

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

**Batch: A2**

**Roll No.: 1911027**

**Experiment No. 01**

**Grade: AA / AB / BB / BC / CC / CD / DD**

**Signature of the Staff In-charge with date**

**TITLE: Exploring basic Commands of UNIX: Shell, Processes, Files**

**AIM:** To Explore basic commands for handling File system under Unix/Linux using shell scripts.(Creating groups, chown , chmod , directory name, tty , diff, umask).

**Expected Outcome of Experiment:**

**CO 1.** To introduce basic concepts and functions of operating systems.

**Books/ Journals/ Websites referred:**

1. Silberschatz A., Galvin P., Gagne G. “Operating Systems Principles”, Willey Eight edition.
2. Achyut S. Godbole , Atul Kahate “Operating Systems”, McGraw Hill Third Edition.
3. Sumitabha Das “ UNIX Concepts & Applications”, McGraw Hill Second Edition.

**Pre Lab/ Prior Concepts:**

An operating system (OS) is a resource manager. It takes the form of a set of software routines that allow users and application programs to access system resources (e.g. the CPU, memory, disks, modems, printers network cards etc.) in safe efficient and abstract way.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

- The operating system kernel is in direct control of the underlying hardware. The kernel provides low-level device, memory and processor management functions (e.g. dealing with interrupts from hardware devices, sharing the processor among multiple programs, allocating memory for programs etc.)
- Basic hardware-independent kernel services are exposed to higher-level programs through a library of system calls (e.g. services to create a file, begin execution of a program, or open a logical network connection to another computer).
- Application programs (e.g. word processors, spreadsheets) and system utility programs (simple but useful application programs that come with the operating system, e.g. programs which find text inside a group of files) make use of system calls. Applications and system utilities are launched using a shell (a textual command line interface) or a graphical user interface that provides direct user interaction.

Operating systems can be distinguished from one another by the system calls, system utilities and user interface they provide, as well as by the resource scheduling policies implemented by the kernel.

UNIX has been a popular OS for more than two decades because of its multi-user, multi-tasking environment, stability, portability and powerful networking capabilities.

Linux is a free open source UNIX OS for PCs.

Linux has all of the components of a typical OS :

- **Kernel**

The Linux kernel includes device driver support for a large number of PC hardware devices (graphics cards, network cards, hard disks etc.), advanced processor and memory management features, and support for many different types of file systems. In terms of the services that it provides to application programs and system utilities, the kernel implements most BSD and SYSV system calls, as well as the system calls described in the POSIX.1 specification.

The kernel (in raw binary form that is loaded directly into memory at system startup time) is typically found in the file `/boot/vmlinuz`, while the source files can usually be found in `/usr/src/linux`.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

- **Shells and GUIs**

Linux supports two forms of command input: through textual command line shells similar to those found on most UNIX systems (e.g. sh - the Bourne shell, bash - the Bourne again shell and csh - the C shell) and through graphical interfaces (GUIs) such as the KDE and GNOME window managers.

- **System Utilities**

Virtually every system utility that you would expect to find on standard implementations of UNIX has been ported to Linux. This includes commands such as ls, cp, grep, awk, sed, bc, wc, more, and so on. These system utilities are designed to be powerful tools that do a single task extremely well (e.g. grep finds text inside files while wc counts the number of words, lines and bytes inside a file). Users can often solve problems by interconnecting these tools instead of writing a large monolithic application program.

- **Application programs**

Linux distributions typically come with several useful application programs as standard. Examples include the emacs editor, xv (an image viewer), gcc (a C compiler), g++ (a C++ compiler), xfig (a drawing package), latex (a powerful typesetting language) and soffice (StarOffice, which is an MS-Office style clone that can read and write Word, Excel and PowerPoint files).

---

Description of Commands and options:

---

**Implementation details:**

**Category : File System**

1) Command : cat

Purpose : The cat (short for “concatenate”) command is one of the most frequently used commands in Linux/Unix-like operating systems. cat command allows us to create single or multiple files, view content of a file, concatenate files and redirect output in terminal or files.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Syntax:

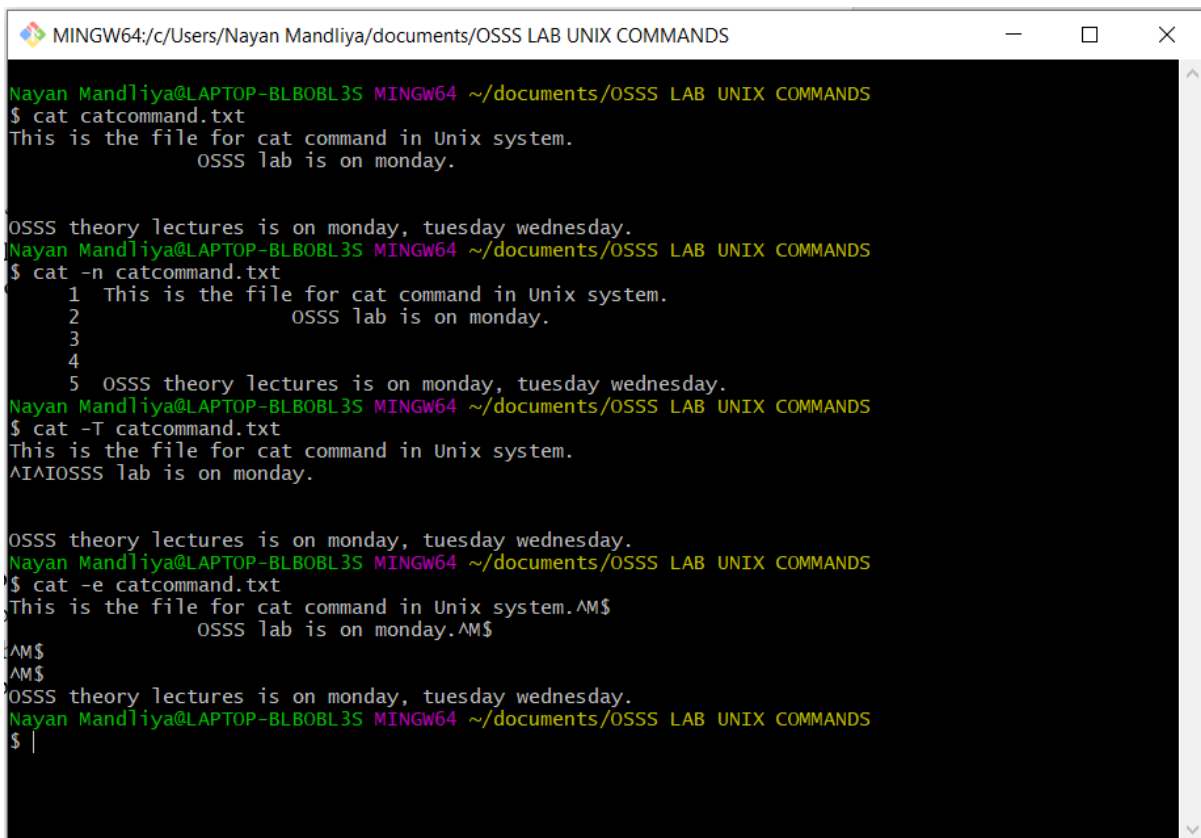
Cat [options] [file].....

Options:

Some options:

1. `-n` : It is used to display each line of a file with line numbers.
2. `-T` : It is used to show `^I` instead of a tab space.
3. `-e` : It shows `$` at the end of the line and also in space showing `$` if there is any gap between paragraphs. This option is useful to squeeze multiple lines into a single line.

Implementation :



```
MINGW64:/c:/Users/Nayan Mandliya/documents/OSSS LAB UNIX COMMANDS
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cat catcommand.txt
This is the file for cat command in Unix system.
    OSSS lab is on monday.

OSSS theory lectures is on monday, tuesday wednesday.
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cat -n catcommand.txt
 1 This is the file for cat command in Unix system.
 2     OSSS lab is on monday.
 3
 4
 5 OSSS theory lectures is on monday, tuesday wednesday.
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cat -T catcommand.txt
This is the file for cat command in Unix system.
^I^IOSSS lab is on monday.

OSSS theory lectures is on monday, tuesday wednesday.
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cat -e catcommand.txt
This is the file for cat command in Unix system.^M$
    OSSS lab is on monday.^M$
^M$
^M$
OSSS theory lectures is on monday, tuesday wednesday.
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ |
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

2) Command : cp

Purpose : cp stands for copy. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. cp command require at least two filenames in its arguments.

Syntax:

Cp [options] Source Destination

Options:

Some options:

1. -i(interactive) : i stands for Interactive copying. With this option system first warns the user before overwriting the destination file. cp prompts for a response, if you press y then it overwrites the file and with any other option leave it uncopied.
2. -b(backup) : With this option cp command creates the backup of the destination file in the same folder with the different name and in different format.
3. -r : Copying directory structure. With this option cp command shows its recursive behaviour by copying the entire directory structure recursively.

Implementation :

```
MINGW64:/c:/Users/Nayan Mandliya/documents/OSSS LAB UNIX COMMANDS
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
catcommand.txt  copycom.txt
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cp copycom.txt newcopy.txt
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
catcommand.txt  copycom.txt  newcopy.txt
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cp -i copycom.txt newcopy.txt
cp: overwrite 'newcopy.txt'? y
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
catcommand.txt  copycom.txt  newcopy.txt
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cp -b copycom.txt newcopyback.txt
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
catcommand.txt  copycom.txt  newcopy.txt  newcopyback.txt
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

```
MINGW64:/c:/Users/Nayan Mandliya/documents/OSSS LAB UNIX COMMANDS

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
Initial/ catcommand.txt copycom.txt newcopy.txt newcopyback.txt

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ cp -r Initial Copied

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
Copied/ Initial/ catcommand.txt copycom.txt newcopy.txt newcopyback.txt
```

### 3) Command : mkdir

Purpose : mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems ). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permissions to create a directory in the parent directory, or he/she may receive a 'permission denied' error.

Syntax:

mkdir [options...] [directories ...]

Options:

Some options:

1. -v(verbose) : It displays a message for every directory created.
2. -p : A flag which enables the command to create parent directories as necessary. If the directories exist, no error is specified.
3. -m : This option is used to set the file modes, i.e. permissions, etc. for the created directories. The syntax of the mode is the same as the chmod command.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ mkdir Initial

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
Initial/

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ mkdir -v Verbise
mkdir: created directory 'Verbise'

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
Initial/ Verbise/

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ mkdir -p GrandChild/Parent/Child

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
GrandChild/ Initial/ Verbise/

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ mkdir -m a=rwx AllPermissions
mkdir: cannot change permissions of 'AllPermissions': Permission denied

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ mkdir -m a=r ReadPermissions
mkdir: cannot change permissions of 'ReadPermissions': Permission denied
```

4) Command : rm

Purpose : rm stands for remove here. rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references (for example, a file with two different names). By default, it does not remove directories.

Syntax:

rm [OPTION]... FILE...



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Options:

Some options:

1. -I (interactive deletion) : Like in cp, the -i option makes the command ask the user for confirmation before removing each file, you have to press y for confirm deletion, any other key leaves the file un-deleted.
2. -r (recursive deletion) : With -r(or -R) option rm command performs a tree-walk and will delete all the files and sub-directories recursively of the parent directory. At each stage it deletes everything it finds. Normally, rm wouldn't delete the directories but when used with this option, it will delete.
3. -f (Force Deletion) : rm prompts for confirmation removal if a file is write protected. The -f option overrides this minor protection and removes the file forcefully.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
DeleteThis/  rmforce.txt  rminter.txt  rmnor.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ rm rmnor.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
DeleteThis/  rmforce.txt  rminter.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ rm -i rminter.txt
rm: remove regular file 'rminter.txt'? y
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
DeleteThis/  rmforce.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ rm -r DeleteThis
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
rmforce.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls -l
total 1
-r--r--r-- 1 Nayan Mandliya 197121 25 Aug 23 15:23 rmforce.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ rm rmforce.txt
rm: remove write-protected regular file 'rmforce.txt'? n
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ rm -f rmforce.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/documents/OSSS LAB UNIX COMMANDS
$ ls
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

5) Command : cmp

Purpose : cmp command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not. When cmp is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no difference is found i.e the files compared are identical. cmp displays no message and simply returns the prompt if the the files compared are identical.

Syntax:

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

Options:

Some options:

1. -b(print-bytes) : If you want cmp displays the differing bytes in the output when used with -b option.
2. -i [bytes-to-be-skipped] : Now, this option when used with cmp command helps to skip a particular number of initial bytes from both the files and then after skipping it compares the files. This can be done by specifying the number of bytes as argument to the -i command line option.
3. -l : This option makes the cmp command print byte position and byte value for all differing bytes.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cmp file1.txt file2.txt
file1.txt file2.txt differ: char 9, line 1

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cmp -b file1.txt file2.txt
file1.txt file2.txt differ: byte 9, line 1 is 146 f 163 s

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cmp -i 10 file1.txt file2.txt
file1.txt file2.txt differ: char 1, line 1

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cmp -l file1.txt file2.txt
 9 146 163
10 151 145
11 162 143
12 163 157
13 164 156
14 40 144
15 146 40
16 151 146
17 154 151
18 145 154
19 56 145
cmp: EOF on file1.txt after byte 19
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Processes

### 1) Command : ps

Purpose : Linux provides us a utility called ps for viewing information related with the processes on a system which stands as abbreviation for “Process Status”. ps command is used to list the currently running processes and their PIDs along with some other information depends on different options. It reads the process information from the virtual files in /proc file-system. /proc contains virtual files, this is the reason it’s referred as a virtual file system.

Syntax:

ps [options]

Options:

Some options:

1. -f : Show processes uids, ppids.
2. -s : Show process summary
3. -W : Show windows as well as Cygwin processes.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ps
  PID   PPID   PGID   WINPID  TTY      UID    STIME  COMMAND
  999     998     999    24944  pty0     197609 00:33:15 /usr/bin/bash
  998      1     998    26128  ?        197609 00:33:15 /usr/bin/mintty
 1121    999    1121    11480  pty0     197609 00:58:54 /usr/bin/ps

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ps -f
  UID    PID   PPID  TTY      STIME  COMMAND
Nayan Ma  999     998  pty0     00:33:15 /usr/bin/bash
Nayan Ma  998      1  ?        00:33:15 /usr/bin/mintty
Nayan Ma 1125    999  pty0     00:58:57 /usr/bin/ps

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ps -s
  PID  TTY      STIME  COMMAND
  999  pty0     00:33:15 /usr/bin/bash
  998  ?        00:33:15 /usr/bin/mintty
 1129  pty0     00:59:00 /usr/bin/ps
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ps -w
  PID   PPID   PGID   WINPID   TTY      UID    STIME COMMAND
  65540     0     0         4 ?         0    Aug 23 System
  65660     0     0        124 ?         0    Aug 23 Registry
  66240     0     0        704 ?         0    Aug 23 C:\Windows\Syste
m32\smss.exe
  66460     0     0        924 ?         0    Aug 23 C:\Windows\Syste
m32\csrss.exe
  65620     0     0         84 ?         0    Aug 23 C:\Windows\Syste
m32\wininit.exe
  66080     0     0        544 ?         0    Aug 23 C:\Windows\Syste
m32\services.exe
```

## 2) Command : kill

Purpose : kill command in unix (located in /bin/kill), is a built-in command which is used to terminate processes manually. kill command sends a signal to a process which terminates the process. If the user doesn't specify any signal which is to be sent along with kill command then default TERM signal is sent that terminates the process.

Syntax:

kill [options]

Options:

Some options:

1. -l : List the signal names. If arguments follow -l they are assumed to be signal numbers for which names should be listed.
2. -s : To show how to send signal to processes.
3. -L : This command is used to list available signals in a table format.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGEMT     8) SIGFPE      9) SIGKILL     10) SIGBUS
11) SIGSEGV     12) SIGSYS    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGURG      17) SIGSTOP   18) SIGTSTP    19) SIGCONT     20) SIGCHLD
21) SIGTTIN     22) SIGTTOU   23) SIGIO      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF   28) SIGWINCH   29) SIGPWR      30) SIGUSR1
31) SIGUSR2     32) SIGRTMIN  33) SIGRTMIN+1 34) SIGRTMIN+2 35) SIGRTMIN+3
36) SIGRTMIN+4 37) SIGRTMIN+5 38) SIGRTMIN+6 39) SIGRTMIN+7 40) SIGRTMIN+8
41) SIGRTMIN+9 42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMIN+16 49) SIGRTMAX-15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9
56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4
61) SIGRTMAX-3  62) SIGRTMAX-2  63) SIGRTMAX-1  64) SIGRTMAX

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ kill -L
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGEMT     8) SIGFPE      9) SIGKILL     10) SIGBUS
11) SIGSEGV     12) SIGSYS    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGURG      17) SIGSTOP   18) SIGTSTP    19) SIGCONT     20) SIGCHLD
21) SIGTTIN     22) SIGTTOU   23) SIGIO      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF   28) SIGWINCH   29) SIGPWR      30) SIGUSR1
31) SIGUSR2     32) SIGRTMIN  33) SIGRTMIN+1 34) SIGRTMIN+2 35) SIGRTMIN+3
36) SIGRTMIN+4 37) SIGRTMIN+5 38) SIGRTMIN+6 39) SIGRTMIN+7 40) SIGRTMIN+8
41) SIGRTMIN+9 42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMIN+16 49) SIGRTMAX-15 50) SIGRTMAX-14
```

**Command not executed / partially executed :**

3) Command : bg

Purpose : bg command in linux is used to place foreground jobs in background.

Syntax:

bg [job\_spec ...]

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
[1]+  Running                  sleep 500 &

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ sleep 100

[2]+  Stopped                  sleep 100

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
[1]-  Running                  sleep 500 &
[2]+  Stopped                  sleep 100

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ bg %2
[2]+  sleep 100 &

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
[1]-  Running                  sleep 500 &
[2]+  Running                  sleep 100 &
```

4) Command : fg

Purpose : fg command in linux used to put a background job in foreground.

Syntax:

fg [job\_spec]

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ sleep 30

[1]+  Stopped                  sleep 30

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
[1]+  Stopped                  sleep 30

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ bg %1
[1]+  sleep 30 &

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ jobs
[1]+  Running                  sleep 30 &

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ fg %1
sleep 30
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

5) Command : nice

Purpose : nice command in Linux helps in execution of a program/process with modified scheduling priority. It launches a process with a user-defined scheduling priority. In this, if we give a process a higher priority, then Kernel will allocate more CPU time to that process. Whereas the renice command allows you to change and modify the scheduling priority of an already running process. Linux Kernel schedules the process and allocates CPU time accordingly for each of them.

Syntax:

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

Options:

Some options:

1. -n : Add integer N to the niceness (default is 10).
2. -help : Display a help message and exit.
3. -version : Output version information and exit.

Implementation :

```
MINGW64:/c/Users/Nayan Mandliya/Documents/OSSS LAB UNIX COMMANDS
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ nice
0

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.
With no COMMAND, print the current niceness. Niceness values range from
-20 (most favorable to the process) to 19 (least favorable to the process).

Mandatory arguments to long options are mandatory for short options too.
  -n, --adjustment=N  add integer N to the niceness (default 10)
  --help              display this help and exit
  --version           output version information and exit

NOTE: your shell may have its own version of nice, which usually supersedes
the version described here. Please refer to your shell's documentation
for details about the options it supports.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/nice>
or available locally via: info '(coreutils) nice invocation'

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ nice --version
nice (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : User environment

### 1) Command : env

Purpose : env is used to either print environment variables. It is also used to run a utility or command in a custom environment. In practice, env has another common use. It is often used by shell scripts to launch the correct interpreter. In this usage, the environment is typically not changed.

Syntax:

```
env [OPTION]... [-][NAME=VALUE]... [COMMAND [ARG]...]
```

Options:

Some options:

1. -i or --ignore-environment or only - : runs a command with an empty environment.
2. -u or --unset : remove variable from the environment.
3. -0 or --null: End each output line with NULL, not newline.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ env
USERDOMAIN=LAPTOP-BLBOBL3S
OS=Windows_NT
COMMONPROGRAMFILES=C:\Program Files\Common Files
PROCESSOR_LEVEL=6
PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\windows\system32\wind
owsPowerShell\v1.0\Modules
CommonProgramW6432=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
platformcode=KV
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ env -i

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ env -u TO_REMOVE
USERDOMAIN=LAPTOP-BLBOBL3S
OS=Windows_NT
COMMONPROGRAMFILES=C:\Program Files\Common Files
PROCESSOR_LEVEL=6
PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\windows\system32\wind
owsPowerShell\v1.0\Modules
CommonProgramW6432=C:\Program Files\Common Files
```

```
$ env -0
USERDOMAIN=LAPTOP-BLBOBL3SOS=windows_NTCOMMONPROGRAMFILES=C:\Program Files\Commo
n FilesPROCESSOR_LEVEL=6PSModulePath=C:\Program Files\WindowsPowerShell\Modules;
C:\windows\system32\WindowsPowerShell\v1.0\ModulesCommonProgramW6432=C:\Program
Files\Common FilesCommonProgramFiles(x86)=C:\Program Files (x86)\Common Filespla
tformcode=KVMSYSTEM_CARCH=x86_64DISPLAY=needs-to-be-definedHOSTNAME=LAPTOP-BLBOB
```

## 2) Command : uname

Purpose : The command 'uname' displays the information about the system.

Syntax:

uname [OPTION]

Options:

Some options:

1. -a : It prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.
2. -n : It prints the hostname of the network node(current computer).
3. -r option: It prints the kernel release date.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ uname
MINGW64_NT-10.0-19043

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ uname -a
MINGW64_NT-10.0-19043 LAPTOP-BLBOBL3S 3.1.7-340.x86_64 2020-10-23 13:08 UTC x86_64 Msys

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ uname -n
LAPTOP-BLBOBL3S

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ uname -r
3.1.7-340.x86_64

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

3) Command : who

Purpose : who command is used to find out the following information : Time of last system boot, Current run level of the system and List of logged in users and more. The who command is used to get information about currently logged in user on to system.

Syntax:

who [options] [filename]

Options:

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Some options:

1. -q : To count number of users logged on to system.
2. -r : To display current run level of the system.
3. -a : To display all details of current logged in user.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ who

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ who -q

# users=0

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ who -r

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ who -a

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

**Command not executed / partially executed :**

4) Command : logname

Purpose : You can display or print the name of the current user (also know as calling user) using logname command.

Syntax:

logname

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ logname
Nayan Mandliya

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

5) Command : mesg

Purpose : On Unix-like operating systems, the mesg command allows you control write access to your terminal by other users. The write command allows other users to send a message to your terminal session; the mesg command is used to toggle these messages on or off.

Syntax:

mesg [n|y]

Implementation :

```
> mesg
is n
> mesg y
> mesg
is y
> |
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Text Processing

1) Command : basename

Purpose : basename strips directory information and suffixes from file names i.e. it prints the file name NAME with any leading directory components removed. The basename command can be easily used by simply writing basename followed by the file name or the full pathname.

Syntax:

basename OPTION NAME

Options:

Some options:

1. -a(multiple option) : This option lets you support multiple arguments and treat each as a NAME i.e you can give multiple file names or full path names with the use of -a option.
2. -s(suffix) : This option removes a trailing suffix, such as a file extension.
3. -z : This separates the output with NULL rather than a newline

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ basename temp/temp2/target
target

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ basename -a temp/temp2/target temp/temp2/first.txt
target
first.txt

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ basename -s temp/temp2/target temp/temp2/first.txt
first.txt

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ basename -az temp/temp2/target temp/temp2/first.txt
targetfirst.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

2) Command : comm

Purpose : comm compare two sorted files line by line and write to standard output; the lines that are common and the lines that are unique. Suppose you have two lists of people and you are asked to find out the names available in one and not in the other, or even those common to both. comm is the command that will help you to achieve this. It requires two sorted files which it compares line by line.

Syntax:

comm [OPTION]... FILE1 FILE2

Options:

Some options:

1. -1 : suppress first column(lines unique to first file).
2. --nocheck-order :do not check that the input is correctly sorted.
3. --output-delimiter=STR :separate columns with string STR.

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ comm file1.txt file2.txt
Anak
Banak      Ashish
           Bhairav
Danak      Canak
           Dhairya
Enak      Emey

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ comm -1 file1.txt file2.txt
Ashish
Bhairav
           Canak
Dhairya
Emey

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ comm --nocheck-order file1.txt file3.txt
Anak
Banak
Canak
           Danak
           Enak
Banak
Canak

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ comm --output-delimiter=@ file1.txt file2.txt
Anak
@Ashish
Banak
@Bhairav
@@Canak
Danak
@Dhairya
@Emey
Enak
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

3) Command : csplit

Purpose : The csplit command is used to split any file into many parts as required by the user. The parts are determined by context lines. Output pieces of FILE separated by PATTERN(s) to files 'xx00', 'xx01', ..., and output byte counts of each piece to standard output.

Syntax:

csplit [OPTION]... FILE PATTERN...

Options:

Some options:

1. -f(prefix) : It use PREFIX in place of 'xx'.
2. -n(digits) : Use given number of digits instead of 2.
3. -z(elide empty files) : Remove empty output files.

Implementation :

```
$ cat file1.txt
1. Nayan
2. OSSS
3. LAB
4. LABWORK
5. Assignment
6. Coding
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ csplit file1.txt 4
27
36

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
file1.txt  xx00  xx01

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat xx00
1. Nayan
2. OSSS
3. LAB

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat xx01
4. LABWORK
5. Assignment
6. Coding
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ csplit -f nayan file1.txt 4
27
36

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
file1.txt  nayan00  nayan01  xx00  xx01
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

```
Nayan_Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ csplit -n 3 file1.txt 2
10
53

Nayan_Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
file1.txt nayan00 nayan01 xx00 xx000 xx001 xx01

Nayan_Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ csplit -z file1.txt 6
54
9

Nayan_Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
file1.txt nayan00 nayan01 xx00 xx000 xx001 xx01
```

#### 4) Command : cut

**Purpose :** The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field. Basically the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided then data from each file is not precedes by its file name.

**Syntax:**

cut OPTION... [FILE]...

**Options:**

**Some options:**

1. **-b(byte):** To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma. Range of bytes can also be specified using the hyphen(-). It is necessary to specify list of byte numbers otherwise it gives error. Tabs and backspaces are treated like as a character of 1 byte.
2. **-c (column):** To cut by character use the -c option. This selects the characters given to the -c option. This can be a list of numbers separated comma or a range of numbers separated by hyphen(-). Tabs and backspaces are treated as a character.
3. **-f (field):** -f option is useful for fixed-length lines. Most unix files doesn't have fixed-length lines. To extract the useful information you need to cut by fields rather than columns. List of the fields number specified must be separated by

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

comma. If -d option is used then it considered space as a field separator or delimiter:

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat file1.txt
1. Nayan
2. OSSS
3. LAB
4. LABWORK
5. Assignment
6. Coding
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cut -b 1,2,3,4,5 file1.txt
1. Na
2. OS
3. LA
4. LA
5. As
6. Co
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cut -c 1,3,4,5,6,7 file1.txt
1 Naya
2 OSSS
3 LAB
4 LABW
5 Assi
6 Codi
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat file2.txt
1. Nayan Mandliya
2. OSSS Lab
3. LAB Work
4. LABWORK Assigned
5. Assignment Done
6. Coding Done
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cut -d " " -f 3 file2.txt
Mandliya
Lab
Work
Assigned
Done
Done
```

5) Command : diff

Purpose : diff stands for difference. This command is used to display the differences in the files by comparing the files line by line. Unlike its fellow members, cmp and comm, it tells us which lines in one file have is to be changed to make the two files identical. The important thing to remember is that diff uses certain special symbols and instructions that are required to make two files identical. It tells you the instructions on how to change the first file to make it match the second file. Special symbols are:

a : add , c : change , d : delete

Syntax:

diff [options] File1 File2

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Options:

Some options:

1. -c (context) : To view differences in context mode, use the -c option.
2. -u (unified) : To view differences in unified mode, use the -u option. It is similar to context mode but it doesn't display any redundant information or it shows the information in concise form.
3. -i : By default this command is case sensitive. To make this command case insensitive use -i option with diff.

Implementation :

```
$ cat file1.txt
1. Nashik
2. Vadodra
3. Pune
4. Kurla

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat file2.txt
1. Nashik
2. Gujarat
3. Gandhinagar
4. Kurla

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ diff file1.txt file2.txt
2,3c2,3
< 2. Vadodra
< 3. Pune
---
> 2. Gujarat
> 3. Gandhinagar

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ diff -c file1.txt file2.txt
*** file1.txt      Sun Aug 29 18:14:00 2021
--- file2.txt      Sun Aug 29 18:14:26 2021
*****
*** 1,4 ****
1. Nashik
! 2. Vadodra
! 3. Pune
4. Kurla
--- 1,4 ----
1. Nashik
! 2. Gujarat
! 3. Gandhinagar
4. Kurla

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ diff -u file1.txt file2.txt
--- file1.txt      2021-08-29 18:14:00.582434900 +0530
+++ file2.txt      2021-08-29 18:14:26.531614400 +0530
@@ -1,4 +1,4 @@
1. Nashik
-2. Vadodra
-3. Pune
+2. Gujarat
+3. Gandhinagar
4. Kurla

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ diff -i file1.txt file2.txt
2,3c2,3
< 2. Vadodra
< 3. Pune
---
> 2. Gujarat
> 3. Gandhinagar
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Shell Built-in

1) Command : echo

Purpose : echo command in unix is used to display line of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

Syntax:

echo [option] [string]

Options:

Some options:

1. -e : It enables the interpretation of some special characters which will format differently while printing.
2. -n : this option is used to omit echoing trailing newline .
3. \* : this command will print all files/folders, similar to ls command .

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ echo "Hello World"
Hello World
```

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ echo -e "Hello \nNayan \nMandliya"
Hello
Nayan
Mandliya
```

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ echo -n "Hello Nayan Mandliya"
Hello Nayan Mandliya
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ echo *
File.txt Temp1 test.sh timepasss.sh
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Command not executed / partially executed :

### 2) Command : alias

Purpose : alias command instructs the shell to replace one string with another string while executing the commands. When we often have to use a single big command multiple times, in those cases, we create something called as alias for that command. Alias is like a shortcut command which will have same functionality as if we are writing the whole command.

Syntax:

alias [-p] [name[=value] ... ]

Options:

Some options:

1. -p : This option prints all the defined aliases in reusable format.
2. -help option : It displays help information.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ alias -p
alias ipython='winpty ipython.exe'
alias ll='ls -l'
alias ls='ls -F --color=auto --show-control-chars'
alias node='winpty node.exe'
alias psql='winpty psql.exe'

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ alias --help
alias: alias [-p] [name[=value] ... ]
    Define or display aliases.

Without arguments, 'alias' prints the list of aliases in the reusable
form 'alias NAME=VALUE' on standard output.

Otherwise, an alias is defined for each NAME whose VALUE is given.
A trailing space in VALUE causes the next word to be checked for
alias substitution when the alias is expanded.

Options:
  -p          print all defined aliases in a reusable format

Exit Status:
alias returns true unless a NAME is supplied for which no alias has been
defined.
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

3) Command : cd

Purpose : cd command in linux known as change directory command. It is used to change current working directory.

Syntax:

cd [directory]

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
File.txt  Temp1/

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cd Temp1

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS/Temp1
$ ls
Temp2/

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS/Temp1
$ cd Temp2

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS/Temp1/Temp2
$ |
```

4) Command : test

Purpose : Test is used in conditional execution. It is used for: File attributes comparisons, Perform string comparisons, Basic arithmetic comparisons. test exits with the status determined by EXPRESSION. Placing the EXPRESSION between square brackets ([ and ]) is the same as testing the EXPRESSION with test. To see the exit status at the command prompt, echo the value "\$?" A value of 0 means the expression evaluated as true, and a value of 1 means the expression evaluated as false.

Syntax:

test EXPRESSION

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ test 100 -gt 99 echo "Yes, that's true." || echo "No, that's false."
bash: test: too many arguments
No, that's false.

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

5) Command : unset

Purpose : unset allows you to remove variables or functions.

Syntax:

Unset variable-name

Options:

Some options:

1. -v: Unset a variable
2. -f: Unset a function

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ unset DEBUG_VALUE
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Searching

### 1) Command : find

**Purpose :** The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions. By using the '-exec' other UNIX commands can be executed on files or folders found.

**Syntax:** find [where to start searching from] [expression determines what to find] [-options] [what to find]

**Options:**

Some options:

1. -name : Search for files that are specified by the text after name.
2. -empty : Search for empty files and directories.
3. -perm text : Search for the file if permission is 'text'.

**Implementation :**

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ls
'Folder 1'/ 'Folder 2'/

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ find -name file2.txt
./Folder 2/Folder 3/file2.txt

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ find -empty

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ find -perm 664

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

2) Command : grep

Purpose : The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

Syntax:

grep [options] pattern [files]

Options:

Some options:

1. -i(Case insensitive search) : The -i option enables to search for a string case insensitively in the give file.
2. -c(Displaying the count of number of matches) : We can find the number of lines that matches the given string/pattern
3. -l(Display the file names that matches the pattern) : We can just display the files that contains the given string/pattern.

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat File.txt
This is a file for unix search functions. In this many search functions are available.
Search is a very useful feature.
Search includes 2 operations grep and find.
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ grep -i "is" File.txt
This is a file for unix search functions. In this many search functions are available.
Search is a very useful feature.

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ grep -c "is" File.txt
2

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ grep -l "is" *
File.txt

Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ |
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Documentation

1) Command : man

Purpose : man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

Syntax:

man [OPTION]... [COMMAND NAME]...

Options:

Some options:

1. -f : One may not be able to remember the sections in which a command is present. So this option gives the section in which the given command is present.
2. -a : This option helps us to display all the available intro manual pages in succession.
3. -k option: This option searches the given command as a regular expression in all the manuals and it returns the manual pages with the section number in which it is found.

Implementation : Command not working.

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ man
bash: man: command not found

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ man printf
bash: man: command not found
```

```
> man printf
No manual entry for printf
See 'man 7 undocumented' for help when manual pages are not available.
> 
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Software Development

1) Command : ar

Purpose : ar command is used to create, modify and extract the files from the archives. An archive is a collection of other files having a particular structure from which the individual files can be extracted. Individual files are termed as the members of the archive.

Syntax:

ar [OPTIONS] archive\_name member\_files

Options:

Some options:

1. r : This is used to create archive, insert files in archive. This is different from q as it deletes any previously existing members. If any member filename does not exist it throws an error. By default, it adds a new member at end of the file.
2. p : This option is used to print the specified members of a archive in a standard output file if you do not use modifier it will print member as it is an output file whereas if you use modifier v then it will show member name before it is copied to output file.
3. d : Deletes modules from archive. Specify names of the modules as member...; When you add modifier v, ar lists each module as it is deleted.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat File.txt
Hello
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat FileNew.txt
Hello world
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ar r super.nayan *File.txt
D:\C++\bin\ar.exe: creating super.nayan
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ar r super.nayan *FileNew.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ar p super.nayan
HelloHello world
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ar d super.nayan FileNew.txt
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ar p super.nayan
Hello
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## **Command not executed / partially executed :**

### 2) Command : ctags

**Purpose :** ctags command in Linux system is used for the with the classic editors. It allows quick access across the files (For example quickly seeing definition of a function). A user can run tags or ctags inside a directory to create a simple index of the source files while working on. Tags-capable editors like vi/vim can then refer to these tags index file to allow you to follow references. Basically, this command generates the tag files for source code. It is also used to create a cross reference file which lists and contain the information about the various source objects found in a set of human-readable language files.

**Syntax:**

ctags [options] [file(s)]

**Options:**

**Some options:**

1. -a : This option used to append the tags to an existing tag file.
2. -B : This option used for backward searching patterns.
3. -e : This option used for output a tag file for use with Emacs. If this program is being executed by the name etags, this option is already enabled by default.

**Implementation :**

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ ctags
bash: ctags: command not found
```

### 3) Command : lex

**Purpose :** On Unix-like operating systems, the lex command generates programs to perform lexical tasks. The lex utility generates C programs to be used in lexical processing of character input, and that can be used as an interface to yacc.

**Syntax:**

lex [-cntv] [-e | -w] [-V -Q [y | n]] [file]...

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Options:

Some options:

1. -c : Indicate C-language action.
2. -n : Suppresses the summary of statistics usually written with the -v option. If no table sizes are specified in the lex source code and the -v option is not specified, then -n is implied.
3. -t : Write the resulting program to standard output instead of lex.yy.c.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ lex
bash: lex: command not found
```

#### 4) Command : make

**Purpose :** The purpose of the make utility is to determine automatically which pieces of a large program need to be recompiled, and issue the commands to recompile them. you can use make with any programming language whose compiler can be run with a shell command. In fact, make is not limited to programs. You can use it to describe any task where some files must be updated automatically from others whenever the others change.

**Syntax:**

`make [ -f makefile ] [ options ] ... [ targets ] ...`

**Options:**

**Some options:**

1. -b : prints online help and exit These options are ignored for compatibility with other versions of make..
2. -d : Print debugging information in addition to normal processing.
3. -e : Give variables taken from the environment precedence over variables from makefiles.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ make
bash: make: command not found
```

5) Command : strip

Purpose : GNU strip discards all symbols from object files objfile. The list of object files may include archives. At least one object file must be given. strip modifies the files named in its argument, rather than writing modified copies under different names.

Syntax:

strip [options] objfile [...]

Options:

Some options:

1. -F bfdname : Treat the original objfile as a file with the object code format bfdname, and rewrite it in the same format.
2. -I bfdname : Treat the original objfile as a file with the object code format bfdname.
3. -O bfdname : Replace objfile with a file in the output format bfdname.

Implementation :

```
$ strip
Usage: D:\C++\bin\strip.exe <option(s)> in-file(s)
Removes symbols and sections from files
The options are:
-I --input-target=<bfdname>      Assume input file is in format <bfdname>
-O --output-target=<bfdname>     Create an output file in format <bfdname>
-F --target=<bfdname>           Set both input and output format to <bfdname>
-p --preserve-dates             Copy modified/access timestamps to the output
-D --enable-deterministic-archives
                                Produce deterministic output when stripping a
rchives
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## Category : Miscellaneous

1) Command : cal

Purpose : If a user wants a quick view of the calendar in the Linux terminal, cal is the command for you. By default, the cal command shows the current month calendar as output. cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.

Syntax:

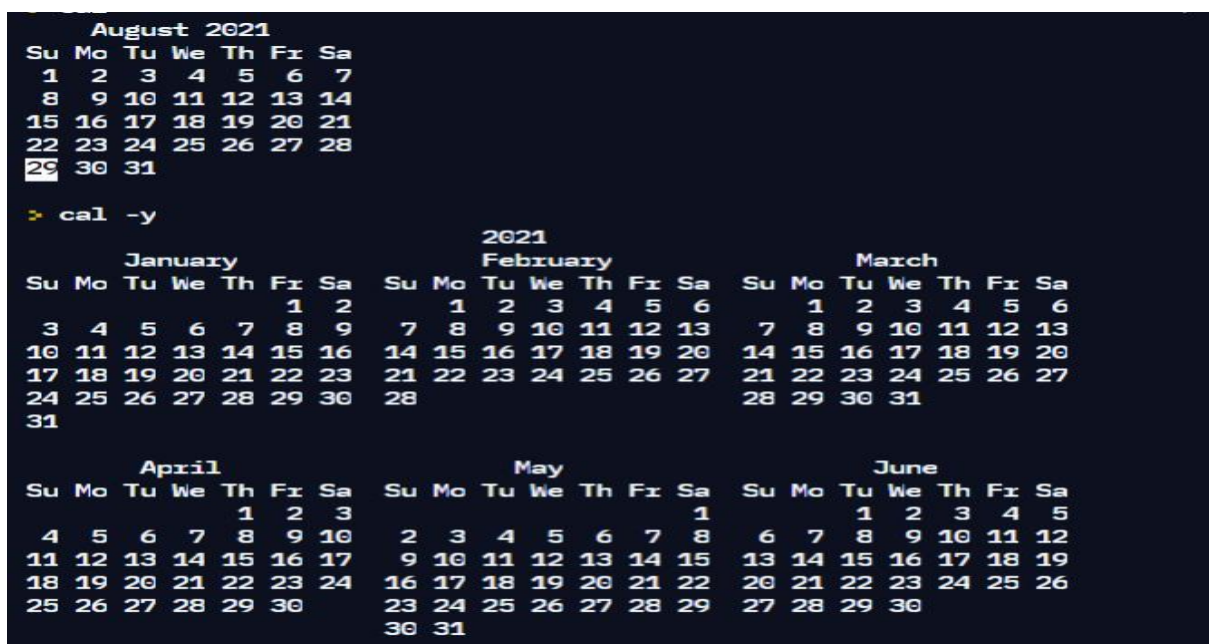
cal [ [ month ] year]

Options:

Some options:

1. -y : Shows the calendar of the complete current year with the current date highlighted.
2. -j : Shows the calendar of the current month in the Julian calendar format not in the default Gregorian calendar format.
3. -3 : Shows calendar of previous, current and next month.

Implementation :



```

August 2021
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

> cal -y

2021
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  7  8  9 10 11 12 13 14 15 16 17 18 19 20
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
24 25 26 27 28 29 30 31

April May June
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  7  8  9 10 11 12 13 14 15 16 17 18 19
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
18 19 20 21 22 23 24 25 26 27 28 29 30 31
25 26 27 28 29 30 31
  
```



**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)

**Department of Computer Engineering**

July							August							September						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2	3	1	2	3	4	5	6	7			1	2	3	4
4	5	6	7	8	9	10	8	9	10	11	12	13	14	5	6	7	8	9	10	11
11	12	13	14	15	16	17	15	16	17	18	19	20	21	12	13	14	15	16	17	18
18	19	20	21	22	23	24	22	23	24	25	26	27	28	19	20	21	22	23	24	25
25	26	27	28	29	30	31	29	30	31					26	27	28	29	30		

October							November							December						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2		1	2	3	4	5	6				1	2	3	4
3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25
24	25	26	27	28	29	30	28	29	30					26	27	28	29	30	31	
31																				

➤ cal -j

August 2021

Su	Mo	Tu	We	Th	Fr	Sa
213	214	215	216	217	218	219
220	221	222	223	224	225	226
227	228	229	230	231	232	233
234	235	236	237	238	239	240
241	242	243				

➤ cal -3

July							August							September						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
					1	2	1	2	3	4	5	6	7				1	2	3	4
4	5	6	7	8	9	10	8	9	10	11	12	13	14	5	6	7	8	9	10	11
11	12	13	14	15	16	17	15	16	17	18	19	20	21	12	13	14	15	16	17	18
18	19	20	21	22	23	24	22	23	24	25	26	27	28	19	20	21	22	23	24	25
25	26	27	28	29	30	31	29	30	31					26	27	28	29	30		

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

2) Command : od

Purpose : od command in Linux is used to convert the content of input in different formats with octal format as the default format. This command is especially useful when debugging Linux scripts for unwanted changes or characters. If more than one file is specified, od command concatenates them in the listed order to form the input. It can display output in a variety of other formats, including hexadecimal, decimal, and ASCII. It is useful for visualizing data that is not in a human-readable format, like the executable code of a program.

Syntax:

od [OPTION]... [FILE]...

Options:

Some options:

1. -b : It displays the contents of input in octal format.
2. -c : It displays the contents of input in character format.
3. -i Option : It displays output as decimal integer.

Implementation :

```
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ cat File.txt
100
200
300
400
500
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ od File.txt
0000000 030061 006460 031012 030060 005015 030063 006460 032012
0000020 030060 005015 030065 000060
0000027
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ od -b File.txt
0000000 061 060 060 015 012 062 060 060 015 012 063 060 060 015 012 064
0000020 060 060 015 012 065 060 060
0000027
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ od -c File.txt
0000000 1 0 0 \r \n 2 0 0 \r \n 3 0 0 \r \n 4
0000020 0 0 \r \n 5 0 0
0000027
Nayan Mandliya@LAPTOP-BLB0BL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ od -i File.txt
0000000 221261873 808464906 808651277 873073968
0000020 168636464 3158069
0000027
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

3) Command : sleep

Purpose : sleep command is used to create a dummy job. A dummy job helps in delaying the execution. It takes time in seconds by default but a small suffix(s, m, h, d) can be added at the end to convert it into any other format. This command pauses the execution for an amount of time which is defined by NUMBER.

Syntax:

sleep NUMBER[SUFFIX]...

Options:

Some options:

1. -help : It displays help information.
2. -version : It displays version information.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ sleep 10

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ sleep --help
Usage: sleep NUMBER[SUFFIX]...
       or: sleep OPTION
Pause for NUMBER seconds.  SUFFIX may be 's' for seconds (the default),
'm' for minutes, 'h' for hours or 'd' for days.  NUMBER need not be an
integer.  Given two or more arguments, pause for the amount of time
specified by the sum of their values.

       --help      display this help and exit
       --version   output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/sleep>
or available locally via: info '(coreutils) sleep invocation'

Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ sleep --version
sleep (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Jim Meyering and Paul Eggert.
```

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

## **Command not executed / partially executed :**

### 4) Command : bc

**Purpose :** bc command is used for command line calculator. It is similar to basic calculator by using which we can do basic mathematical calculations. Arithmetic operations are the most basic in any kind of programming language. Linux or Unix operating system provides the bc command and expr command for doing arithmetic calculations. You can use these commands in bash or shell script also for evaluating arithmetic expressions.

**Syntax:**

bc [ -hlwsqv ] [long-options] [ file ... ]

**Options:**

**Some options:**

1. -l[mathlib] : Define the standard math library.
2. -q : To avoid system generated message on output screen.
3. -h : To display help information.

**Implementation :**

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ bc
bash: bc: command not found
```

### 5) Command : lp

**Purpose :** The lp command in Linux stands for 'Line printer' which lets you print the files through the terminal. There is no need to change or manage the settings through the GUI. You can simply manage the printers using lp command. This command is also known as the printer management command Linux. The lp command in Linux is one of the CUPS i.e. Common Unix Printing System.

**Syntax:**

lp [filename]

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Options:

Some options:

1. -d : To print the file using a particular printer.
2. -n : Print multiple copies of the document
3. -q : Set the priority of the printer in the printer queue.

Implementation :

```
Nayan Mandliya@LAPTOP-BLBOBL3S MINGW64 ~/Documents/OSSS LAB UNIX COMMANDS
$ lp
bash: lp: command not found
```

## Category : Unix system status command

1) Command : hostname

Purpose : hostname command in Linux is used to obtain the DNS(Domain Name System) name and set the system's hostname or NIS(Network Information System) domain name. A hostname is a name which is given to a computer and it attached to the network. Its main purpose is to uniquely identify over a network.

Syntax:

hostname -[option] [file]

Options:

Some options:

1. -i : This option is used to get the IP(network) addresses. This option works only if the hostname is resolvable.
2. -f : This option is used to get the Fully Qualified Domain Name(FQDN). It contains short hostname and DNS domain name.
3. -s : This option is used to get the hostname in short. The short hostname is the section of hostname before the first period/dot(.). If the hostname has no period, the full hostname is displayed.



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

Implementation :

```
1 hostname
2 hostname -i
3 hostname -f
4 hostname -s
5
```

```
$bash -f main.sh
1f9df56d7c9c
::1 127.0.0.2
1f9df56d7c9c
1f9df56d7c9c
```

**Conclusion:** By performing this experiment understood different commands associated with unix and also seen categories in which different commands fall. Understood how different tasks are performed by operating system. Also understood what shell is and how to execute different commands in shell.

**Post Lab Descriptive Questions**

1. Explain different functions of operating system.

ANS) 1) **Security** – The operating system uses password protection to protect user data and similar other techniques. it also prevents unauthorized access to programs and user data.

2) **Control over system performance** – Monitors overall system health to help improve performance. records the response time between service requests and system response to have a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems.

3) **Job accounting** – Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of user.

4) **Error detecting aids** – Operating system constantly monitors the system to detect errors and avoid the malfunctioning of computer system.

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

**5) Coordination between other software and users** – Operating systems also coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.

**6) Memory Management** – The operating system manages the Primary Memory or Main Memory. Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. Main memory is a fast storage and it can be accessed directly by the CPU. For a program to be executed, it should be first loaded in the main memory. An Operating System performs the following activities for memory management: It keeps tracks of primary memory, i.e., which bytes of memory are used by which user program. The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used. In multi programming, the OS decides the order in which process are granted access to memory, and for how long. It Allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

**7) Processor Management** – In a multi programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has. This function of OS is called process scheduling. An Operating System performs the following activities for processor management. Keeps tracks of the status of processes. The program which perform this task is known as traffic controller. Allocates the CPU that is processor to a process. De-allocates processor when a process is no more required.

**8) Device Management** – An OS manages device communication via their respective drivers. It performs the following activities for device management. Keeps tracks of all devices connected to system. designates a program responsible for every device known as the Input/Output controller. Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.

**9) File Management** – A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file and more... These facilities are collectively known as the file system.

2. What are the default permission assigned by Unix for Directory.



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

ANS) 1) The system default permission values are 777 ( rwxrwxrwx ) for folders and 666 ( rw-rw-rw- ) for files. The default mask for a non-root user is 002, changing the folder permissions to 775 ( rwxrwxr-x ), and file permissions to 664 ( rw-rw-r-- ).

2) The default umask 002 used for normal user. With this mask default directory permissions are 775 and default file permissions are 664.

3) The default umask for the root user is 022 result into default directory permissions are 755 and default file permissions are 644.

4) For directories, the base permissions are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).

3. Give difference between DOS and WINDOWS.

ANS)

Key	DOS	Windows
Definition	DOS (Disk Operating System) are simple text command operating systems that were popular from 1981 to 1995.	Windows is a range of graphical interface operating systems that are developed and sold by Microsoft.
Tasking Nature	DOS is single tasking OS.	Windows is multi-tasking OS.
Power Consumption	DOS consumes quite low power.	Windows consumes high power.
Memory consumption	DOS memory requirements are quite low.	Windows memory requirements are quite high as compared to DOS.
Usage	DOS is complex in usage. You need to remember commands to use DOS properly.	Windows usages is user-friendly and is quite simple to use.
Speed	DOS command execution is faster than Windows.	Windows operations are slower as compared to DOS.
GUI	DOS used a text based interface that required text and codes to operate.	Windows uses graphics, images and text.

**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)

**Department of Computer Engineering**

Registry and Swap Files	DOS uses a directory system, where all the files are contained within a particular directory or a subdirectory.	Windows uses a different registry compared to DOS, making it difficult to manually delete programs. An excessive number of temporary files and file fragments can cause the system to slow down or crash.
Current Uses	More ideally used for prototyping, testing, and making automated systems.	Used worldwide as the most popular operating system.

**4. Explain Booting Process.**

ANS) Booting is basically the process of starting the computer. When the CPU is first switched on it has nothing inside the Memory. In order to start the Computer, load the Operating System into the Main Memory and then Computer is ready to take commands from the User. Booting happens when you start the computer. This happens when we turned ON the power or the computer restarts. The system BIOS (Basic Input/Output System) makes the peripheral devices active. Further, it requires that the boot device loads the operating system into the main memory.

**Types of Booting**

There are two types of booting:

**Cold Booting:** A cold boot is also called a hard boot. It is the process when we first start the computer. In other words, when the computer is started from its initial state by pressing the power button it is called cold boot. The instructions are read from the ROM and the operating system is loaded in the main memory.

**Warm Booting:** Warm Boot is also called soft boot. It refers to when we restart the computer. Here, the computer does not start from the initial state. When the system gets stuck sometimes it is required to restart it while it is ON. Therefore, in this condition the warm boot takes place. Restart button or CTRL+ALT+DELETE keys are used for warm boot.

**Steps of Booting**

**We can describe the boot process in six steps:**

**K. J. Somaiya College of Engineering, Mumbai-77**  
(A Constituent College of Somaiya Vidyavihar University)  
**Department of Computer Engineering**

- 1. The Startup :** It is the first step that involves switching the power ON. It supplies electricity to the main components like BIOS and processor.
- 2. BIOS:** Power On Self Test. It is an initial test performed by the BIOS. Further, this test performs an initial check on the input/output devices, computer's main memory, disk drives, etc. Moreover, if any error occurs, the system produces a beep sound.
- 3. Loading of OS:** In this step, the operating system is loaded into the main memory. The operating system starts working and executes all the initial files and instructions.
- 4. System Configuration:** In this step, the drivers are loaded into the main memory. Drivers are programs that help in the functioning of the peripheral devices.
- 5. Loading System Utilities:** System utilities are basic functioning programs, for example, volume control, antivirus, etc. In this step, system utilities are loaded into the memory.
- 6. User Authentication:** If any password has been set up in the computer system, the system checks for user authentication. Once the user enters the login Id and password correctly the system finally starts.

**Date: 12 / 9 / 2021**

**Signature of faculty in-charge**