

1. Display the date using 'date' command.

```
hrisha_regex@Hrisha:~$ date
Tue Jul 23 17:30:10 IST 2024
hrisha_regex@Hrisha:~$
```

2. Check who are the users logged in using the 'who' command.

```
hrisha_regex@Hrisha:~$ who
hrisha_regex pts/1      2024-07-23 17:30
hrisha_regex@Hrisha:~$
```

3. Check the running processes using the 'ps' command.

```
hrisha_regex@Hrisha:~$ ps
  PID TTY          TIME CMD
  558 pts/0        00:00:00 bash
  791 pts/0        00:00:00 ps
hrisha_regex@Hrisha:~$
```

4. List the files with 'ls' command with and without -l option.

```
hrisha_regex@Hrisha:~$ ls
q
hrisha_regex@Hrisha:~$ ls -l
total 12
-rw-r--r-- 1 hrisha_regex hrisha_regex 9785 Jul 22 11:42 q
hrisha_regex@Hrisha:~$
```

5. Check the manual of 'ls' command.

```
ls(1)                                User Commands                                ls(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTIONS]... [FILE]...

DESCRIPTION
  List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuwxX nor
  --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
      with -l, print the Author of each file
  -b, --escape
      print C-style escapes for non-graphic characters
  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., "--block-size=M"; see SIZE format below
  -B, --ignore-backups
      do not list implied entries ending with ~

(Press h for help or q to quit)
```

6. Show the command used to display (i) filenames (ii) processes (iii) users.

```
hrisha_regex@Hrisha:~$ ls
q
hrisha_regex@Hrisha:~$ ls -l
total 12
-rw-r--r-- 1 hrisha_regex hrisha_regex 9785 Jul 22 11:42 q
hrisha_regex@Hrisha:~$ ps
  PID TTY          TIME CMD
   558 pts/0    00:00:00 bash
   818 pts/0    00:00:00 ps
hrisha_regex@Hrisha:~$ who
hrisha_regex pts/1          2024-07-23 17:30
hrisha_regex@Hrisha:~$
```

7. Check and state the difference between man and whatis command by checking man cp & whatis cp

```
OP(1)                                     User Commands                                     OP(1)
NAME
  cp - copy files and directories

SYNOPSIS
  cp [OPTION]... [-t] SOURCE DEST
  cp [OPTION]... SOURCE... DIRECTORY
  cp [OPTION]... -A DIRECTORY SOURCE...

DESCRIPTION
  Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --archive
      same as --dt --preserve=all
  --attributes-only
      don't copy the file data, just the attributes
  --backup[=CONTROL]
      make a backup of each existing destination file
  -b      like --backup but does not accept an argument
  --copy-contents
      copy contents of special files when recursive
  -d      same as --no-dereference --preserve=links
  -f, --force
      log file:

hrisha_regex@Hrisha:~$ man cp
hrisha_regex@Hrisha:~$
hrisha_regex@Hrisha:~$ whatis cp
cp (1)          - copy files and directories
```

8. What is primary difference between printf and echo command. Check and print

```
hrisha_regex@Hrisha:~$ printf hello
hellohrisha_regex@Hrisha:~$ echo hello
hello
hrisha_regex@Hrisha:~$
```

9. In the home directory, create a directory MCA2022 inside the MCA2022, create another directory and get into the directory [-/MCA2022/Ankur\_A\_00\$].

```
hrisha_regex@Hrisha:~$ mkdir MCA2022
hrisha_regex@Hrisha:~$ cd MCA2022
hrisha_regex@Hrisha:~/MCA2022$ mkdir Hrisha_B_55
hrisha_regex@Hrisha:~/MCA2022$ cd Hrisha_B_55
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55$ pwd
/home/hrisha_regex/MCA2022/Hrisha_B_55
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55$
```

10. Go to the subdirectory and create another subdirectory 'Unix\_File\_System' within it.

```
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55$ mkdir Unix_File_System
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55$ cd Unix_File_System
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ pwd
/home/hrisha_regex/MCA2022/Hrisha_B_55/Unix_File_System
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$
```

11. Create the subdirectories TestA,TestB,TestC and corresponding sub-directories TestA-1,TestA-2, TestB-1,TestB-2, TestC-1,TestB-2-i in a single command.

```
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ mkdir
r -p TestA/TestA-1 TestA/TestA-2 TestB/TestB-1 TestB/TestB-2 Tes
tB/TestB-3 TestC/TestC-1 TestB-2-i
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ tree

Command 'tree' not found, but can be installed with:

sudo snap install tree # version 1.8.0+pkg-3fd6, or
sudo apt install tree # version 1.8.0-1

See 'snap info tree' for additional versions.

hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ find
.
./TestB-2-i
./TestC
./TestC/TestC-1
./TestB
./TestB/TestB-2
./TestB/TestB-1
./TestB/TestB-3
./TestA
./TestA/TestA-1
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ |
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$
```

12. Show the absolute path of TestB-2-i.

```
hrisha_regex@Hrisha:~/MCA2022/Hrisha_B_55/Unix_File_System$ realpath TestB/TestB-2-i/home/Hrisha/MCA2022/Hrisha_B_55/Unix_File_S
ysten/-TestB/Test-2-i
```