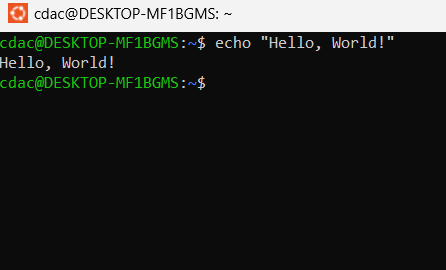
**Part A**

What will the following commands do?

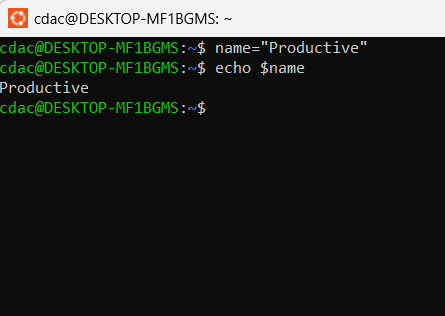
1. echo "Hello, World!"

→ It is going to print "Hello, World!"

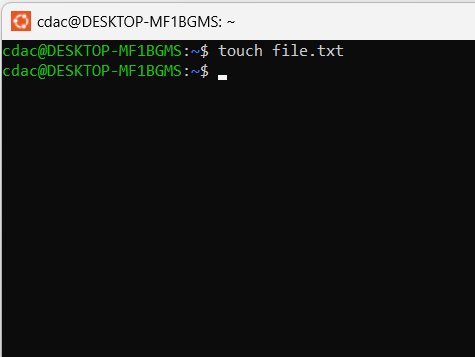


2. name="Productive"

→ Here name is considered as variable name, after accessing that variable using **echo** command It will directly print Productive.

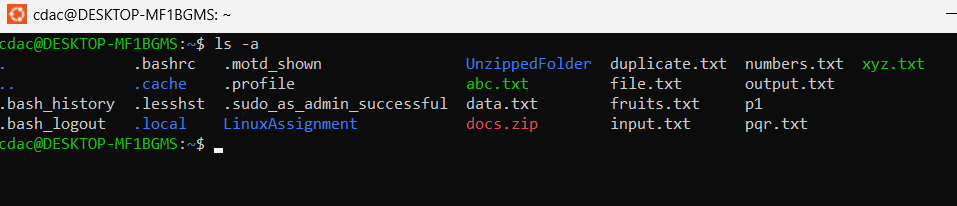


3. touch file.txt

→

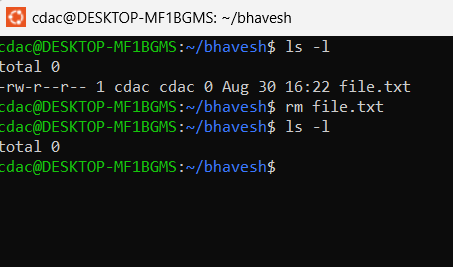
4. ls -a

→ ls gives us the list and -a includes all the files, i.e hidden files also.



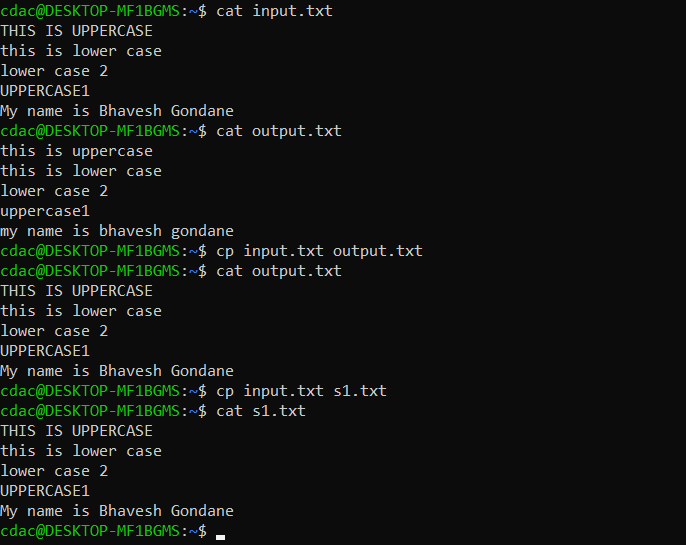
5. rm file.txt

→ It removes the file.



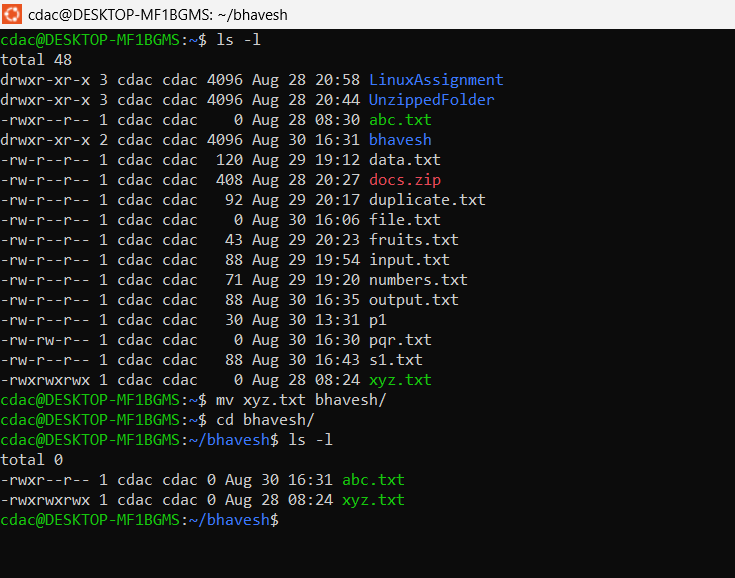
6. cp file1.txt file2.txt

→ It copies the text/data from file1.txt to file2.txt



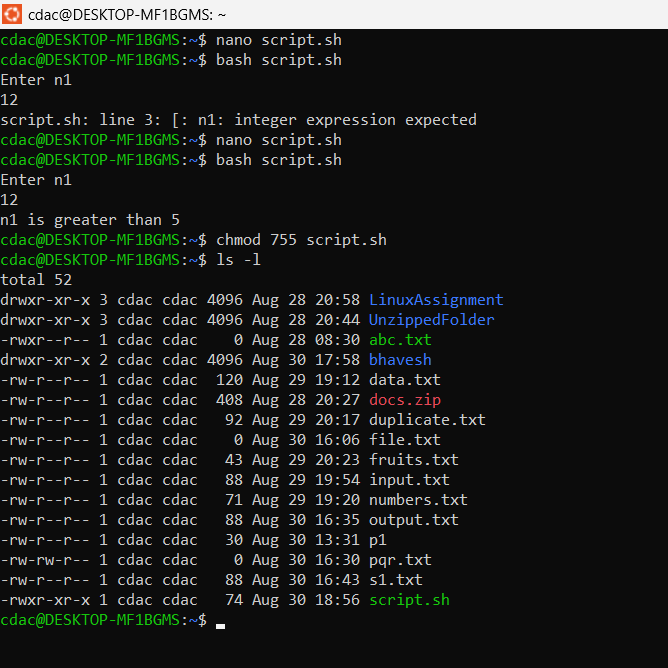
7. mv file.txt /path/to/directory/

→ It moves the file.txt file to the destination folder.



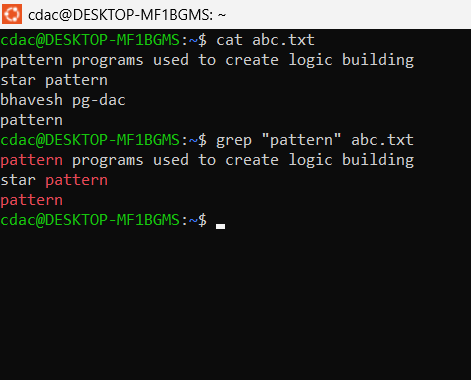
8. chmod 755 script.sh

→ It changes permissions of script.sh. 7 tells us that OWNER permission to be changed to read write and execute. 5 tells us that GROUP permission to be change to read and execute. And last 5 also tells us that OTHER’s permission to be changed to read and execute.



9. grep "pattern" file.txt

→ grep command is used to get some String pattern outputs from a file. Here it will list all the lines containing “pattern” word in file.txt



10. kill PID

→ kiil PID usually kills i.e terminates the process with the specified process id.

11. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

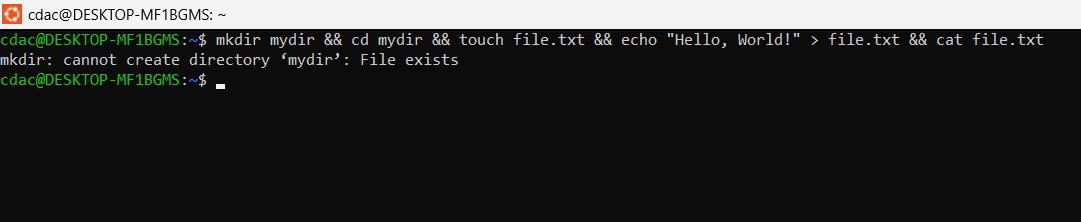
→ Here mkdir creates a new directory named mydir. && operator is used to make sure that the previous command is executed successfully, if it is executed the only it goes forward.

Next command changes the current directory to mydir,

then new file.txt is created is not exists and if it exists the it renews it’s timestamp .

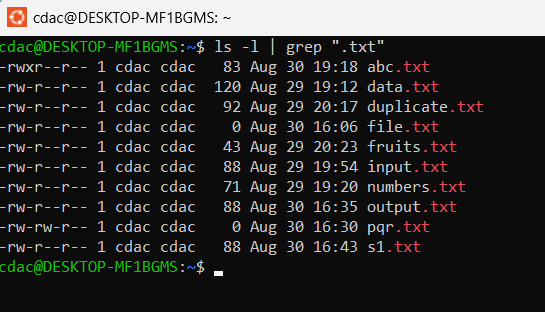
Then "Hello, World!" is added to the file.txt

And finally "Hello, World!" is displayed using cat command.



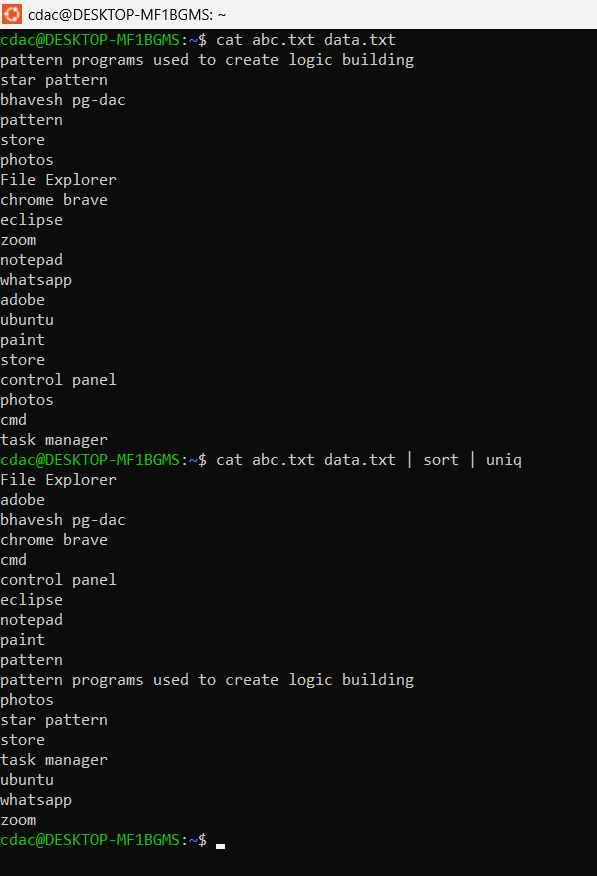
12.ls -l | grep ".txt"

→ It lists down all the files having .txt



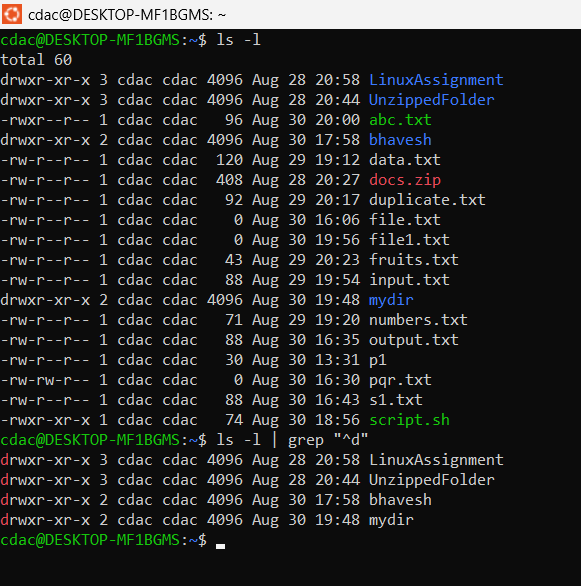
13. cat file1.txt file2.txt | sort | uniq

→ First it displays all the ccontent of file1 and file2 then it sort it out according to alphabetical order and finally it eliminates the repeated lines.



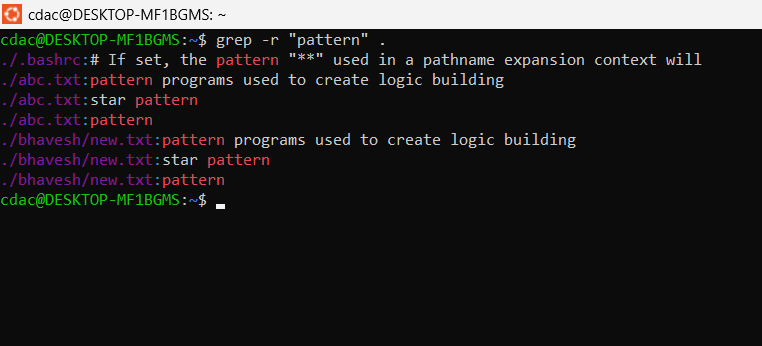
14.ls -l | grep "^d"

→ It will first list down all the files and directories and the it will display only the files and directories with start letter d.



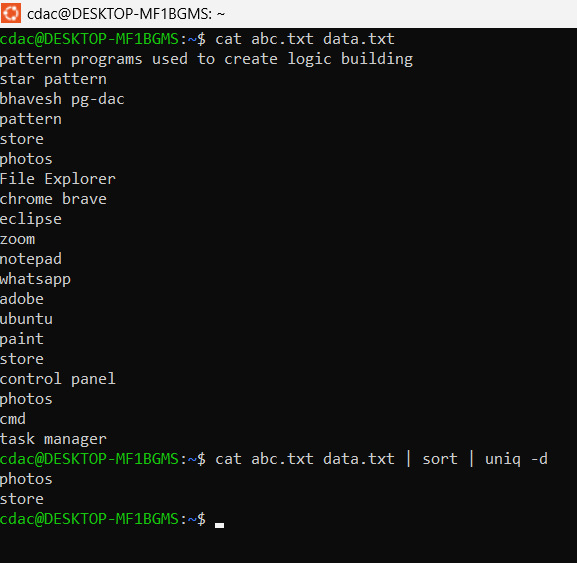
15. grep -r "pattern" /path/to/directory/

→ It will list out all the lines with the word pattern exclusively/recursively in that directory.



16. cat file1.txt file2.txt | sort | uniq –d

→ Contents of file1 and file 2 will be displayed and then the displayed content will be sorted in the alphabetical order and then only the duplicate lines will be displayed.



17. chmod 644 file.txt

→ It changes the permissions of owner to read and write, group to read and other to read of file.txt

Code Execution:-

cdac@DESKTOP-MF1BGMS:~$ chmod 644 abc.txt

cdac@DESKTOP-MF1BGMS:~$ ls -l

total 60

drwxr-xr-x 3 cdac cdac 4096 Aug 28 20:58 LinuxAssignment

drwxr-xr-x 3 cdac cdac 4096 Aug 28 20:44 UnzippedFolder

-rw-r--r-- 1 cdac cdac 96 Aug 30 20:00 abc.txt

drwxr-xr-x 2 cdac cdac 4096 Aug 30 20:17 bhavesh

-rw-r--r-- 1 cdac cdac 120 Aug 29 19:12 data.txt

18. cp -r source\_directory destination\_directory

→ It copies the source\_directory recursively i.e all the sub file and folders are also copied to the destination\_directory.

Code Execution:-

cdac@DESKTOP-MF1BGMS:~$ cd bhavesh/

cdac@DESKTOP-MF1BGMS:~/bhavesh$ ls

abc.txt gondane new.txt xyz.txt

cdac@DESKTOP-MF1BGMS:~/bhavesh$ cd ..

cdac@DESKTOP-MF1BGMS:~$ cp -r bhavesh/ chaitu/

cdac@DESKTOP-MF1BGMS:~$ ls

LinuxAssignment abc.txt chaitu docs.zip file.txt fruits.txt mydir output.txt pqr.txt script.sh

UnzippedFolder bhavesh data.txt duplicate.txt file1.txt input.txt numbers.txt p1 s1.txt

cdac@DESKTOP-MF1BGMS:~$ cd chaitu/

cdac@DESKTOP-MF1BGMS:~/chaitu$ ls

abc.txt gondane new.txt xyz.txt

[cdac@DESKTOP-MF1BGMS](mailto:cdac@DESKTOP-MF1BGMS):~/chaitu$

19. find /path/to/search -name "\*.txt"

→ It finds the .txt files in that particular folder

Code Execution:-

cdac@DESKTOP-MF1BGMS:~$ find . -name "\*.txt"

./numbers.txt

./output.txt

./file.txt

./s1.txt

./file1.txt

./input.txt

./pqr.txt

./fruits.txt

./chaitu/xyz.txt

./chaitu/abc.txt

./chaitu/new.txt

./duplicate.txt

./LinuxAssignment/file1.txt

./LinuxAssignment/docs/file2.txt

./LinuxAssignment/docs/file1.txt

./UnzippedFolder/LinuxAssignment/docs/file2.txt

./abc.txt

./data.txt

./mydir/file.txt

./bhavesh/xyz.txt

./bhavesh/abc.txt

./bhavesh/new.txt

cdac@DESKTOP-MF1BGMS:~$

20. chmod u+x file.txt

→ It changes permision of owner to execute only of file.txt

Code Execution:-

cdac@DESKTOP-MF1BGMS:~$ chmod u+x abc.txt

cdac@DESKTOP-MF1BGMS:~$ ls -l

total 64

drwxr-xr-x 3 cdac cdac 4096 Aug 28 20:58 LinuxAssignment

drwxr-xr-x 3 cdac cdac 4096 Aug 28 20:44 UnzippedFolder

---x------ 1 cdac cdac 96 Aug 30 20:00 abc.txt

21. echo $PATH

→ It will print the value of the path variable.

Code Exceution:-

cdac@DESKTOP-MF1BGMS:~$ path=100

cdac@DESKTOP-MF1BGMS:~$ echo $path

100

[cdac@DESKTOP-MF1BGMS](mailto:cdac@DESKTOP-MF1BGMS):~$

**Part B**

Identify True or False:

1. ls is used to list files and directories in a directory.

→ True

2. mv is used to move files and directories.

→ True (Also used to rename)

3. cd is used to copy files and directories.

→ Flase

4. pwd stands for "print working directory" and displays the current directory.

→False

5. grep is used to search for patterns in files.

→True

6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute

permissions to group and others.

→True

7. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1

if directory1 does not exist.

→True

8. rm -rf file.txt deletes a file forcefully without confirmation.

→ True

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

→ Incorrect

2. cpy is used to copy files and directories.

→ Incorrect

3. mkfile is used to create a new file.

→ Incorrect

4. catx is used to concatenate files.

→ Incorrect

5. rn is used to rename files.

→ Incorrect

**Part C**

Question 1: Write a shell script that prints "Hello, World!" to the terminal.

A screenshot of a computer screen

Description automatically generated

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the

value of the variable.

A screenshot of a computer

Description automatically generated

Question 3: Write a shell script that takes a number as input from the user and prints it.

A screenshot of a computer program

Description automatically generated

Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the

result.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise

prints "Odd".

A computer screen shot of a computer code

Description automatically generated

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

A computer screen with green and white text

Description automatically generated

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

A screenshot of a computer program

Description automatically generated

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it

does, print "File exists", otherwise, print "File does not exist".

A computer screen shot of a black screen

Description automatically generated

Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and

prints a message accordingly.

A screenshot of a computer program

Description automatically generated

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers

from 1 to 5. The output should be formatted nicely, with each row representing a number and each

column representing the multiplication result for that number.

A screenshot of a computer

Description automatically generated

Question 11: Write a shell script that uses a while loop to read numbers from the user until the user enters

a negative number. For each positive number entered, print its square. Use the break statement to exit the

loop when a negative number is entered.

A screenshot of a computer program

Description automatically generated

**Part E**

1. Consider the following processes with arrival times and burst times:

| Process | Arrival Time | Burst Time |

|---------|--------------|------------|

| P1 | 0 | 5 |

| P2 | 1 | 3 |

| P3 | 2 | 6 |

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.

A screenshot of a graph

Description automatically generated

2. Consider the following processes with arrival times and burst times:

| Process | Arrival Time | Burst Time |

|---------|--------------|------------|

| P1 | 0 | 3 |

| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

A screenshot of a computer

Description automatically generated

3. Consider the following processes with arrival times, burst times, and priorities (lower number

indicates higher priority):

| Process | Arrival Time | Burst Time | Priority |

|---------|--------------|------------|----------|

| P1 | 0 | 6 | 3 |

| P2 | 1 | 4 | 1 |

| P3 | 2 | 7 | 4 |

| P4 | 3 | 2 | 2 |

Calculate the average waiting time using Priority Scheduling.

A screenshot of a computer

Description automatically generated

4. Consider the following processes with arrival times and burst times, and the time quantum for

Round Robin scheduling is 2 units:

| Process | Arrival Time | Burst Time |

|---------|--------------|------------|

| P1 | 0 | 4 |

| P2 | 1 | 5 |

| P3 | 2 | 2 |

| P4 | 3 | 3 |

Calculate the average turnaround time using Round Robin scheduling.

A screenshot of a computer

Description automatically generated

5. Consider a program that uses the fork() system call to create a child process. Initially, the parent

process has a variable x with a value of 5. After forking, both the parent and child processes

increment the value of x by 1.

What will be the final values of x in the parent and child processes after the fork() call?

🡪 The final values of x in the parent and child processes after the fork() call will be **x=6** for parent process as well as for child process, since when the fork( ) method is used both the parent and child process executes same task.