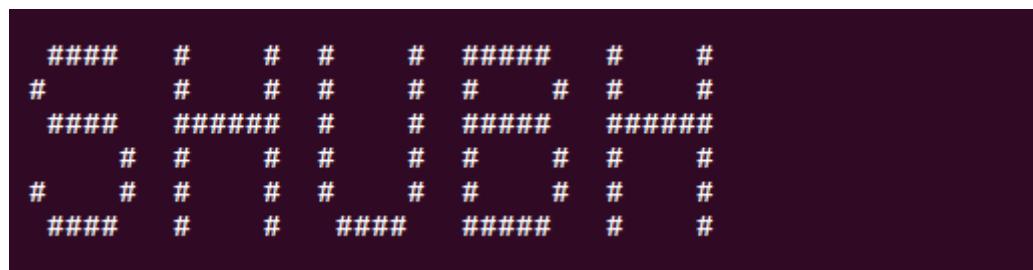
5.11 BANNER

'banner' is a command which prints a high resolution text banner on the system console or if you have a printer connected to your machine, you can redirect the output to the printer. This utility is available on all Linux / Unix platforms.

SYNTAX: banner text

USING BANNER:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ banner SHUBH
```



5.12 USERMOD

Modify a user account

SYNTAX:

```
usermod [OPTIONS...] LOGIN
```

OPTIONS:

TAG	DESCRIPTION
-a, --append	Add the user to the supplementary group(s). Use only with the -G option.
-d, --home HOME_DIR	The user's new login directory.

<code>-e EXPIRE_DATE, -- expiredate EXPIRE_DATE</code>	The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD.
<code>-g GROUP, --gid GROUP</code>	The group name or number of the user's new initial login group. The group must exist.
<code>-l NEW_LOGIN, -- login NEW_LOGIN</code>	The name of the user will be changed from <i>LOGIN</i> to <i>NEW_LOGIN</i> . Nothing else is changed.
<code>-m, --move-home</code>	Move the content of the user's home directory to the new location.
<code>-o, --non-unique</code>	When used with the -u option, this option allows to change the user ID to a non-unique value.
<code>-p PASSWORD, -- password PASSWORD</code>	The encrypted password, as returned by crypt.
<code>-R CHROOT_DIR, -- root CHROOT_DIR</code>	Apply changes in the <i>CHROOT_DIR</i> chroot directory and use the configuration files from the <i>CHROOT_DIR</i> directory.

USING USERMOD WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:/home$ sudo usermod -d /tanveer temp1
```

```
[sudo] password for Iammaddoc:
```

```
Iammaddoc@iammaddoc-virtual-machine:/home$ sudo usermod -L temp
```

PRACTICAL - 6

AIM: To implement the following commands in LINUX:

ADDUSER, USERADD, DELUSER, USERDEL, UMASK, CHMOD, WALL, WRITE, CUT, CAL, ULIMIT, CLEAR, FINGER

6.1 ADDUSER

The adduser command adds a new user to the system. adduser and addgroup add users and groups to the system according to command line options and configuration information in /etc/adduser.conf. They are friendlier front ends to the low-level tools like useradd, groupadd and usermod programs, by choosing policy-conformant UID and GID values, creating a home directory with skeletal configuration, running a custom script, and other features.

SYNTAX:

```
adduser [--system] [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
        [--firstuid ID] [--lastuid ID] [--ingroup GROUP | --gid ID]
        [--disabled-password] [--disabled-login] [--gecos GECOS]
        [--add_extra_groups] [--encrypt-home] [--quiet] [--debug]
        [--force-badname] [--help|-h] [--version] [--conf FILE] user
```

OPTIONS:

TAG	DESCRIPTION
--conf FILE	Use FILE instead of /etc/adduser.conf.
--home	Use DIR as the user's home directory, rather than the default specified by the configuration file.
--quiet	Suppress informational messages, only show warnings and errors.
--debug	Be verbose, which is most useful if you want to nail down a problem with adduser.
--system	Create a system user or group.

USING ADDUSER WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ adduser temp
adduser: Only root may add a user or group to the system.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo adduser temp
Adding user `temp' ...
Adding new group `temp' (1001) ...
Adding new user `temp' (1001) with group `temp' ...
The home directory `/home/temp' already exists. Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for temp
Enter the new value, or press ENTER for the default
    Full Name []: Shubhkirti Sharma
    Room Number []: 101
    Work Phone []: 9968591776
    Home Phone []: 9968591776
    Other []: none
Is the information correct? [Y/n] y
```

USING ADDUSER WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo adduser --quiet temp
adduser: The user `temp' already exists.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo adduser --quiet temp1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for temp1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []
Is the information correct? [Y/n] y
Iammaddoc@iammaddoc-virtual-machine:~$ sudo adduser --conf temp2
adduser: Only one or two names allowed.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo adduser --debug temp2
Adding user `temp2' ...
Selecting UID from range 1000 to 29999 ...
Selecting GID from range 1000 to 29999 ...
Adding new group `temp2' (1004) ...
/usr/sbin/groupadd -g 1004 temp2
Adding new user `temp2' (1004) with group `temp2' ...
/usr/sbin/useradd -d /home/temp2 -g temp2 -s /bin/bash -u 1004 temp2
Creating home directory `/home/temp2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
No password supplied
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
/usr/bin/chfn temp2
Changing the user information for temp2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []
Is the information correct? [Y/n] y
Iammaddoc@iammaddoc-virtual-machine:~$ adduser --home /home/newuserhome newuser
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for temp1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []
Is the information correct? [Y/n] y
```

6.2 USERADD

useradd is a low-level utility for adding users to a system. In general, the more friendly adduser should be used instead.

SYNTAX:

```
useradd [options] LOGIN
```

OPTIONS:

TAG	DESCRIPTION
-d, --home <i>HOME_DIR</i>	The new user will be created using <i>HOME_DIR</i> as the value for the user's login directory.
-D, --defaults	Set new default values.
-e, --expiredate <i>EXPIRE_DATE</i>	The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD.
-f, --inactive <i>INACTIVE</i>	The number of days after a password expires until the account is permanently disabled.
-l, --no-log-init	Do not add the user to the lastlog and faillog databases.
-m, --create-home	Create the user's home directory if it does not exist

USING ADDUSER WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd temp4
[sudo] password for Iammaddoc:
Sorry, try again.
[sudo] password for Iammaddoc:
```

USING USERADD WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
Iammaddoc@iammaddoc-virtual-machine:~$ useradd --home-dir /home/iammaddoc/ temp6
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd --home-dir /home/iammaddoc/ temp6
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd --expiredate 2018-02-25 temp7
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd -f 10 temp8
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd -l temp9
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd -m temp7
useradd: user 'temp7' already exists
Iammaddoc@iammaddoc-virtual-machine:~$ sudo useradd -m temp10
```

6.3 DELUSER

deluser and delgroup remove users and groups from the system according to command line options and configuration information in /etc/deluser.conf and /etc/adduser.conf. They are friendlier front ends to the userdel and groupdel programs, removing the home directory as option or even all files on the system owned by the user to be removed, running a custom script, and other features.

SYNTAX:

```
deluser [--quiet] [--system] [--help] [--version] [--conf FILE] [--force]
[--remove-home] [--remove-all-files] [--backup]
[--backup-to DIR] user
```

OPTIONS:

TAG	DESCRIPTION
--quiet	Suppress progress messages.
--system	Only delete if user/group is a system user/group. This avoids accidentally deleting non-system users/groups. Additionally, if the user does not exist, no error value is returned.
--backup	Backup all files contained in the userhome and the mailspool-file to a file named /\$user.tar.bz2 or /\$user.tar.gz.
--backup-to	Place the backup files not in / but in the directory specified by this parameter. This implicitly sets --backup also.
--remove-home	Remove the home directory of the user and its mailspool. If --backup is specified, the files are deleted after having performed the backup.

USING DELUSER WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:$ sudo deluser temp
```

USING DELUSER WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:$ deluser -remove-home temp1
Iammaddoc@iammaddoc-virtual-machine:~$ deluser --force temp3
Iammaddoc@iammaddoc-virtual-machine:~$ deluser backup temp8s
Iammaddoc@iammaddoc-virtual-machine:~$ deluser --backup-to /home temp4
Iammaddoc@iammaddoc-virtual-machine:~$ sudo deluser temp2
Removing user `temp2' ...
Warning: group `temp2' has no more members.
Done.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo deluser --quiet temp4
Iammaddoc@iammaddoc-virtual-machine:~$ sudo deluser --system temp6
The user `temp6' is not a system user. Exiting.
Iammaddoc@iammaddoc-virtual-machine:~$ sudo deluser --remove-all-files temp7
Looking for files to backup/remove ...
/usr/sbin/deluser: Cannot handle special file /sys/kernel/security/apparmor/.null
/usr/sbin/deluser: Cannot handle special file /lib/systemd/system/mountdevsubfs.service
/usr/sbin/deluser: Cannot handle special file /lib/systemd/system/umountfs.service
/usr/sbin/deluser: Cannot handle special file /lib/systemd/system/rc.service
/usr/sbin/deluser: Cannot handle special file /lib/systemd/system/x11-common.service
/usr/sbin/deluser: Cannot handle special file /lib/systemd/system/bootlogs.service
...
...
...
```

6.4 USERDEL

userdel is a low-level utility for removing users. (It is not very user-friendly. In most cases, administrators are advised to use the friendlier deluser utility instead). The userdel command modifies the system account files, deleting all entries that refer to the username LOGIN. The named user must exist.

SYNTAX:

```
userdel [options] LOGIN
```

OPTIONS:

TAG	DESCRIPTION
-f, --force	This option forces the removal of the user account, even if the user is still logged in.
-r, --remove	Files in the user's home directory will be removed along with the home directory itself and the user's mail spool.

USING USERDEL WITHOUT OPTIONS

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo userdel temp8
```

USING USERDEL WITH OPTIONS

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo userdel -f temp9
Iammaddoc@iammaddoc-virtual-machine:~$ sudo userdel -R temp10
userdel: invalid chroot path 'temp10'
```

6.5 UMASK

On Linux and other Unix-like operating systems, new files are created with a default set of permissions. Specifically, a new file's permissions may be restricted in a specific way by applying a permissions "mask" called the umask.

SYNTAX:

```
umask [-S] [mask]
```

OPTIONS:

TAG	DESCRIPTION
-S	Accept a symbolic representation of a mask, or return one.
Mask	If a valid <i>mask</i> is specified, the umask is set to this value. If no <i>mask</i> is specified, the current umask value is returned.

USING UMASK WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ umask
0022
Iammaddoc@iammaddoc-virtual-machine:~$ su
Password:
root@iammaddoc:/home/iammaddoc# umask
0022
root@iammaddoc:/home/iammaddoc# su any
any@iammaddoc:/home/iammaddoc$ umask
0002
```

USING UMASK WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ umask 0555
Iammaddoc@iammaddoc-virtual-machine:~$ umask
0555
Iammaddoc@iammaddoc-virtual-machine:~/OS$ umask 0577 file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ umask 0222 file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -ll
```

6.6 CHMOD

On Linux and other Unix-like operating systems, there is a set of rules for each file which defines who can access that file, and how they can access it. These rules are called file permissions or file modes. The command name chmod stands for "change mode", and it is used to define the way a file can be accessed.

SYNTAX:

```
chmod [OPTION]... MODE[,MODE]... FILE...
chmod [OPTION]... OCTAL-MODE FILE...
chmod [OPTION]... --reference=FILE FILE..
```

OPTIONS:

TAG	DESCRIPTION
-c, --changes	Like --verbose, but gives verbose output only when a change is actually made.
-f, --silent, --quiet	Quiet mode; suppress most error messages.
-v, --verbose	Verbose mode; output a diagnostic message for every file processed.
--no-preserve-root	Do not treat '/' (the root directory) in any special way, which is the default setting.
--preserve-root	Do not operate recursively on '/'.
--reference=FILE	Set permissions to match those of file FILE, ignoring any specified MODE.
-R, --recursive	Change files and directories recursively.

USING CHMOD WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod 555 file2
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -ll
total 64
drwxr-xr-x 2 Iammaddoc iammaddoc 4096 Feb 19 00:48 Dir4
-rw-r--r-- 1 Iammaddoc iammaddoc 26 Feb 18 06:13 f3
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 06:13 f4
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 06:30 file
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 07:13 file0
-rw-r--r-- 2 Iammaddoc iammaddoc 42 Feb 19 00:12 file1
-rwxr--r-x 1 Iammaddoc iammaddoc 42 Feb 19 00:46 file2
-rw-r--r-- 1 Iammaddoc iammaddoc 79 Feb 18 07:14 file3
-rw-r--r-- 1 Iammaddoc iammaddoc 85 Feb 18 06:10 file4
-rw-r--r-- 1 Iammaddoc iammaddoc 57 Feb 18 06:10 file5
-rw-r--r-- 1 Iammaddoc iammaddoc 123 Feb 18 06:22 file6
-rw-r--r-- 1 Iammaddoc iammaddoc 44 Jan 10 17:30 file7
-rw-r--r-- 1 Iammaddoc iammaddoc 27 Feb 18 06:32 file8
-rw-r--r-- 1 Iammaddoc iammaddoc 137 Feb 19 00:58 file9
-rw-r--r-- 1 Iammaddoc iammaddoc 18955 Feb 24 14:28 file.odt
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod ugo+rwx file4
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls -ll
total 64
drwxr-xr-x 2 Iammaddoc iammaddoc 4096 Feb 19 00:48 Dir4
```

```
-rw-r--r-- 1 Iammaddoc iammaddoc 26 Feb 18 06:13 f3
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 06:13 f4
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 06:30 file
-rw-r--r-- 1 Iammaddoc iammaddoc 0 Feb 18 07:13 file0
-rw-r--r-- 2 Iammaddoc iammaddoc 42 Feb 19 00:12 file1
-r-xr-xr-x 1 Iammaddoc iammaddoc 42 Feb 19 00:46 file2
----- 1 Iammaddoc iammaddoc 79 Feb 18 07:14 file3
-rwxrwxrwx 1 Iammaddoc iammaddoc 85 Feb 18 06:10 file4
-rw-r--r-- 1 Iammaddoc iammaddoc 57 Feb 18 06:10 file5
-rw-r--r-- 1 Iammaddoc iammaddoc 123 Feb 18 06:22 file6
-rw-r--r-- 1 Iammaddoc iammaddoc 44 Jan 10 17:30 file7
-rw-r--r-- 1 Iammaddoc iammaddoc 27 Feb 18 06:32 file8
-rw-r--r-- 1 Iammaddoc iammaddoc 137 Feb 19 00:58 file9
-rw-r--r-- 1 Iammaddoc iammaddoc 18955 Feb 24 14:28 file.odt
```

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ chmod 557 file3

USING CHMOD WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -c 557 file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -c 555 file3
mode of 'file3' changed from 0557 (r-xr-xrwx) to 0555 (r-xr-xr-x)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -v 555 file3
mode of 'file3' retained as 0555 (r-xr-xr-x)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -s file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -f 000 file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -c ugo=r file5
mode of 'file5' changed from 0644 (rw-r--r--) to 0444 (r--r--)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -c 777 file6
mode of 'file6' changed from 0644 (rw-r--r--) to 0777 (rwxrwxrwx)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ chmod -v ugo+rwx file4
mode of 'file4' retained as 0777 (rwxrwxrwx)
```

6.7 WALL

Wall displays the contents of file or, by default, its standard input, on the terminals of all currently logged in users. The command will cut any lines that are over 79 characters to new lines. Short lines are white space padded to have 79 characters. The command will always put carriage return and new line at the end of each line.

SYNTAX:

wall [-n] [-t TIMEOUT] [file]

OPTIONS:

TAG	DESCRIPTION
-n, --nobanner	Suppress banner
-t, --timeout <i>TIMEOUT</i>	Write timeout to terminals in seconds. <i>TIMEOUT</i> must be positive integer. Default value is 300 seconds, which is a legacy from time when people ran terminals over modem lines.
-V, --version	Display version information and exit.
-h, --help	Display a help message and exit.

USING WALL WITHOUT OPTIONS:

Iammaddoc@iammaddoc-virtual-machine:~/OS\$ wall 'hello'

USING WALL WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wall -n 'hello'
wall: --nobanner is available only for root
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sudo wall -n 'hello'
[sudo] password for Iammaddoc:
Iammaddoc@iammaddoc-virtual-machine:~/OS$ wall --version 'hello'
wall from util-linux 2.27.1
```

6.8 WRITE

The write utility allows you to communicate with other users by copying lines from your terminal to theirs.

SYNTAX:

```
write user [tty]
```

OPTIONS:

TAG	DESCRIPTION
user	The user to write to.
tty	The specific terminal to write to, if the user is logged in to more than one session.

USING WRITE WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ write yay
```

USING WRITE WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ write yay tty7
```

DIFFERENCE BETWEEN WALL AND WRITE:

- Write is to send a short message to another online user. It does not keep a window open at both side as the write command do, so not suitable for continuous chatting. All users can use this also.
- Wall is for sending short message not to a single user/group but all online users in the network. Only admins can use this command. Write can be blocked by each user but wall cant.

6.9 CUT

Print selected parts of lines from each FILE to standard output.

SYNTAX:

```
cut OPTION... [FILE]...
```

OPTIONS:

Tag	Description
<code>-b, --bytes=LIST</code>	Select only the bytes from each line as specified in <i>LIST</i> . <i>LIST</i> specifies a byte, a set of bytes, or a range of bytes; see Specifying <i>LIST</i> below.
<code>-c, --characters=LIST</code>	Select only the characters from each line as specified in <i>LIST</i> .
<code>-d, --delimiter=DELIM</code>	use character <i>DELIM</i> instead of a tab for the field delimiter.
<code>-f, --fields=LIST</code>	select only these fields on each line; also print any line that contains no delimiter character, unless the <code>-s</code> option is specified.
<code>-n</code>	This option is ignored, but is included for compatibility reasons.
<code>--complement</code>	complement the set of selected bytes, characters or fields.
<code>-s, --only-delimited</code>	do not print lines not containing delimiters.
<code>--output-delimiter=STRING</code>	use <i>STRING</i> as the output delimiter string. The default is to use the input delimiter.

USING WHATIS WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat data
One
Two
Three
Four
five
alpha
beta
gamma
delta
epsilon
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cut -f 3 data
three
gamma
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cut 2-4 data
Two
Three
four
beta
gamma
delta
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cut -c 3-12 data.txt
E
Two
thre
pha
beta
g
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cut -b 3-12 data.txt
E
Two
thre
pha
beta
g
```

6.10 CAL

The cal utility displays a simple calendar in traditional format and ncal offers an alternative layout, more options and the date of Easter. The new format is a little cramped but it makes a year fit on a 25x80 terminal. If arguments are not specified, the current month is displayed.

SYNTAX:

```
cal [-3hjy] [-A number] [-B number] [[month] year]
```

OPTIONS:

Option	Description
-h	Don't highlight today's date.
-m month	Specify a month to display.
-y year	Specify a year to display. For example, -y 1970 will display the entire calendar of the year 1970.
-3	Display last month, this month, and next month.
-1	Display only this month. This is the default.
-A num	Display num months occurring after any months already specified.
-B num	Display num months occurring before any months already specified.
-d YYYY-MM	Operate as if the current month is number MM of year YYYY.

USING CAL WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal
February 2018
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
```

USING CAL WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -h
February 2018
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
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -j
February 2018
Su Mo Tu We Th Fr Sa
                  32 33 34
35 36 37 38 39 40 41
42 43 44 45 46 47 48
49 50 51 52 53 54 55
56 57 58 59
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -m Jan
January 2018
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
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -y
```

```

2018
January          February          March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6   1  2  3  4  5  6  7  8  9 10   1  2  3  4  5  6  7  8  9 10
  7  8  9 10 11 12 13   4  5  6  7  8  9 10 11 12 13 14 15 16 17 11 12 13 14 15 16 17
14 15 16 17 18 19 20   11 12 13 14 15 16 17 18 19 20 21 22 23 24 18 19 20 21 22 23 24
21 22 23 24 25 26 27   25 26 27 28               25 26 27 28 29 30 31
28 29 30 31
.....          30 31
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -3
2018
January          February          March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6   1  2  3  4  5  6  7  8  9 10   1  2  3  4  5  6  7  8  9 10
  7  8  9 10 11 12 13   4  5  6  7  8  9 10 11 12 13 14 15 16 17 11 12 13 14 15 16 17
14 15 16 17 18 19 20   11 12 13 14 15 16 17 18 19 20 21 22 23 24 18 19 20 21 22 23 24
21 22 23 24 25 26 27   25 26 27 28               25 26 27 28 29 30 31
28 29 30 31
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cal -B 3
November 2017      December 2017      January 2018
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4           1  2           1  2  3  4  5  6
  5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16 14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23 21 22 23 24 25 26 27
26 27 28 29 30         24 25 26 27 28 29 30 28 29 30 31
31
.....

```

6.11 ULIMIT

To get and set the user limit.

SYNTAX:

`ulimit [OPTIONS]`

OPTIONS:

TAG	DESCRIPTION
<code>ulimit -a</code>	To display all of your current settings
<code>ulimit -c</code>	To display size of all core files
<code>ulimit -u</code>	To display max user processes

USING ULIMIT WITH OPTIONS:

```

Iammaddoc@iammaddoc-virtual-machine:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size            (kbytes, -d) unlimited
scheduling priority      (-e) 0
file size                (blocks, -f) unlimited
pending signals          (-i) 7650
max locked memory        (kbytes, -l) 16384
max memory size          (kbytes, -m) unlimited
open files               (-n) 1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority        (-r) 0
stack size                (kbytes, -s) 8192
cpu time                 (seconds, -t) unlimited
max user processes        (-u) 7650
virtual memory             (kbytes, -v) unlimited
file locks                (-x) unlimited

```

```
iammaddoc@iammaddoc-virtual-machine:~$ ulimit -c  
0  
iammaddoc@iammaddoc-virtual-machine:~$ ulimit -u  
7650
```

6.12 CLEAR

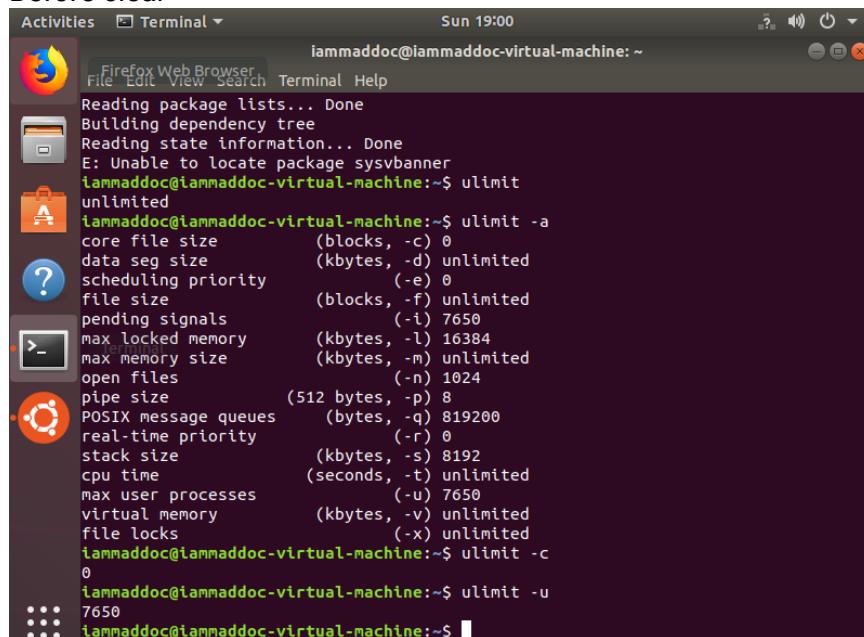
Clears the screen.

SYNTAX:

```
clear
```

USING CLEAR:

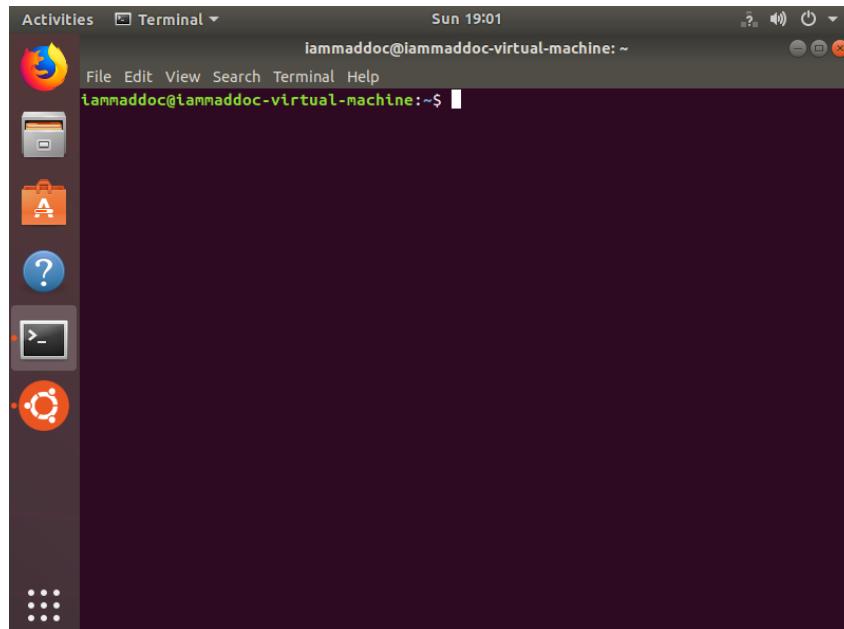
Before clear



A screenshot of a Linux desktop environment. On the left is a dock with icons for Firefox, Dash, Applications, Help, Terminal, and others. In the center is a terminal window titled 'Terminal'. The terminal shows the following command and its output:

```
Sun 19:00  
iammaddoc@iammaddoc-virtual-machine: ~  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
E: Unable to locate package sysvbanner  
iammaddoc@iammaddoc-virtual-machine:~$ ulimit  
unlimited  
iammaddoc@iammaddoc-virtual-machine:~$ ulimit -a  
core file size          (blocks, -c) 0  
data seg size           (kbytes, -d) unlimited  
scheduling priority     (-e) 0  
file size               (blocks, -f) unlimited  
pending signals          (-i) 7650  
max locked memory       (kbytes, -l) 16384  
max memory size         (kbytes, -m) unlimited  
open files              (-n) 1024  
pipe size               (512 bytes, -p) 8  
POSIX message queues    (bytes, -q) 819200  
real-time priority      (-r) 0  
stack size              (kbytes, -s) 8192  
cpu time                (seconds, -t) unlimited  
max user processes       (-u) 7650  
virtual memory          (kbytes, -v) unlimited  
file locks              (-x) unlimited  
iammaddoc@iammaddoc-virtual-machine:~$ ulimit -c  
0  
iammaddoc@iammaddoc-virtual-machine:~$ ulimit -u  
7650  
iammaddoc@iammaddoc-virtual-machine:~$
```

After clear



A screenshot of the same Linux desktop environment as before, but after the 'clear' command has been run. The terminal window now contains a single blank line, indicating that the screen has been cleared.

6.13 FINGER

finger looks up and displays information about system users.

SYNTAX:

```
finger [-lmsp] [user ...] [user@host ...]
```

OPTIONS:

TAG	DESCRIPTION
-s	Displays the user's login name, real name, terminal name and write status, idle time, login time, office location and office phone number.
-l	Produces a multi-line format displaying all of the information described for the -s option as well as the user's home directory, home phone number, login shell, mail status, and the contents of the files ".plan", ".project", ".pgpkey" and ".forward" from the user's home directory.
-p	Prevents the -l option of finger from displaying the contents of the ".plan", ".project" and ".pgpkey" files.
-m	Prevent matching of usernames. The <i>user</i> is usually a login name; however, matching will also be done on the users' real names, unless the -m option is supplied. All name matching performed by finger is case insensitive.

USING FINGER WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/FILE$ finger
Login      Name      Tty      Idle  Login Time   Office      Office Phone
Iammaddoc  iammaddoc  tty7      1d    Apr 26 20:34 (:0)
```

USING FINGER WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/FILE$ finger -l
Login: Iammaddoc                         Name: iammaddoc
Directory: /home/iammaddoc                 Shell: /bin/bash
On since Thu Apr 26 20:34 (IST) on tty7 from :0
    1 day 21 hours idle
No mail.
No Plan.

Iammaddoc@iammaddoc-virtual-machine:~/OS/FILE$ finger -lp
Login: Iammaddoc                         Name: yay
Directory: /home/iammaddoc                 Shell: /bin/bash
On since Thu Apr 26 20:34 (IST) on tty7 from :0
    1 day 21 hours idle
No mail.

Iammaddoc@iammaddoc-virtual-machine:~/OS/FILE$ finger -p temp
Login: temp                               Name: Iammaddoc Iammaddoc
Directory: /home/temp                      Shell: /bin/bash
Office: 101, 996-859-1776
Home Phone: 996-859-1776
Never logged in.
No mail.
Iammaddoc
```

PRACTICAL - 7

AIM: To implement the following commands in LINUX:

GZIP, ZIP, UNZIP, GUNZIP, TAR, SPLIT, PS, KILL, NOHUP, BATCH, AT, CRONTAB, NICE, TOP

7.1 GZIP

gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership modes, access, and modification times. (The default extension is -gz for VMS, z for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. gzip will only attempt to compress regular files. In particular, it will ignore symbolic links.

SYNTAX:

```
gzip [ -acdfhlLnNrtvV19 ] [-S suffix] [ name ... ]
```

OPTIONS:

TAG	DESCRIPTION
-a --ascii	ASCII text mode: convert end-of-lines using local conventions.
-d --decompress	Decompress.
--uncompress	
-h --help	Display a help screen and quit.
-q --quiet	Suppress all warnings.
-r --recursive	Travel the directory structure recursively.
-S .suf --suffix .suf	When compressing, use suffix .suf instead of .gz.
-t --test	Test. Check the compressed file integrity.
-v --verbose	Verbose. Display the name and percentage reduction for each file compressed or decompressed.
-V --version	Version. Display the version number and compilation options then quit.
-# --fast --best	Regulate the speed of compression using the specified digit #

USING GZIP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip file1
gzip: file1 has 1 other link -- unchanged
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip file4
```

USING GZIP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -c file5
◆'Z#file5◆◆◆K◆0SH◆◆I5◆
◆◆,V◆◆◆T◆HQq      X◆+81◆ 'U◆$◆◆D!3,f◆#◆9
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -d file5
gzip: file5: unknown suffix -- ignored
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -d file4
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -f file5
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -f file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -k file3
gzip: file3: Permission denied
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -fk file3
gzip: file3: Permission denied
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sudo gzip -k file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4 f4    file0    file2.gz   file3.gz   file5.gz   file7    file9
f3    file    file1.gz   file3    file4    file6    file8    file.odt
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -l
Iammaddoc Iammaddoc is the current user.
```

```

gzip: stdin: not in gzip format
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -l file4
gzip: file4: not in gzip format
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -l file3.gz
gzip: file3.gz: Permission denied
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sudo gzip -l file3.gz
      compressed           uncompressed   ratio   uncompressed_name
                     91                      79    15.2% file3
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -n file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4  f4  file0  file2.gz  file3.gz  file5.gz  file7.gz  file9
f3  file  file1.gz  file3  file4  file6  file8  file.odt
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -d file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip -N file7
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Dir4  f4  file0  file2.gz  file3.gz  file5.gz  file7.gz  file9
f3  file  file1.gz  file3  file4  file6  file8  file.odt

```

7.2 ZIP

ZIP is a compression and file packaging utility for Unix. Each file is stored in single .zip { .zip-filename} file with the extension .zip.

SYNTAX:

```
zip [OPTIONS] zipfile [FILES]
```

OPTIONS:

TAG	DESCRIPTION
Add	Update existing entries and add new files and create it if the archive does not exist. This option is the default mode.
update (-u)	Update existing entries if newer on the file system and add new files. If the archive does not exist issue warning then create a new archive.
freshen (-f)	Update existing entries of an archive if newer on the file system. Does not add new files to the archive.
delete (-d)	Select entries in an existing archive and delete them.
copy (-U)	Select entries in an existing archive and copy them to a new archive.

USING ZIP WITHOUT OPTIONS:

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip zip.zip file0 file8 file9
  adding: file0 (stored 0%)
  adding: file8 (stored 0%)
  adding: file9 (deflated 99%)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Contacts.csv  file0  file4  file.odt  OldContacts      zip.zip
Dir4          file0.zip  file5.gz  myfile.txt  ouput
f3            file1.gz  file6    ne        Page_Faults
f4            file2.gz  file7.gz  new       PRACTICAL 1.docx
file          file3    file8    new1     Practical 5.pdf
FILE          file3.gz  file9    NewContacts  Scripting

```

USING ZIP WITH OPTIONS:

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip -d zip.zip file0
deleting: file0

```

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip -u zip.zip file0
  adding: file0 (stored 0%)

```

```

Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat >file0
hey there

```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ cat file0
hey there
Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip -u zip.zip file0
updating: file0 (stored 0%)

Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip -m zip.zip file0 file4 f3
updating: file0 (stored 0%)
  adding: file4 (deflated 16%)
  adding: f3 (stored 0%)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ ls
Contacts.csv  file1.gz  file7.gz  new          PRACTICAL 1.docx
Dir4          file2.gz  file8    new1         Practical 5.pdf
f4            file3    file9    NewContacts  Scripting
file          file3.gz  file.odt  OldContacts zip.zip
FILE          file5.gz  myfile.txt ouput
file0.zip     file6    ne      Page_Faults

Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip directory.zip docs/
  adding: docs/ (stored 0%)
Iammaddoc@iammaddoc-virtual-machine:~/OS$ zip -r directory.zip docs/
updating: docs/ (stored 0%)
  adding: docs/unix.pdf (stored 0%)
  adding: docs/linux.pdf (stored 0%)
  adding: docs/windows.pdf (stored 0%)

Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ zip -v zip.zip *.pdf
  adding: linux.pdf   (in=0) (out=0) (stored 0%)
  adding: unix.pdf    (in=0) (out=0) (stored 0%)
  adding: windows.pdf (in=0) (out=0) (stored 0%)
total bytes=0, compressed=0 -> 0% savings
```

7.3 UNZIP

The unzip command lists, extracts, and tests **compressed files in a ZIP archive**.

SYNTAX:

```
unzip [-Z] [-cflptTuvz[abjnoqsCDKLMUVWX$/:^]] file[.zip] [file(s) ...]
      [-x xfile(s) ...] [-d exdir]
```

OPTIONS:

-Z	zipinfo mode. If the first option on the command line is -Z, the remaining options are taken to be zipinfo options.
-A	[OS/2, Unix DLL] print extended help for the DLL's programming interface (API).
-c	extract files to stdout/screen ("the CRT").
-p	extract files to pipe (standard output, stdout).
-t	test archive files.
-T	[most OSes] set the timestamp on the archive(s) to that of the newest file in each one.
-u	update existing files and create new ones if needed.

USING UNZIP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip zip.zip
Archive: zip.zip
replace linux.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  extracting: linux.pdf
replace unix.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  extracting: unix.pdf
replace windows.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  extracting: windows.pdf
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ ls
linux.pdf  unix.pdf  windows.pdf  zip.zip
```

USING UNZIP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip -l zip.zip
Archive: zip.zip
  Length      Date      Time    Name
----- -----
  0  2018-04-28 23:42  linux.pdf
  0  2018-04-28 23:42  unix.pdf
  0  2018-04-28 23:42  windows.pdf
-----
          0           3 files

Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip -t zip.zip
Archive: zip.zip
  testing: linux.pdf          OK
  testing: unix.pdf          OK
  testing: windows.pdf        OK
No errors detected in compressed data of zip.zip.
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip -q zip.zip
replace linux.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace unix.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace windows.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip --ca zip.zip *x.pdf
Archive: zip.zip
replace linux.pdf? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  extracting: linux.pdf          [empty]
  extracting: unix.pdf          [empty]
Iammaddoc@iammaddoc-virtual-machine:~/OS/docs$ unzip --fo zip.zip
Archive: zip.zip
  extracting: linux.pdf
  extracting: unix.pdf
  extracting: windows.pdf
```

7.4 GUNZIP

Restore compressed files to their original form, renaming the files by removing the extension. Gunzip can currently decompress files created by gzip, zip, compress, compress -H or pack. The detection of the input format is automatic.

SYNTAX:

```
gunzip  [FILENAME]
```

OPTIONS:

TAG	DESCRIPTION
Gunzip -N	restores the original file name and time stamp
Gunzip -q	quiets all warnings
Gunzip -r	unzips the directory recursively

USING GUNZIP WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ sudo gunzip file3.gz
[sudo] password for Iammaddoc:
gzip: file3 already exists; do you wish to overwrite (y or n)? y
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -k file5.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -l file5.gz
      compressed      uncompressed      ratio   uncompressed_name
                  74                      57     12.3%   file5
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -r file7.gz
```

USING GUNZIP WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -N file1.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -q file1.gz
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gzip file1
Iammaddoc@iammaddoc-virtual-machine:~/OS$ gunzip -r file1.gz
```

7.5 TAR

"tar" stands for tape archive. It is an archiving file format.

tar was originally developed in the early days of Unix for the purpose of backing up files to tape-based storage devices. It was later formalized as part of the POSIX standard, and today is used to collect, distribute, and archive files, while preserving file system attributes such as user and group permissions, access and modification dates, and directory structures.

SYNTAX:

```
tar [-] A --catenate --concatenate | c --create | d --diff --compare |
--delete | r --append | t --list | --test-label | u --update |
x --extract --get [options] [pathname ...]
```

OPTIONS:

TAG	DESCRIPTION
A, --catenate, --concatenate	Append tar files to an archive.
c, --create	Create a new archive.
d, --diff, --compare	Calculate any differences between the archive and the file system.
--delete	Delete from the archive. (This function doesn't work on magnetic tapes).
r, --append	Append files to the end of a tar archive.
t, --list	List the contents of an archive.
--test-label	Test the archive label, and exit.
u, --update	Append files, but only those that are newer than the copy in the archive.
x, --extract, --get	Extract files from an archive.

USING TAR WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine~:$ tar -cf archive.tar file1 file2
```

DIFFERENCE BETWEEN GZIP, ZIP AND TAR**GZIP:**

- A .gz file was produced by gzip, which is a compression tool.
- It does not archive multiple files together. It compresses single files.
- If we tell it to compress several files, we get several compressed files.

TAR:

- The tar utilities is an archiving tool.
- It puts one or more files into an archive, and the common filename suffix for that is .tar.
- The name tar is an abbreviation for tape archive.

- The files that tar produces are uncompressed. GNU tar provides two options -z and -Z, which call gzip (.tar.gz) or compress respectively. They will produce a compressed version of that same archive.

ZIP:

- A .zip file was produced by a Zip utility of some sort.
- It isn't commonly used on Linux systems.
- There are Linux tools for dealing with .zip files. They are compressed archives, the conceptual equivalent of a .tar.gz file.

SOME COMBINATIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ zip ziptar.zip tar.tar
  adding: tar.tar (deflated 97%)
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ wc ziptar.zip
 3 12 474 ziptar.zip
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ tar -cvf tarzip zip.zip
zip.zip
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ wc tarzip
 3 25 10240 tarzip
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ gzip -k zip.zip
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ wc zip.zip.gz
 2 16 427 zip.zip.gz
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ zip tarzip.zip tar.tar
  adding: tar.tar (deflated 97%)
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ wc tarzip.
tarzip.gz  tarzip.zip
```

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IM0$ wc tarzip.zip
 3 12 474 tarzip.zip
```

7.6 SPLIT

`split` outputs fixed-size pieces of input INPUT to files named PREFIXaa, PREFIXab, ...

The default size for each split file is 1000 lines, and default PREFIX is "x". With no INPUT, or when INPUT is a dash ("‐"), read from standard input.

SYNTAX:

```
split [OPTION]... [INPUT [PREFIX]]
```

OPTIONS:

TAG	DESCRIPTION
<code>-a N, --suffix-length=N</code>	Use suffixes of length N (default 2)
<code>-b SIZE, --bytes=SIZE</code>	Write SIZE bytes per output file.
<code>-C SIZE, --line-bytes=SIZE</code>	Write at most SIZE bytes of lines per output file.
<code>-d, --numeric-suffixes</code>	Use numeric suffixes instead of alphabetic.
<code>-e, --elide-empty-files</code>	Do not generate empty output files with "-n"
<code>--filter=COMMAND</code>	Write to shell command COMMAND; file name is \$FILE
<code>-l NUMBER, --lines=NUMBER</code>	Put NUMBER lines per output file.
<code>-n CHUNKS, --number=CHUNKS</code>	Generate CHUNKS output files. (See below.)
<code>-u, --unbuffered</code>	Immediately copy input to output with "-n r/...".
<code>--verbose</code>	Print a verbose diagnostic just before each output file is opened.

USING SPLIT WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ split original o
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ ls
faa fac naa nac nae nag nai new original
fab file nab nad naf nah naj oaa xaa
```

USING SPLIT WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ cat >new
yayy
ok
thats it
fo
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ split -b 2 new n
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ la
naa nab nac nad nae naf nag nah nai naj new
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ cat >file
yya
okay
line1
line2
line5
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ split -l 2 file f
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ ls
faa fab fac file naa nab nac nad nae naf nag nah nai naj new
```

7.7 PS

Reports a snapshot of the status of currently running processes.

SYNTAX:

`ps [options]`

```
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ ps
  PID TTY          TIME CMD
 3158 pts/0    00:00:00 bash
 3292 pts/0    00:00:00 ps
iammaddoc@iammaddoc-virtual-machine:~/Desktop$
```

7.8 NO-HUP

When using the command shell, prefixing a command with nohup prevents the command from being aborted automatically when you log out or exit the shell. The name nohup stands for "no hangup."

SYNTAX:

`nohup command [command-argument ...]`
`nohup [OPTION]`

```
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ nohup sudo apt-get update
nohup: ignoring input and appending output to 'nohup.out'
[sudo] password for iammaddoc:
iammaddoc@iammaddoc-virtual-machine:~/Desktop$
```

7.9 KILL

The kill command sends a signal to a process. If you don't specify which signal to send, by default the KILL signal is sent, which terminates the process.

SYNTAX:

```
kill process-id ...
```

```
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ ps
  PID TTY          TIME CMD
 3158 pts/0    00:00:00 bash
 3519 pts/0    00:00:00 ps
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ kill 3158
```

NOTE: 'kill -9' forcefully kills a process

7.10 NICE

Runs a command with a modified scheduling priority. Niceness values range from -20 (most favorable to the process) to 19 (least favorable to the process). With no COMMAND, nice prints the current niceness level.

SYNTAX:

```
nice [OPTION] [COMMAND [ARG]...]
```

```
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ nice
0
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ █
```

7.11 AT & BATCH

at and batch read commands from standard input or a specified file which are to be executed at a later time, using the shell set by the user's environment variable SHELL, the user's login shell, or ultimately /bin/sh.

SYNTAX:

```
batch [-V] [-q queue] [-f file] [-mv] [TIME]
at [-V] [-q queue] [-f file] [-m ldbv] TIME
```

OPTIONS:

Tag	Description
At	executes commands at a specified time.
Atq	lists the user's pending jobs, unless the user is the superuser.
Atrm	deletes jobs, identified by their job number.
Batch	executes commands when system load levels permit.

Tag	Description
-V	prints the version number to standard error.
-m	Send mail to the user when the job has completed even if there was no output.
-f file	Reads the job from file rather than standard input.
-l	Is an alias for atq.
-d	Is an alias for atrm.
-c	cats the jobs listed on the command line to standard output.
-t time_arg	Submit the job to be run at the time specified by the time_arg option argument

USING AT WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ echo -c "hello" | at 12:11
warning: commands will be executed using /bin/sh
job 14 at Sun Apr 29 12:11:00 2018
```

USING AT WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS$ at -f file 12:10
warning: commands will be executed using /bin/sh
job 12 at Sun Apr 29 12:10:00 2018
Iammaddoc@iammaddoc-virtual-machine:~/OS$ at -l
12      Sun Apr 29 12:10:00 2018 a Iammaddoc
Iammaddoc@iammaddoc-virtual-machine:~/OS$ atq
12      Sun Apr 29 12:10:00 2018 a Iammaddoc
Iammaddoc@iammaddoc-virtual-machine:~/OS$ echo "hello" | at 12:11
warning: commands will be executed using /bin/sh
job 13 at Sun Apr 29 12:11:00 2018
-l:
Iammaddoc@iammaddoc-virtual-machine:~/OS$ at -l
13      Sun Apr 29 12:11:00 2018 a Iammaddoc
Iammaddoc@iammaddoc-virtual-machine:~/OS$ at --help
at: invalid option -- '-'
Usage: at [-V] [-q x] [-f file] [-mMlbv] timespec ...
      at [-V] [-q x] [-f file] [-mMlbv] -t time
      at -c job ...
      atq [-V] [-q x]
      at [ -rd ] job ...
      atrm [-V] job ...
      batch
```

USING BATCH WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~/OS/IMO$ batch
warning: commands will be executed using /bin/sh
at> echo "hey" >/dev/pts/18
at> echo `cat >file1` >/dev/pts/18
at> <EOT>
job 16 at Mon Apr 30 01:58:00 2018
```

7.12 CRONTAB

The crontab (short for "cron table") is a list of commands that are scheduled to run at regular time intervals on your computer system. The crontab command opens the crontab for editing, and lets you add, remove, or modify scheduled tasks.

SYNTAX:

```
crontab [-u user] file
```

RUNNING CRONTAB:

```
cat>output (Leave Empty)
cat>crontab_file
40-41 16 07 03 04 echo hurry up > /home/iammaddoc/output
crontab crontab_file
gedit output
```

DIFFERENCE BETWEEN AT, BATCH AND CRONTAB

BATCH:

- Batch are low priority processes.
- Batch executes commands when system load levels permit; in other words, when the load average drops below 1.5.
- They can be scheduled for once.
- Cannot be edited.

AT:

- AT executes commands at a specified time.
- AT can be scheduled for once.
- Cannot be edited but can be deleted and listed.

CRONTAB:

- Cron can be scheduled with different choices (minute, hour, month, day of month, day of week).
- Cron can be edited, deleted, listed and scheduled at any time and for repeated time intervals.

7.13 TOP

The top program provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel.

SYNTAX:

```
top -hv | -bcHisS -d delay -n limit -uU user | -p pid -w [cols]
```

OPTIONS:

TAG	DESCRIPTION
-h -v	Show library version and the usage prompt, then quit.
-b	Starts top in 'Batch' mode
-c	Starts top with the last remembered 'c' state reversed.
-H	Instructs top to display individual threads.
-i	Starts top with the last remembered 'i' state reversed.
-n limit	Specifies the maximum number of iterations, or frames

USING TOP:

```
iammaddoc@iammaddoc-virtual-machine:~/Desktop$ top

top - 20:13:08 up 3:01, 1 user, load average: 0.22, 0.06, 0.02
Tasks: 257 total, 1 running, 194 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.3 us, 2.3 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2017292 total, 298564 free, 1173284 used, 545444 buff/cache
KiB Swap: 969960 total, 956892 free, 13068 used. 682684 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
1071 iammadd+  20   0 395916  53940 29724 S  1.3  2.7  0:24.84 Xorg
1227 iammadd+  20   0 2977552 164564 58360 S  1.0  8.2  1:22.01 gnome-she+
3147 iammadd+  20   0 870412  38496 28292 S  0.7  1.9  0:03.10 gnome-ter+
3646 iammadd+  20   0 51316   3984   3304 R  0.3  0.2  0:00.07 top
 1 root       20   0 160900   9364  6328 S  0.0  0.5  0:03.21 systemd
 2 root       20   0     0     0     0 S  0.0  0.0  0:00.00 kthreadd
 4 root       0  -20    0     0     0 I  0.0  0.0  0:00.00 kworker/0+
 6 root       0  -20    0     0     0 I  0.0  0.0  0:00.00 mm_percpu+
 7 root       20   0     0     0     0 S  0.0  0.0  0:04.44 ksoftirqd+
 8 root       20   0     0     0     0 I  0.0  0.0  0:01.24 rcu_sched
 9 root       20   0     0     0     0 I  0.0  0.0  0:00.00 rcu_bh
10 root      rt   0     0     0     0 S  0.0  0.0  0:00.00 migration+
11 root      rt   0     0     0     0 S  0.0  0.0  0:00.02 watchdog/0
12 root      20   0     0     0     0 S  0.0  0.0  0:00.00 cpuhp/0
13 root      20   0     0     0     0 S  0.0  0.0  0:00.00 kdevtmpfs
14 root      0  -20    0     0     0 I  0.0  0.0  0:00.00 netns
```

7.14 SUDO

sudo ("superuser do", or "switch user do") allows a user with proper permissions to execute a command as another user, such as the superuser. sudo allows a permitted user to execute a command as another user, according to specifications in the /etc/sudoers file. The real and effective uid and gid of the issuing user are then set to match those of the target user account as specified in the passwd file.

SYNTAX:

```
sudo [-V | -h | -l | -L | -v | -k | -K | -s | [-H] [-P] [-S] [-b] |  
[-p prompt] [-c class|-] [-a auth_type] [-r role] [-t type] |  
[-u username|#uid] command
```

OPTIONS:

TAG	DESCRIPTION
-V	The -V (version) option causes sudo to print the version number and exit.
-l	The -l (list) option will print out the commands allowed (and forbidden) the user on the current host.
-L	The -L (list defaults) option will list out the parameters that may be set in a Defaults line along with a short description for each.
-h	The -h (help) option causes sudo to print a usage message and exit.
-K	The -K (sure kill) option to sudo removes the user's timestamp entirely.

-P	The -P (preserve group vector) option causes sudo to preserve the user's group vector unaltered.
-t	The -t (type) option causes the new (SELinux) security context to have the type (domain) specified by TYPE.
-S	The -S (stdin) option causes sudo to read the password from standard input instead of the terminal device.

USING SUDO WITHOUT OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo apt-get update
```

USING SUDO WITH OPTIONS:

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo shutdown -r now
```

```
Iammaddoc@iammaddoc-virtual-machine:~$ sudo -u iammaddoc ls /home/user/Documents
```

7.15 SU

The su (short for substitute user) command makes it possible to change a login session's owner (i.e., the user who originally created that session by logging on to the system) without the owner having to first log out of that session.

SYNTAX:

```
su [options] [commands] [-] [username]
```

USING SU:

```
Iammaddoc@iammaddoc-virtual-machine:~$ su
Password:
root@iammaddoc:/home/iammaddoc# exit
exit
Iammaddoc@iammaddoc-virtual-machine:~$ su -l
Password:
root@iammaddoc:~# exit
logout
Iammaddoc@iammaddoc-virtual-machine:~$ su -m
Password:
```

PRACTICAL - 8

AIM:

- (a) Introduction to vi editor
- (b) Introduction to vim editor
- (c) Introduction to command line arguments and logical statements in vi editor

Introduction

The default editor that comes with the UNIX operating system is called vi (**v**isual editor). The UNIX vi editor is a full screen editor and has two modes of operation:

1. Command mode commands which cause action to be taken on the file, and
2. Insert mode in which entered text is inserted into the file.

In the command mode, every character typed is a command that does something to the text file being edited; a character typed in the command mode may even cause the vi editor to enter the insert mode. In the insert mode, every character typed is added to the text in the file; pressing the <Esc> (*Escape*) key turns off the Insert mode.

To Start vi

To use vi on a file, type in vi filename. If the file named filename exists, then the first page (or screen) of the file will be displayed; if the file does not exist, then an empty file and screen are created into which you may enter text.

vi filename	edit filename starting at line 1
-------------	----------------------------------

To Exit vi

Usually the new or modified file is saved when you leave vi. However, it is also possible to quit vi without saving the file.

:wq	quit vi, writing out modified file to file named in original invocation
:q	quit vi
:w	Save file

Moving the cursors

The mouse does not move the cursor within the vi editor screen. So we use the keys listed below:

j	Move cursor down one line
k	Move cursor up one line
h	Move cursor left one character
l	Move cursor right one character
0	Move cursor to start of current line
\$	Move cursor to end of current line
w	Move cursor to beginning of next word
b	Move cursor to the end of preceding word

:0	Move cursor to first line of file
:\$	Move cursor to last line of a file

Inserting or adding text

Each of these commands puts the vi editor into insert mode.

:i	Insert text before cursor
:a	Append text after cursor

The vi editor has three modes, command mode, insert mode and command line mode.

1. **Command mode:** letters or sequence of letters interactively command vi. Commands are case sensitive. The ESC key can end a command.
2. **Insert mode:** Text is inserted. The ESC key ends insert mode and returns you to command mode. One can enter insert mode with the "i" (insert), "a" (insert after), "A" (insert at end of line), "o" (open new line after current line) or "O" (Open line above current line) commands.
3. **Command line mode:** One enters this mode by typing ":" which puts the command line entry at the foot of the screen.

(b) Introduction to vim editor

Functionally, **vim** is almost a proper superset of **vi**. Vim adds onto those features. Here are some of the extended vim features:

- Vim has been ported to a much wider range of OS's than vi.
- Vim includes support (syntax highlighting, code folding, etc) for several popular programming languages (C/C++, Python, Perl, shell, etc).
- Vim integrates with cscope.
- Vim can be used to edit files using network protocols like SSH and HTTP.
- Vim includes multilevel undo/redo.
- Vim allows the screen to be split for editing multiple files.
- Vim can edit files inside a compressed archive (gzip, zip, tar, etc).
- Vim includes a built in diff for comparing files (vimdiff).
- Vim includes support for plugins, and finer control over config and startup files.
- Vim can be scripted with vimscript, or with an external scripting language (e.g. python, perl, shell).

(c) Introduction to command line arguments and logical statements in vi editor

```
x=5 y=10
echo $((x+$y))
Output: 15
echo $((x-$y))
Output: -5
echo $((x*$y)) Output: 50
echo $((y/$x)) Output: 2
```

If **Condition** is used for decision making in shell script, if given condition is true then command1 is executed.

Syntax

```

if condition then
command
fi
Example
j=9 i=10
count=0
if [ $((j+1)) -eq $i ] then count=$((count+1))
echo $i
fi
Output- 10

```

If elif condition is used for decision making in shell script, If given condition is true then command1 is executed, otherwise control goes to else if section to check whether it is correct or not.

Syntax

```

if condition then
command1
elif condition then
command2
fi

```

Example

```

if [ $marks -gt 90 ] then echo A
elif [ $marks -gt 80] then echo B
elif [ $marks -gt 70] then echo C
elif [ $marks -gt 60] then echo D
elif [ $marks -gt 50] then echo E
else echo F
fi
Input
8
Output
F

```

Looping Statements

1. For loop

```

for { variable name } in { list }
do
execute one for each item in the list until the list is not finished (And repeat all statement
between do and done)
done

```

Example

```

for j in $(seq 2 1 5) do Echo $j
Done

```

```

Output
2
3
4
5

```

2. While loop

Syntax

```

while [ condition ] do command1 command2
Done

```

Example:

```

number=5
while [ $number -ne 0] do
Echo $number
done

```

Output

5
4
3
2
1

PRACTICAL - 9

AIM: To implement the following programs in vim editor

1. Check whether a given year is leap year or not.
2. Check whether a given number is odd or even.
3. Calculate the gross salary of an employee.
4. Input marks. Calculate the average marks.
5. To find the circumference and area of a circle.

1. Check whether a given year is leap year or not.

```
echo "enter the year"
read y
a=$((y%4))
b=$((y%100))
c=$((y%400))
if [ $( $a -eq 0 ) -a $( $b -ne 0 ) -o $( $c -eq 0 ) ]
then
echo "leap year"
else
echo "not a leap year"
fi
```

2. Check whether a given number is odd or even.

```
echo "enter the number"
read a
b=$((a%2))
if [ $b -eq 0 ]
then
echo "$a is even"
else
echo "$a is odd"
fi
```

3. Calculate the gross salary of an employee.

```
echo "enter salary of 5 employees"
read a
read b
read c
read d
read e
s=$((a+b+c+d+e))
avg=$((s/5))
echo "Gross salary is $s "
echo " Average salary is $avg"
~
```

4. Input marks. Calculate the average marks.

```
echo " enter marks in 5 subjects "
read m1 m2 m3 m4 m5
echo $m1
echo $m2
echo $m3
echo $m4
echo $m5
s=$((m1+m2+m3+m4+m5))
echo "total marks= $s "
r=$((s/5))
if [ $r -ge 90 ]
then
echo grade-A
elif [ $r -ge 70 -a $r -lt 90 ]
then
echo grade-B
elif [ $r -ge 50 -a $r -lt 70 ]
then
echo grade-C
else
echo grade-F
fi
echo "percentage marks: $r %"
```

5. To find the Circumference and Area of a circle.

```
echo "enter the radius of circle"
read r
p=22/7
area=$((p*r*r))
cir=$((2*p*r))
echo "area of circle: $area"
echo "circumference of circle: $cir"
```

PRACTICAL - 10

AIM: To implement the following programs in vim editor

1. To find the perimeter and area of a rectangle.
2. To find the factorial of a number.
3. To find the sum of digits of a number.
4. To find the reverse of a number.
5. To print the Fibonacci series.

1. To find the perimeter and area of a rectangle.

```
echo "enter the length and breadth of rectangle"
read l b
area=$((l*b))
p=$((l+b))
per=$((2*p))
echo "area of rectangle: $area"
echo "perimeter of rectangle: $per"
~
```

2. To find the factorial of a number.

```
echo "enter the no. to find factorial"
read n
if [ $n -eq 0 ]
then
echo "factorial is 1"
else
i=1
sum=1
while [ $i -le $n ]
do
sum=$((sum*i))
i=$((i+1))
done
echo "factorial is $sum"
fi
~
```

3. To find the sum of digits of a number.

```
echo "Enter the number"
read n
sum=0
while [ $n -ne 0 ]
do
b=$((n%10))
sum=$((sum+b))
n=$((n/10))
done
echo "the sum of digits of $n is: $sum"
~
```

4. To find the reverse of a number.

```
echo "enter the number"
read n
sum=0
while [ $n -ne 0 ]
do
p=$((sum*10))
r=$((n%10))
sum=$((p+r))
n=$((n/10))
done
echo "the reverse of $n is : $sum"
~
```

5. To print the Fibonacci series.

```
echo "enter the number of terms in fibonacci series"
read n
if [ $n -eq 1 ]
then
echo "sum of fibonacci series: 0"
elif [ $n -eq 2 ]
then
echo "fibonacci series: 0 1"
echo " sum of fibonacci series: 1"
else
sum=1
i=1
echo "fibonacci series upto $n terms: 0"
echo "1"
a=0
b=1
m=$((n-2))
while [ $i -le $m ]
do
c=$((a+b))
echo "$c"
sum=$((sum+c))
i=$((i+1))
a=$b
b=$c
done
echo "sum of fibonacci series: $sum"
fi
```

PRACTICAL - 11

AIM: C++ program to implement the following scheduling algorithms

- 1) First Come First Serve (FCFS)
- 2) Shortest Job First (SJF) non-preemptive and preemptive
- 3) Priority non-preemptive and preemptive
- 4) Round Robin

1. FCFS

```
#include <iostream>
using namespace std;
struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
};
void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
        {
            if(p[j].at > p[j+1].at)
            {
                t = p[j];
                p[j] = p[j+1];
                p[j+1] = t;
            }
        }
}
void waiting(process* p[])
{
    int n = 4, t=0;
    for (int i = 1; i <= n; ++i)
    {
        p[i]->wt = p[i-1]->bt - p[i]->at;
        t += p[i]->bt;
        p[i]->tat = t - p[i]->at;
    }
}

int main()
{
    process p[] = {{0,0,0}, {1,0,7}, {2,2,4}, {3,4,1}, {4,5,4}};
    float awt = 0, atat = 0;
    sort(p);
    int n = 4, t=0;
    for (int i = 1; i <= n; ++i)
    {
        p[i].wt = t - p[i].at;
        t += p[i].bt;
        p[i].tat = t - p[i].at;
    }

    cout<<"P\tAT\tBT\tWT\tTAT"<<endl;
    for (int i = 1; i <= n; ++i)
    {
        cout<<p[i].pno<<"\t"<<p[i].at<<"\t"<<p[i].bt<<"\t"<<p[i].wt<<"\t"<<p[i].tat<<endl;
        awt += p[i].wt;
    }
}
```

```

        atat += p[i].tat;
    }
    cout<<endl;
    cout<<"\nAverage Waiting Time : "<<awt/n;
    cout<<"\nAverage Turn Around Time : "<<atat/n<<"\n";
    return 0;
}

```

OUTPUT:

P	AT	BT	WT	TAT
1	0	7	0	7
2	2	4	5	9
3	4	1	7	8
4	5	4	7	11

```

Average Waiting Time : 4.75
Average Turn Around Time : 8.75

-----
Process exited after 0.1016 seconds with return value 0
Press any key to continue . . .

```

2. NON-PREEMPTIVE SJF:

```

#include <iostream>
#include<iomanip>
using namespace std;
struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
    int tl;
};
void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
        {
            if(p[j].at > p[j+1].at)
            {
                t = p[j];
                p[j] = p[j+1];
                p[j+1] = t;
            }
        }
}
int main()
{
    process a[] = {{1,0,7}, {2,2,4}, {3,4,1}, {4,5,4}};
    int i, n = 4, total = 0;
    float t = a[0].at, awt=0, atat=0;
    int k=0, index, count = 1;
    cout<<"\nNumber of processes to be entered: ";
    cin>>n;
    sort(a);
    for (i = 0; i < 4; ++i)

```

```

{
    a[i].tl = a[i].bt;
    total += a[i].bt;
}

int gantt[total];
cout<<"\n";

for (i = 0; i < total; ++i)
{
    int j = 0, small = 1000;
    while(j < 4)
    {
        if( (a[j].at <= t) && (a[j].tl != 0) )
        {
            if(a[j].tl < small)
            {
                index = j;
                small = a[j].tl;
            }
        }
        j++;
    }
    t++;
    gantt[i] = a[index].pno;
    a[index].tl--;
    a[index].wt = i;
    if( (gantt[i] != gantt[i-1]) && i!=0)
        count++;

}
for(i = 0; i < n; i++)
{
    a[i].tat = a[i].wt + 1 - a[i].at;
    a[i].wt = (a[i].tat - a[i].bt);
    awt += a[i].wt;
    atat += a[i].tat;
}

awt = awt/n;
atat = atat/n;

cout<<"P\tAT\tBT\tWT\tTAT"<<endl;
for (i = 0; i < n; ++i)
    cout<<a[i].pno<<"\t"<<a[i].at<<"\t"<<a[i].bt<<"\t"<<a[i].wt<<"\t"<<a[i].tat<<endl;
cout<<"\nAverage Waiting Time : "<<awt;
cout<<"\nAverage Turn Around Time : "<<atat<<"\n";
}

```

OUTPUT:

P	AT	BT	WT	TAT
1	0	7	0	7
2	2	4	5	9
3	4	1	7	8
4	5	4	7	11

Average Waiting Time : 4.75
Average Turn Around Time : 8.75

Process exited after 0.1016 seconds with return value 0
Press any key to continue . . .

3. PREEMPTIVE SJF

```
#include <iostream>
#include<iomanip>
using namespace std;
struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
    int tl;
};
void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
        {
            if(p[j].at > p[j+1].at)
            {
                t = p[j];
                p[j] = p[j+1];
                p[j+1] = t;
            }
        }
}
int main()
{
    process a[] = {{1,0,7,1}, {2,2,4,2}, {3,4,1,3}, {4,5,4,2}};
    int i, n = 4, total = 0, count = 1;
    float t = a[0].at, awt=0, atat=0;
    int k=0, index;
    cout<<"\n";
    sort(a);
    for (i = 0; i < n; ++i)
    {
        a[i].tl = a[i].bt;
        total += a[i].bt;
    }
    int gantt[n], time[n+1];
    time[0] = 0;
    for (i = 0; i < n; ++i)
    {
        int j = 0, small = 1000;
        while(j < n)
        {
            if( (a[j].at <= t) && (a[j].tl != 0) )
            {
                if(a[j].bt < small)
                {
                    index = j;
                    small = a[j].bt;
                }
            }
            j++;
        }
        a[index].wt = t;
        t += a[index].bt;
        gantt[i] = a[index].pno;
        a[index].tl = 0;
        time[i+1] = t;
    }
}
```

```

for(i = 0; i < n; i++)
{
    a[i].tat = a[i].wt + a[i].bt - a[i].at;
    a[i].wt = (a[i].wt - a[i].at);
    awt += a[i].wt;
    atat += a[i].tat;
}
cout<<"P\tAT\tBT\tWT\tTAT"=><endl;
for (i = 0; i < n; ++i)
{
    cout<<a[i].pno<<"\t"=><a[i].at<<"\t"=><a[i].bt<<"\t"=><a[i].wt<<"\t"=><a[i].tat<<endl;
}
cout<<"\nAverage Waiting Time : "<<(awt/n);
cout<<"\nAverage Turn Around Time : "<<(atat/n)<<"\n\n";
}

```

OUTPUT:

P	AT	BT	WT	TAT
1	0	7	0	7
2	2	4	5	9
3	4	1	7	8
4	5	4	7	11

Average Waiting Time : 4.75
Average Turn Around Time : 8.75

Process exited after 0.1016 seconds with return value 0
Press any key to continue . . .

4. NON-PREEMPTIVE PRIORITY

```

#include <iostream>
#include<iomanip>

using namespace std;

struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
    int tl;
};

void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
        {
            if(p[j].at > p[j+1].at)
            {
                t = p[j];
                p[j] = p[j+1];
                p[j+1] = t;
            }
        }
}

```

```

        }
    }

int main()
{
    process a[] = {[1,0,7,1], {2,2,4,2}, {3,4,1,3}, {4,5,4,2}};
    int i, n = 4, total = 0, count = 1;
    float t = a[0].at, awt=0, atat=0;
    int k=0, index;
    cout<<"\n";

    sort(a);

    for (i = 0; i < n; ++i)
    {
        a[i].tl = a[i].bt;
        total += a[i].bt;
    }

    int gantt[n], time[n+1];
    time[0] = 0;
    for (i = 0; i < n; ++i)
    {
        int j = 0, small = 1000;
        while(j < n)
        {
            if( (a[j].at <= t) && (a[j].tl != 0) )
            {

                if(a[j].priority < small)
                {
                    index = j;
                    small = a[j].priority;
                }
            }
            j++;
        }
        a[index].wt = t;
        t += a[index].bt;
        gantt[i] = a[index].pno;
        a[index].tl = 0;
        time[i+1] = t;
    }

    for(i = 0; i < n; i++)
    {
        a[i].tat = a[i].wt + a[i].bt - a[i].at;
        a[i].wt = (a[i].wt - a[i].at);

        awt += a[i].wt;
        atat += a[i].tat;
    }

    cout<<"P\tAT\tBT\tWT\tTAT"=><endl;
    for (i = 0; i < n; ++i)
    cout<<a[i].pno<<"\t"<<a[i].at<<"\t"<<a[i].bt<<"\t"<<a[i].wt<<"\t"<<a[i].tat<<endl;

    cout<<endl<<"Gantt chart: \n";
    for (i = 0; i < n; ++i)
        cout<<"=====";
    cout<<endl<<"| ";
    for (i = 0; i < n; ++i)
    {
        cout<<"P"<<gantt[i]<<" | ";
        while(gantt[i] == gantt[i+1])
            i++;
    }
}

```

```

    }
    cout<<endl<<"";
    for (i = 0; i < n; ++i)
        cout<<"======">> endl;
    cout<<endl;
    for (i = 0; i <= n; ++i)
        cout<<time[i]<<setw(9);
    cout<<endl;
    cout<<"\nAverage Waiting Time : "<<(awt/n);
    cout<<"\nAverage Turn Around Time : "<<(atat/n)<<"\n\n";
}
}

```

OUTPUT:

P	AT	BT	WT	TAT
1	0	7	0	7
2	2	4	5	9
3	4	1	11	12
4	5	4	6	10
0	7	11	15	16
Average Waiting Time : 5.5				
Average Turn Around Time : 9.5				
<hr/> Process exited after 0.04961 seconds with return value 0 Press any key to continue . . .				

5. PREEMPTIVE PRIORITY:

```

#include <iostream>
#include<iomanip>

using namespace std;

struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
    int tl; //time-left
};

//sort on the basis of arrival time
void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
        {
            if(p[j].at > p[j+1].at)
            {
                t = p[j];
                p[j] = p[j+1];
                p[j+1] = t;
            }
        }
}

```

```

        p[j+1] = t;
    }
}
int main()
{
    process a[] = {[1,0,7,1], {2,2,4,2}, {3,4,1,3}, {4,5,4,2}};
    int i, n = 4, total = 0, count = 1;
    float t = a[0].at, awt=0, atat=0;
    int k=0, index;
    cout<<"\n";
    sort(a);
    for (i = 0; i < n; ++i)
    {
        a[i].tl = a[i].bt;
        total += a[i].bt;
    }
    int gantt[total];
    for (i = 0; i < total; ++i)
    {
        int j = 0, small = 1000;
        while(j < n)
        {
            if( (a[j].at <= t) && (a[j].tl != 0) )
                if(a[j].priority < small)
                    index = j;
                    small = a[j].priority;
            j++;
        }
        t++;
        gantt[i] = a[index].pno;
        a[index].tl--;
        a[index].wt = i;

        if( (gantt[i] != gantt[i-1]) && i!=0)
            count++;
    }
    for(i = 0; i < n; i++)
    {
        a[i].tat = a[i].wt + 1 - a[i].at;
        a[i].wt = (a[i].wt - a[i].at - a[i].bt + 1);
        awt += a[i].wt;
        atat += a[i].tat;
    }
}

cout<<"P\tAT\tBT\tWT\tTAT"<<endl;
for (i = 0; i < n; ++i)
{
    cout<<a[i].pno<<"\t"<<a[i].at<<"\t"<<a[i].bt<<"\t"<<a[i].wt<<"\t"<<a[i].tat<<endl;
}

cout<<"\nAverage Waiting Time : "<<(awt/n);
cout<<"\nAverage Turn Around Time : "<<(atat/n)<<"\n";

```

```

cout<<endl<<"Ganttt chart: \n";

for (i = 0; i < count; ++i)
    cout<<"=====";
    cout<<endl<<"|  ";
for (i = 0; i < total; ++i)
{
    cout<<"P"<<gantt[i]<<"|  ";
}

```

```

        while(gantt[i] == gantt[i+1])
            i++;
    }

    cout<<endl<<"";
    for (i = 0; i < count; ++i)
        cout<<"======">>>>
    cout<<endl;

    for (i = 0; i < total; ++i)
    {
        cout<<i<<setw(9);
        while(gantt[i] == gantt[i+1])
            i++;
    }
    cout<<i<<endl;
}

```

OUTPUT:

```

P      AT      BT      WT      TAT
1      0       7       0       7
2      2       4       5       9
3      4       1       11      12
4      5       4       6       10

0      7       11      15      16

Average Waiting Time : 5.5
Average Turn Around Time : 9.5

-----
Process exited after 0.1065 seconds with return value 0
Press any key to continue . . .

```

4. ROUND-ROBIN

```

#include <iostream>
#include<iomanip>

using namespace std;

struct process
{
    int pno;
    int at;
    int bt;
    int priority;
    int wt;
    int tat;
    int tl;
};

void sort(process p[])
{
    int n = 4;
    process t;
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1-i; j++)
            if(p[j].at > p[j+1].at)

```

```

    {
        t = p[j];
        p[j] = p[j+1];
        p[j+1] = t;
    }
}

int main()
{
    process a[] = {{1,0,7,1}, {2,2,4,2}, {3,4,1,3}, {4,5,4,2}};
    int i, n = 4, total = 0;
    float t = a[0].at, awt=0, atat=0;
    int q, k=0, index;
    cout<<"\n";
    cout<<"Quantum: ";
    cin>>q;
    sort(a);
    for (i = 0; i < 4; ++i)
    {
        a[i].tl = a[i].bt;
        total += a[i].bt;
    }
    int gantt[total], time[total];
    time[0] = 0;

    int j = 0, count = 0;
    for (i = 0; i < total; i += q)
    {
        while(j < n)
        {
            if( (a[j].at <= t) && (a[j].tl != 0) )
            {
                index = j;
                j = (j+1)%n;
                break;
            }
            j = (j+1)%n;
        }
        if(a[index].tl < q)
        {
            t += (a[index].tl);
            i = i - (q- a[index].tl); // i = i - 3 + 1 + 3
            a[index].tl = 0;
        }
        else
        {
            t += q;
            a[index].tl -= q;
        }
        a[index].wt = i;
        time[count + 1] = t;
        gantt[count++] = a[index].pno;
    }
    for(i = 0; i < n; i++)
    {
        a[i].tat = a[i].wt + q - a[i].at;
        a[i].wt = (a[i].wt - a[i].at - a[i].bt + q);
        awt += a[i].wt;
        atat += a[i].tat;
    }
    cout<<"\nP\tAT\tBT\tWT\tTAT"<<endl;
    for (i = 0; i < n; ++i)
        cout<<a[i].pno<<"\t" <<a[i].at<<"\t" <<a[i].bt<<"\t" <<a[i].wt<<"\t" <<a[i].tat<<endl;
    cout<<"\nAverage Waiting Time : "<<(awt/n);
    cout<<"\nAverage Turn Around Time : "<<(atat/n)<<"\n";
    for (i = 0; i <= count; ++i)
        cout<<time[i]<<setw(9);
}

```

OUTPUT:

```
Quantum: 2
P      AT      BT      WT      TAT
1      0       7       9       16
2      2       4       5       9
3      4       1       0       1
4      5       4       4       8

Average Waiting Time      : 4.5
Average Turn Around Time : 8.5
0          2          4          5          7          9          11         13         15         16

-----
Process exited after 11.41 seconds with return value 0
Press any key to continue . . .
```

PRACTICAL - 12

AIM: To write C++ program to implement the following banker's algorithm:

- a) Safety Algorithm
- b) Request and Resource Allocation Algorithm

PROGRAM:

```
# Written in Python 3 by Shubhkirti Sharma
# safetyAlgorithm.py
import numpy
available = [3, 4, 2]
process = ['P0', 'P1', 'P2', 'P3', 'P4']
allocated = [
    [0, 1, 0],
    [2, 0, 0],
    [3, 0, 2],
    [2, 1, 1],
    [0, 0, 2]
]
max = [
    [7, 5, 3],
    [3, 2, 2],
    [9, 0, 2],
    [2, 2, 2],
    [4, 3, 3]
]
allocated = numpy.array(allocated)
max = numpy.array(max)
available = numpy.array(available)
need = numpy.array(max - allocated)
request = numpy.array([0, 0, 0])
var = numpy.array([1, 1, 1])
finish = [False, False, False, False, False]
job = []
work = numpy.array(available)
def safeSequence(work):
    for i in range(0, 5):
        for j in range(0, 5):
            if finish[j] is False and all(numpy.greater(work, need[j])):
                print('-----')
                print(j, 'being added to queue')
                print('Available: ', work)
                print('Need: ', need[j])
                print(j, ' completed successfully!')
                work = work + allocated[j]
                print('Updated Available: ', work)
                finish[j] = True
                job.append(j)
                break
def main():
    safeSequence(work)
    print('-----')
    print(job)
    return
main()
```

```
# Written in Python 3 by Shubhkirti Sharma
# requestResource.py
import numpy
process = ['P0', 'P1', 'P2', 'P3', 'P4']
available = numpy.array([3, 3, 2])
allocated = numpy.array([
    [0, 1, 0],
```

```
[2, 0, 0],  
[3, 0, 2],  
[2, 1, 1],  
[0, 0, 2]  
])  
max = numpy.array([  
    [7, 5, 3],  
    [3, 2, 2],  
    [9, 0, 2],  
    [2, 2, 2],  
    [4, 3, 3]  
])  
need = numpy.array(max - allocated)  
request = numpy.array([0, 0, 0])  
var = numpy.array([1, 1, 1])  
def requestResource(request, allocated, max, available, need, i):  
    if all(numpy.greater(need[i]+var, request)):  
        if all(numpy.greater(available+var, request)):  
            allocated[i] = allocated[i] + request  
            need[i] = [0, 0, 0]  
            available = available - request  
            print('Available Resources: ', available)  
            print('Request Granted!')  
        else:  
            print('Not enough resources!')  
    else:  
        print('Request exceeds the need amount!')  
def main():  
    i = 1  
    request = [1, 0, 2]  
    requestResource(request, allocated, max, available, need, i)  
    return  
main()
```

PRACTICAL - 13

AIM: To write C++ program to implement the following Process synchronization algorithms with semaphores:

1. Producer - Consumer problem
2. Reader - Writers problem
3. Dining - Philosopher problem

1. PRODUCER CONSUMER PROBLEM

```
#include<iostream>
#include<cstdlib>
using namespace std;
class BoundedBuffer
{
    int full, empty, mutex, flagp, flagc; //flag tells if the process is waiting
public:
    BoundedBuffer(int n)
    {
        full = 0;
        empty = n;
        mutex = 1;
        flagp = flagc = 0;
    }
    int wait(int S, int *flag)
    {
        if(S - 1 < 0)
        {
            *flag = 1;
            cout<<"\nWAIT\n";
            return -1;
        }
        else
            return --S;
    }
    int signal(int S)
    {
        return ++S;
    }
    void producer()
    {
        cout<<"\nItem produced by producer";
        if (wait(empty, &flagp) == -1)
            return;
        empty = wait(empty, &flagp);
        mutex = wait(mutex, &flagp);
        cout<<"\nItem is added to buffer";
        cout<<"\nEmpty buffers: "<<empty;
        mutex = signal(mutex);
        full = signal(full);
        if(flagc == 1)
        {
            cout<<"\n\nConsumer was waiting\n";
            consumer();
            flagc = 0;
        }
    }
    void consumer()
    {
        if (wait(full, &flagc) == -1)
            return;
        full = wait(full, &flagc);
        mutex = wait(mutex, &flagc);
        cout<<"\nItem is removed from buffer";
        cout<<"\nFull buffers: "<<full;
    }
}
```

```

cout<<"\nItem is consumed by consumer";
    mutex = signal(mutex);
    empty = signal(empty);
    if(flagp == 1)
    {
        cout<<"\n\nProducer was waiting\n";
        producer();
        flagc = 0;
    }
}

};

int main()
{
    int n;
    cout << "Enter the number of buffers ";
    cin >> n;
    BoundedBuffer b(n);
    char ch;
    int r;
    do
    {
        cout << "\n1. Produce an item\n2.Consume an item\n3.Exit";
        cout << "\nEnter your choice: ";
        cin >> r;
        switch(r)
        {
            case 1:
                b.producer(); break;

            case 2:
                b.consumer(); break;
            case 3:
                exit(0);
        }
        cout << "\nDo you want to continue? ";
        cin >> ch;
    }while(ch == 'y');
}

```

2. READERS - WRITERS PROBLEM

```

#include<iostream>
#include<cstdlib>
using namespace std;
class ReadersWriters
{
    int wrt, mutex;
    int readcount, flagr, flagw;
public:
    ReadersWriters()
    {
        wrt = 1;
        mutex = 1;
        readcount = 0;
        flagr = flagw = 0;
    }
    int wait(int S, int *flag)
    {
        if(S - 1 < 0)
        {
            *flag = 1;
            cout<<"\nWAIT\n";
            return -1;
        }
        else
            return --S;
    }
}

```

```

    }
    int signal(int S)
    {
        return ++S;
    }
    void readersEntry()
    {

        mutex = wait(mutex, &flagr);
        readcount++;
        if(readcount == 1)
        {
            if (wait(wrt, &flagr) == -1)
                return;
            wrt = wait(wrt, &flagr);
        }
        mutex = signal(mutex);
        cout<<"\nReader is reading now";
        cout<<"\nReaders at this moment: "<<readcount;
    }
    void readersExit()
    {
        mutex = wait(mutex, &flagr);
        if(readcount == 0)
        {
            cout<<"0 readers in critical section";
            return;
        }
        readcount--;
        cout<<"Readers at this moment: "<<readcount;
        if (readcount == 0)
        {
            wrt = signal(wrt);
            if(flagw == 1)
            {
                cout<<"\n\nReading process is done. Writer was waiting earlier.\n";
                writers();
                flagw = 0;
            }
        }
        mutex = signal(mutex);
    }
    void writers()
    {
        if (wait(wrt, &flagw) == -1)
            return;
        wrt = wait(wrt, &flagw);
        cout<<"\nwrt is "<<wrt;
        cout<<"\nWriter is writing now";
        wrt = signal(wrt);
        if(flagr == 1)
        {
            cout<<"\n\nWriting process is done. Reader was waiting earlier.\n";
            readersEntry();
            flagw = 0;
        }
    }
};

int main()
{
    ReadersWriters b;
    char ch;
    int r;
    do
    {
        cout << "\n1. Begin writing\n2. Begin reading\n3. End reading\n4. Exit";
        cout << "\nEnter your choice: ";
        cin >> r;

```

```

        switch(r)
        {
            case 1:
                b.writers(); break;
            case 2:
                b.readersEntry(); break;
            case 3:
                b.readersExit(); break;
            case 4:
                exit(0);
        }
        cout << "\nDo you want to continue? ";
        cin >> ch;

    }while(ch == 'y');
}

```

3. DINING PHILOSOPHER PROBLEM

```

#include<iostream>
#include<cstdlib>
using namespace std;
class DiningPhilosopher
{
    int chopstick[5], flag[5], p[5];
public:
DiningPhilosopher()
{
    for (int i = 0; i < 5; ++i)
    {
        chopstick[i] = 1;
        flag[i] = 0;
        p[i] = 0;
    }
}
int wait(int S, int i)
{
    if(S - 1 < 0)
    {
        flag[i] = 1;
        cout<<"\nWAIT\n";
        return -1;
    }
    else
        return --S;
}
int signal(int S)
{
    return ++S;
}
void eat(int i)
{
    if (wait(chopstick[i], i) == -1)
        return;
    if (wait(chopstick[(i+1)%5], i ) == -1)
        return;
    chopstick[i] = wait(chopstick[i], i);
    chopstick[(i+1)%5] = wait(chopstick[(i+1)%5], i);
    cout<<"\n---EAT---";
    p[i] = 1;
    cout << endl;
    for (int j = 0; j < 5; ++j)
        if(p[j] == 1)
            cout << "P" << j+1 << " ";
}
void think(int i)
{

```

```
chopstick[i] = signal(chopstick[i]);
chopstick[ (i+1) % 5 ] = signal(chopstick[(i+1) % 5]);
p[i] = 0;
cout<<"\n----THINKING---";

for (int j = 0; j < 5; ++j)
    if(flag[j] == 1)
    {
        cout<<"\n\nPhilosopher "<< j+1 << " was waiting\n";
        flag[j] = 0;
        eat(j);
    }
}

};

int main()
{
    DiningPhilosopher b;
    int r,i;
    do
    {
        cout << "\n1. Eat\n2.Think\n3.Exit";
        cout << "\nEnter your choice: ";
        cin >> r;
        cout << "\nChoose a philosopher: ";
        cin >> i;
        switch(r)
        {
            case 1:
                b.eat(i-1); break;

            case 2:
                b.think(i-1); break;

            case 3:
                exit(0);
        }
    }while(1);
}
```

PRACTICAL - 14

AIM: To write C++ program to implement the following Memory management algorithm:

1. MFT
2. MVT
 - o First fit
 - o Best fit
 - o Worst fit

1. MFT

```
#include<iostream>
using namespace std;

int block = 450;

void MFT(int n)
{
    int a[2*n], b[n], i;

    cout<<"\nMaximum space of this block is "<<block;
    cout<<"\nEnter the size of the blocks: ";

    for (i = 0; i < 2*n; ++i)
        a[i] = block;

    for (i = 0; i < n; ++i)
        cin >> b[i];

    i = 0;

    for (int k = 0; k < n; ++k)
    {
        if(a[i] - b[k] > 0)
            cout << "\nInternal Fragmentation of block"<<i+1;

        if((a[i] - b[k]) < 0)
        {
            cout<<"\nExternal fragmentation of block"<<i+1;
            b[k] = b[k] - a[i];
            a[i] = 0;
            a[++i] -= b[k];
        }

        else
            a[i] -= b[k];
        i++;
    }

    cout<<"\nSpace left"; //Space occupied";
    for (int j = 0; j < i; ++j)
        cout<<"\n"<<j+1<<". "<<a[j]<<endl;//<<"\t\t"<<b[j];
}

int main()
{
    int n;
    cout<<"-----MFT-----\n";
    cout<<"Enter the number of blocks to be entered ";
    cin>>n;
    MFT(n);
    return 0;
}
```

2. MVT

```
#include<iostream>
using namespace std;

struct page
{
    int no;
    int size;
};

void FirstFit(int n, page a[], page b[])
{
    int internal;
    cout << "\n-----FIRST FIT-----";
    int i = 0;

    for (int k = 0; k < n; ++k)
    {
        if(i == n)
        {
            cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
            break;
        }
        if((a[i].size - b[k].size) < 0)
        {
            if(i == n-1)
            {
                cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
                break;
            }

            cout<<"\nExternal fragmentation of block"<<b[k].no;
            b[k].size = b[k].size - a[i].size;
            a[i].size = 0;
            i++;
            a[+i].size -= b[k].size;
        }
        else
            a[i].size -= b[k].size;
        i++;
        cout<<"\nBlock "<<b[k].no<<" allocated.";
    }
}

//-----BEST FIT-----
void BestFit(int n, page a[], page b[])
{
    int internal = 0;

    cout << "\n-----BEST FIT-----";
    int i = 0;

    //----sorting of b----
    for (i = 0; i < n; ++i)
        for (int j = 0; j < n - i - 1; ++j)
            if(b[i].size > b[i+1].size)
            {
                page t = b[i];
                b[i] = b[i+1];
                b[i+1] = t;
            }

    i = 0;
    for (int k = 0; k < n; ++k)
```

```

{
    if(i == n)
    {
        cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
        break;
    }
    if(a[i].size - b[k].size > 0)
        internal += a[i].size - b[k].size;
    if((a[i].size - b[k].size) < 0)
    {
        if(i == n-1)
        {
            cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
            break;
        }
        cout<<"\nExternal fragmentation of block"<<b[k].no;
        b[k].size = b[k].size - a[i].size;
        a[i].size = 0;
        i++;
        //a[++i].size -= b[k].size;
    }
}
}

//-----WORST FIT-----
void WorstFit(int n, page a[], page b[])
{
    int internal = 0;

    cout << "\n-----WORST FIT-----";
    int i = 0;

    //----sorting of b----
    for (i = 0; i < n; ++i)
        for (int j = 0; j < n - i - 1; ++j)
            if(b[i].size < b[i+1].size)
            {
                page t = b[i];
                b[i] = b[i+1];
                b[i+1] = t;
            }

    i = 0;
    for (int k = 0; k < n; ++k)
    {
        if(i == n)
        {
            cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
            break;
        }

        if(a[i].size - b[k].size > 0)
        {
            internal += a[i].size - b[k].size;
        }
        if((a[i].size - b[k].size) < 0)
        {
            if(i == n-1)
            {
                cout<<"\nBlock "<<b[k].no<<" cannot be allocated. Memory full";
                break;
            }
            cout<<"\nExternal fragmentation of block"<<b[k].no;
            b[k].size = b[k].size - a[i].size;
            a[i].size = 0;
            i++;
            //a[++i].size -= b[k].size;
        }
    }
}

```

```
i++;
}
}
//-----MVT-----
void MVT()
{
    int n;
    cout<<"Enter the number of blocks you want to enter: ";
    cin>>n;
    page a[n], c[n], e[n];
    for (int i = 0; i < n; ++i)
    {
        cin >> a[i].size;
        a[i].no = i+1;
    }
    page b[n], d[n], f[n];
    for (int i = 0; i < n; ++i)
    {
        b[i].no = a[i].no;
        c[i].size = a[i].size;
        c[i].no = a[i].no;
        cout<<"\nBlock "<<i+1<<" allocated";
    }
    cout<<"\n\nProgram has been executed. Therefore, all the blocks are empty now. Enter again\n";
    for (int i = 0; i < n; ++i)
    {
        cin >> b[i].size;
        d[i].size = b[i].size;
        d[i].no = b[i].no;
        f[i].no = b[i].no;
    }
    BestFit(n, a, b);
    cout<<endl;
    FirstFit(n, c, d);
    cout<<endl;
    WorstFit(n, e, f);
}

int main()
{
    MVT();
    cout<<endl;
}
```

PRACTICAL - 15

AIM: To write C++ program to implement the following Page fault algorithm:

1. FIFO Page Replacement
2. Optimal Page Replacement
3. Least Recently Used (LRU)

1. FIFO Page Replacement

```
#include<iostream>
using namespace std;
#define RANGE 10
int c[3] = {-1,-1,-1};
struct page_array
{
    int page;
    bool valid;
};

void FCFS(page_array *p, int n)
{
    int j, page_faults = 0;
    for (int i = 0; i < n; ++i)
    {
        if(p[i].valid == false)
            cout<<"\nPage is invalid as its valid bit is OFF.";
        else
        {
            for (j = 0; j < 3; ++j)
            {
                if(p[i].page == c[j])
                    break;
            }
            if(j == 3)
            {
                for(int k = 0; k < 2; k++)
                    c[k] = c[k+1];

                c[2] = p[i].page;
                page_faults++;
            }
            cout << endl;
            for(int k = 0; k < 3; k++)
                cout << c[k]<< " ";
        }
    }
    cout << "\n" << page_faults << " page faults occurred in FIFO method" << endl;
}
int main()
{
    int n;
    cout << "Enter the number of pages you want to load from memory: ";
    cin >> n;
    page_array *p = new page_array [n];
    cout << "\nEnter the page values ";
    for (int i = 0; i < n; ++i)
        cin >> p[i].page;
    for (int i = 0; i < n; ++i)
    {
        if( (p[i].page >= 0) && (p[i].page <= RANGE) )
            p[i].valid = true;
        else
            p[i].valid = false;
    }
    FCFS(p, n);
}
```

```
    return 0;
}
```

1. Optimal Page Replacement

```
#include<stdio.h>

int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k,
pos, max, faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);
    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);
    printf("Enter page reference string: ");
    for(i = 0; i < no_of_pages; ++i){
        scanf("%d", &pages[i]);
    }
    for(i = 0; i < no_of_frames; ++i){
        frames[i] = -1;
    }
    for(i = 0; i < no_of_pages; ++i){
        flag1 = flag2 = 0;
        for(j = 0; j < no_of_frames; ++j){
            if(frames[j] == pages[i]){
                flag1 = flag2 = 1;
                break;
            }
        }
        //PAGE FAULT
        if(flag1 == 0){
            for(j = 0; j < no_of_frames; ++j){
                if(frames[j] == -1){
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }
        if(flag2 == 0){
            flag3 = 0;
            for(j = 0; j < no_of_frames; ++j)
            {
                temp[j] = -1;
                for(k = i + 1; k < no_of_pages; ++k)
                    if(frames[j] == pages[k])
                {
                    temp[j] = k;
                    break;
                }
            }
            for(j = 0; j < no_of_frames; ++j){
                if(temp[j] == -1){
                    pos = j;
                    flag3 = 1;
                    break;
                }
            }
            if(flag3 == 0){
                max = temp[0];
                pos = 0;
                for(j = 1; j < no_of_frames; ++j){
                    if(temp[j] > max){
                        max = temp[j];
                    }
                }
            }
        }
    }
}
```

```

        pos = j;
    }
}
}
frames[pos] = pages[i];
faults++;
}
printf("\n");
for(j = 0; j < no_of_frames; ++j){
printf("%d\t", frames[j]);
}
}
printf("\n\nTotal Page Faults = %d", faults);

return 0;
}

```

2. Least Recently Used

```

#include<stdio.h>

int findLRU(int time[], int n){
    int i, minimum = time[0], pos = 0;

    for(i = 1; i < n; ++i){
        if(time[i] < minimum){
            minimum = time[i];
            pos = i;
        }
    }

    return pos;
}

int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j,
pos, faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);

    printf("Enter reference string: ");

    for(i = 0; i < no_of_pages; ++i){
        scanf("%d", &pages[i]);
    }

    for(i = 0; i < no_of_frames; ++i){
        frames[i] = -1;
    }
    for(i = 0; i < no_of_pages; ++i){
        flag1 = flag2 = 0;

        for(j = 0; j < no_of_frames; ++j){
            if(frames[j] == pages[i]){
                counter++;
                time[j] = counter;
                flag1 = flag2 = 1;
                break;
            }
        }

        if(flag1 == 0){

```

```
for(j = 0; j < no_of_frames; ++j){  
    if(frames[j] == -1){  
        counter++;  
        faults++;  
        frames[j] = pages[i];  
        time[j] = counter;  
        flag2 = 1;  
        break;  
    }  
}  
}  
  
if(flag2 == 0){  
    pos = findLRU(time, no_of_frames);  
    counter++;  
    faults++;  
    frames[pos] = pages[i];  
    time[pos] = counter;  
}  
  
printf("\n");  
for(j = 0; j < no_of_frames; ++j){  
    printf("%d\t", frames[j]);  
}  
}  
printf("\n\nTotal Page Faults = %d", faults);  
return 0;  
}
```

PRACTICAL - 16

AIM:

To write C++ program to implement the following Disk scheduling algorithm:

1. FCFS
2. Shortest Seek Time First (SSTF)
3. Scan
4. C-scan
5. Look
6. C-Look

1. FCFS

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int q[num];

    int front = 0, rear = -1;
    cout<<"Input requests - ";
    for(int i = 0; i < num; i++)
        cin>>q[++rear];
    cout<<"Input initial position of head - ";
    int init;
    cin>>init;
    int total = 0;
    while(front <= rear)
    {
        total += abs(init - q[front]);
        init = q[front++];
    }
    cout<<"Total head movement - "<<total;
}
```

2. Shortest Seek Time First (SSTF)

```
#include<bits/stdc++.h>
using namespace std;
int head = 0;
bool compare(int a, int b)
{
    if(abs(head-a) < abs(head-b))
        return true;
    return false;
}

int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int arr[num];

    cout<<"Input processes - ";
    for(int i = 0; i < num; i++)
        cin>>arr[i];

    cout<<"Input starting position of head - ";
```

```

cin>>head;

int movement = 0;

for(int i = 0; i < num; i++)
{
    sort(arr+i, arr+num, compare);
    movement += (abs(head-arr[i]));
    head = arr[i];
}
cout<<"Total movement - "<<movement;
}

```

3. SCAN

```

#include<bits/stdc++.h>
using namespace std;

int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int arr[num], min = 9999, max = -9999;

    cout<<"Input processes - ";
    for(int i = 0; i < num; i++)
    {
        cin>>arr[i];
        if(arr[i] < min)
            min = arr[i];
        else if(arr[i] > max)
            max = arr[i];
    }

    int head;
    cout<<"Input starting position of head - ";
    cin>>head;

    int movement = head + max;

    cout<<"Total movement - "<<movement;
}

```

4. C-SCAN

```

#include<bits/stdc++.h>
using namespace std;

int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int arr[num];

    cout<<"Input processes - ";
    for(int i = 0; i < num; i++)
        cin>>arr[i];

    int head;
    cout<<"Input starting position of head - ";
    cin>>head;

    int max;
    cout<<"Input maximum number of blocks - ";

```

```

    cin>>max;

    int cur = 0;
    for(int i = 0; i < num; i++)
    {
        if(arr[i] > cur && arr[i] <= head)
            cur = arr[i];
    }

    int movement = 2 * max - head + cur;
    cout<<"Total movement - "<<movement;
}

```

5. LOOK

```

#include<bits/stdc++.h>
using namespace std;

int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int arr[num], min = 9999, max = -9999;

    cout<<"Input processes - ";
    for(int i = 0; i < num; i++)
    {
        cin>>arr[i];
        if(arr[i] < min)
            min = arr[i];
        else if(arr[i] > max)
            max = arr[i];
    }

    int head;
    cout<<"Input starting position of head - ";
    cin>>head;

    int movement = max - min + head - min;
    cout<<"Total movement - "<<movement;
}

```

6. C-LOOK

```

#include<bits/stdc++.h>
using namespace std;

int main()
{
    int num;
    cout<<"Input number of requests - ";
    cin>>num;
    int arr[num], min = 9999, max = -9999;

    cout<<"Input processes - ";
    for(int i = 0; i < num; i++)
    {
        cin>>arr[i];
        if(arr[i] < min)
            min = arr[i];
        else if(arr[i] > max)
            max = arr[i];
    }

    int head;

```

```
cout<<"Input starting position of head - ";
cin>>head;

int cur = 0;
for(int i = 0; i < num; i++)
{
    if(arr[i] > cur && arr[i] <= head)
        cur = arr[i];
}

int movement = 2 * (max - min) - (head - cur);
cout<<"Total movement - "<<movement;
}
```