Assignment 1

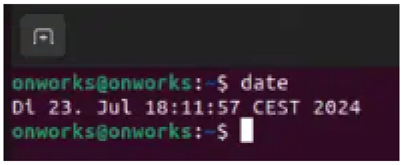
OPERATING SYSTEM

TOPIC: BASIC UNIX COMMANDS & FILE SYSTEM COMMANDS

NAME: Devesh Tulshyan
MCA 3A
Roll Number: 22

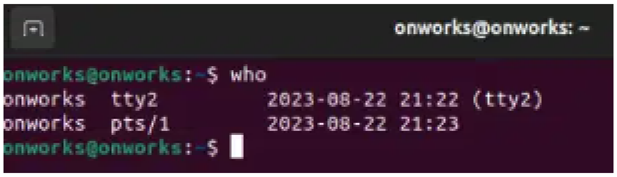
---

1. Display the date using the “**date**” command.



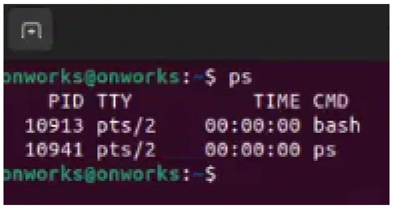
```
onworks@onworks:~$ date
Di 23. Jul 18:11:57 CEST 2024
onworks@onworks:~$ 
```

2. Check who are the users logged in using the “**who**” command.



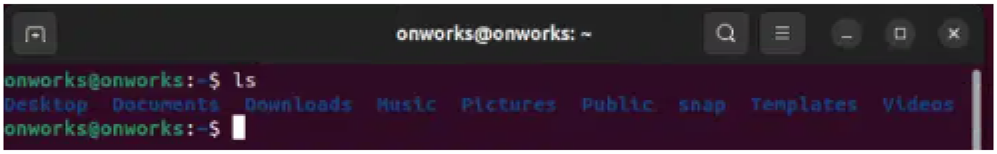
```
onworks@onworks:~$ who
onworks  tty2         2023-08-22 21:22 (tty2)
onworks  pts/1        2023-08-22 21:23
onworks@onworks:~$ 
```

3. Check the running processes using the “**ps**” command.

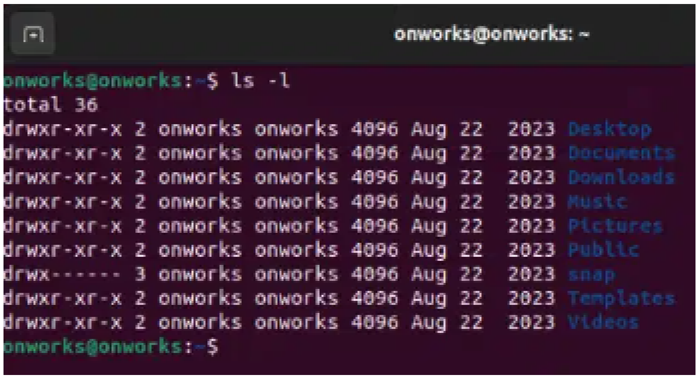


```
onworks@onworks:~$ ps
    PID TTY          TIME CMD
  10913 pts/2    00:00:00 bash
  10941 pts/2    00:00:00 ps
onworks@onworks:~$
```

4. List the files with “**ls**” command with and without **-l** option.

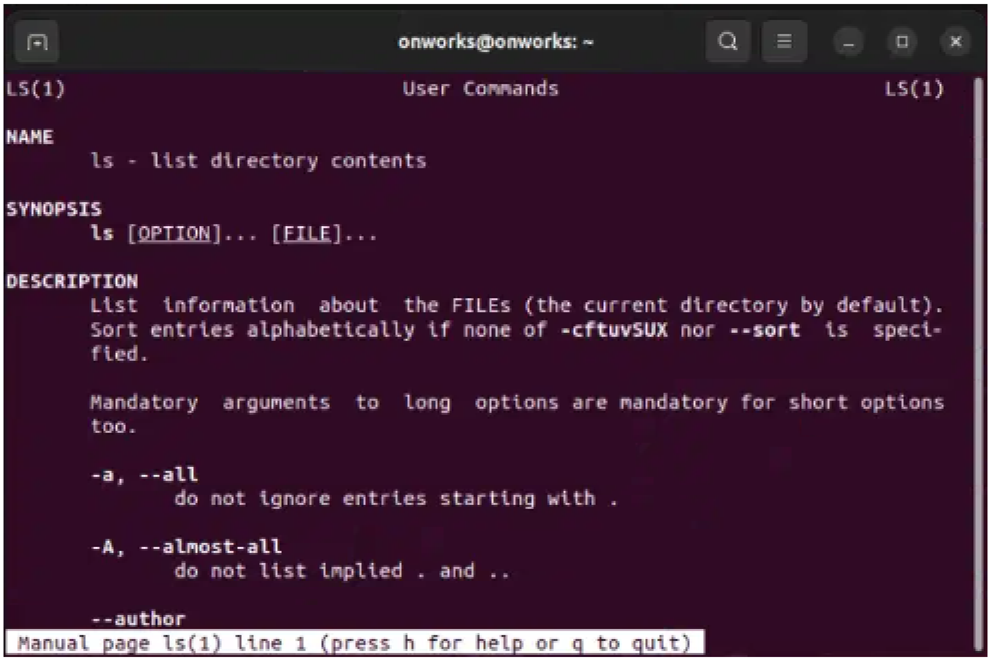


```
onworks@onworks:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
onworks@onworks:~$ 
```



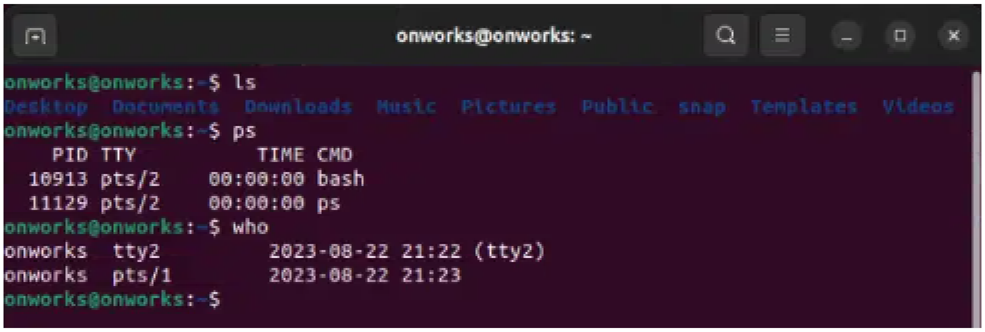
```
onworks@onworks:~$ ls -l
total 36
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Desktop
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Documents
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Downloads
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Music
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Pictures
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Public
drwx------ 3 onworks onworks 4096 Aug 22  2023 snap
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Templates
drwxr-xr-x 2 onworks onworks 4096 Aug 22  2023 Videos
onworks@onworks:~$
```

5. Check the *manual of* **ls** command.



```
LS(1)                            User Commands                           LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

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

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

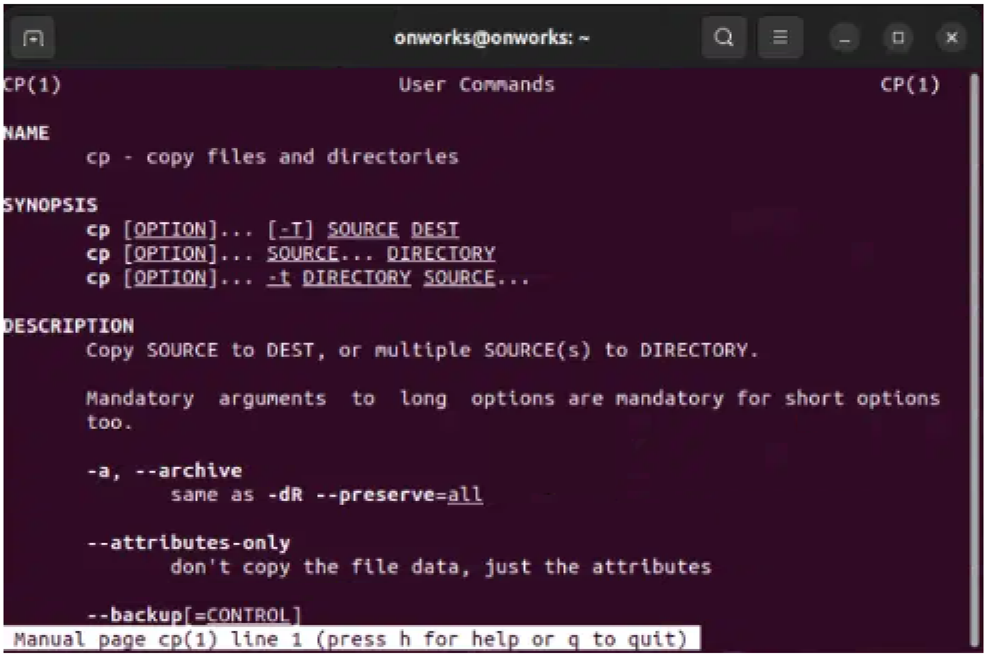
6. Show the commands used to display (i) filenames (ii) processes (iii) users.



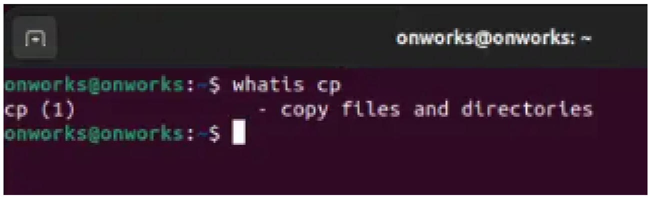
```
onworks@onworks:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
onworks@onworks:~$ ps
    PID TTY          TIME CMD
  10913 pts/2    00:00:00 bash
  11129 pts/2    00:00:00 ps
onworks@onworks:~$ who
onworks  tty2         2023-08-22 21:22 (tty2)
onworks  pts/1        2023-08-22 21:23
onworks@onworks:~$
```

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

   The man command prints the manual for any command specified to it as arguments, the manual contains each and every possible way we can add flags to the command and its syntax so that we can get custom outputs for our needs. Whereas, the whatis command gives the general use description of the command specified, just for the user to understand what the command is about.



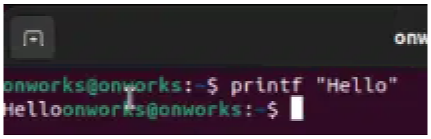
```
CP(1)                         User Commands                        CP(1)

NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t 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 -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes

       --backup[=CONTROL]
 Manual page cp(1) line 1 (press h for help or q to quit)
```



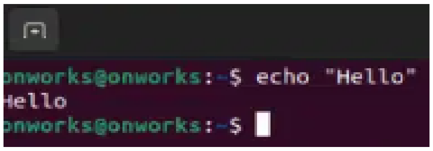
```
onworks@onworks:~$ whatis cp
cp (1)               - copy files and directories
onworks@onworks:~$
```

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

The difference between printf and echo command is that the echo command appends a newline character to anything we try to output in the console but the printf command doesn't. So, after the echo statement, we get to a new line but in case of printf command, we stay on the same line after the output.

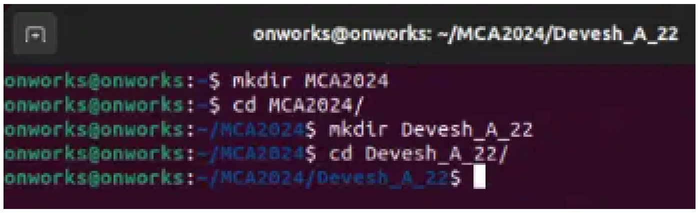


```
nworks@onworks:~$ printf "Hello"
Helloonworks@onworks:~$ 
```



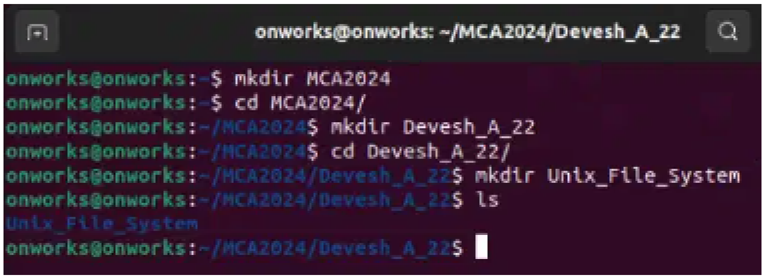
```
nworks@onworks:~$ echo "Hello"
Hello
nworks@onworks:~$ 
```

9. In the home directory, create a directory *MCA2024*. Inside the *MCA2024,* create another directory *<FistName_ Section_ ClassRoll>* and get into the directory [**~/MCA2024/Ankur_A_00$**].



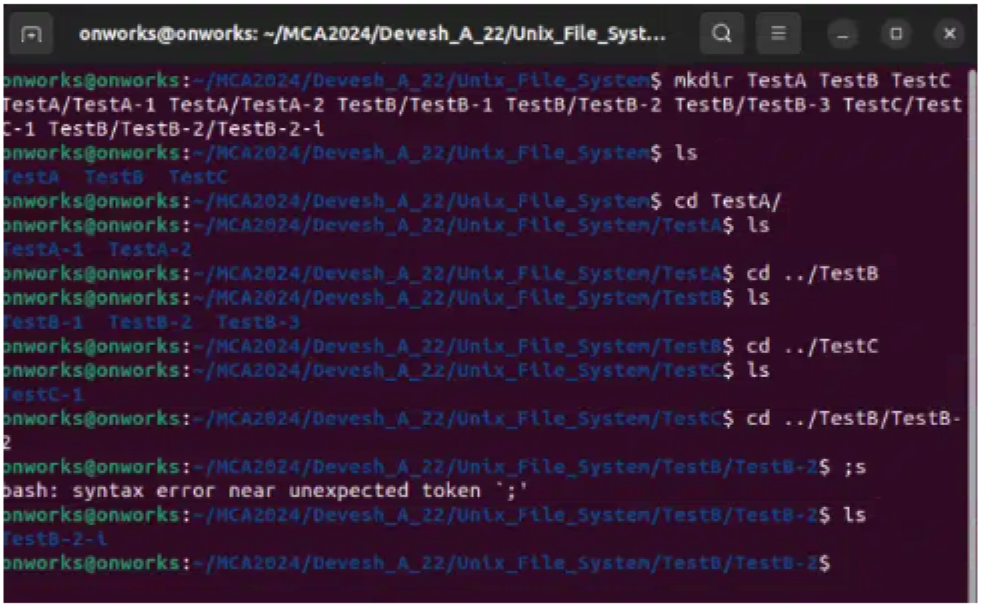
```
onworks@onworks:~$ mkdir MCA2024
onworks@onworks:~$ cd MCA2024/
onworks@onworks:~/MCA2024$ mkdir Devesh_A_22
onworks@onworks:~/MCA2024$ cd Devesh_A_22/
onworks@onworks:~/MCA2024/Devesh_A_22$ 
```

10. Go to the subdirectory and create another subdirectory "**Unix_File_System**" within it.



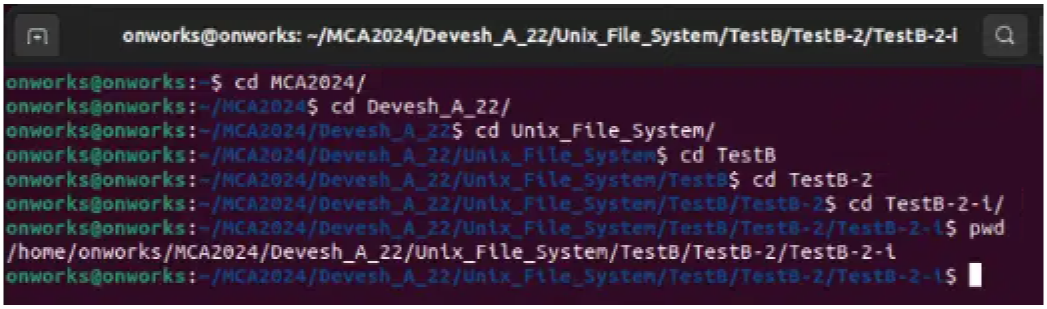
```
onworks@onworks:~$ mkdir MCA2024
onworks@onworks:~$ cd MCA2024/
onworks@onworks:~/MCA2024$ mkdir Devesh_A_22
onworks@onworks:~/MCA2024$ cd Devesh_A_22/
onworks@onworks:~/MCA2024/Devesh_A_22$ mkdir Unix_File_System
onworks@onworks:~/MCA2024/Devesh_A_22$ ls
Unix_File_System
onworks@onworks:~/MCA2024/Devesh_A_22$ 
```

11. Create the subdirectories `TestA, TestB, TestC` and corresponding sub-subdirectories `TestA-1, TestA-2, TestB-1, TestB-2, TestB-3, TestC-1, TestB-2-i` in a single command.



```
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System$ mkdir TestA TestB TestC
TestA/TestA-1 TestA/TestA-2 TestB/TestB-1 TestB/TestB-2 TestB/TestB-3 TestC/Test
C-1 TestB/TestB-2/TestB-2-i
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System$ ls
TestA  TestB  TestC
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System$ cd TestA/
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestA$ ls
TestA-1  TestA-2
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestA$ cd ../TestB
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB$ ls
TestB-1  TestB-2  TestB-3
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB$ cd ../TestC
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestC$ ls
TestC-1
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestC$ cd ../TestB/TestB-
2
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2$ ;s
bash: syntax error near unexpected token `;'
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2$ ls
TestB-2-i
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2$
```

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


onworks@onworks: ~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2/TestB-2-i

```
onworks@onworks:~$ cd MCA2024/
onworks@onworks:~/MCA2024$ cd Devesh_A_22/
onworks@onworks:~/MCA2024/Devesh_A_22$ cd Unix_File_System/
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System$ cd TestB
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB$ cd TestB-2
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2$ cd TestB-2-i/
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2/TestB-2-i$ pwd
/home/onworks/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2/TestB-2-i
onworks@onworks:~/MCA2024/Devesh_A_22/Unix_File_System/TestB/TestB-2/TestB-2-i$
```