Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

- echo "Hello, World!" : Prints Hello World!
- name="Productive" : Creates variable ‘name’ and assigns it the value ‘Productive’
- touch file.txt : Creates file names as file.txt
- ls -a : Lists **all files and directories**, including hidden ones (files starting with . and . .)
- rm file.txt : Deletes the file file.txt.
- cp file1.txt file2.txt : Copies the contents of file1.txt into a new file file2.txt.
- mv file.txt /path/to/directory/ : Moves file.txt into the specified directory.
- chmod 755 script.sh : Changes permissions of script.sh to **rwxr-xr-x** (owner can read/write/execute; group and others can read/execute).
- grep "pattern" file.txt : Searches for the word pattern inside file.txt and prints matching lines.
- kill PID : Terminates the process with the given **process ID**.
- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt : Creates a new directory mydir. Moves into it. Creates a file file.txt. Writes Hello, World! into the file. Displays its content (Hello, World!).
- ls -l | grep ".txt" : Lists files in **long format**(display all info) and filters the output to only show entries containing .txt.
- cat file1.txt file2.txt | sort | uniq : Concatenates file1.txt and file2.txt, sorts all lines, and removes duplicates.
- ls -l | grep "^d" : Lists files and filters only directories (since directory entries start with d).
- grep -r "pattern" /path/to/directory/ : Recursively searches for "pattern" in all files under the given directory.
- cat file1.txt file2.txt | sort | uniq –d : Concatenates files, sorts lines, and shows only **duplicate lines**
- chmod 644 file.txt : Sets file permissions to **rw-r--r--** (owner can read/write; group and others can only read).

- cp -r source_directory destination_directory : Copies an entire directory (source_directory) and its contents recursively into destination_directory
- find /path/to/search -name "*.txt" : Finds and lists all files ending with .txt under the given directory path.
- chmod u+x file.txt : Gives the file owner (u) execute permission for file.txt.
- echo $PATH : Prints the system's PATH environment variable (a list of directories where the shell looks for executables).

Part B

Identify True or False:

ls is used to list files and directories in a directory.	True
mv is used to move files and directories.	True
cd is used to copy files and directories.	False
pwd stands for "print working directory" and displays the current directory.	True
grep is used to search for patterns in files.	True
chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.	True
mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.	True
rm -rf file.txt deletes a file forcefully without confirmation.	True

Part C

Identify the Incorrect Commands:

1. chmodx is used to change file permissions → chmod
2. cpy is used to copy files and directories. → cp
3. mkfile is used to create a new file. → touch filename
4. catx is used to concatenate files. → cat
5. rn is used to rename files. → mv oldfilename newname

Part D

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

- rutuja@Rutuja:~$ echo "Hello, World!"
- Hello, World!

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

- rutuja@Rutuja:~$ vi que2.txt
- rutuja@Rutuja:~$ ./que2.txt
- -bash: ./que2.txt: Permission denied
- rutuja@Rutuja:~$ chmod u+x que2.txt
- rutuja@Rutuja:~$ cat que2.txt
- #!/bin/bash
- name="CDAC Mumbai"
- echo "The variable is: $name"
- rutuja@Rutuja:~$ ./que2.txt
- The variable is: CDAC Mumbai

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

- rutuja@Rutuja:~$ vi que3.txt
    - #!/bin/bash
    - echo -n "Enter the number: "
    - read num
    - echo " You Entered: $num "
- rutuja@Rutuja:~$ ./que3.txt
- -bash: ./que3.txt: Permission denied
- rutuja@Rutuja:~$ chmod u+x que3.txt
- rutuja@Rutuja:~$ ./que3.txt
- Enter the number: 1
- You Entered: 1

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

- rutuja@Rutuja:~$ vi que4.txt
- rutuja@Rutuja:~$ chmod u+x que4.txt
- rutuja@Rutuja:~$ cat que4.txt
    - #!/bin/bash
    - a=5
    - b=3

    - sum=$((a + b))
    - echo "Sum: $sum"
- rutuja@Rutuja:~$ ./que4.txt
- Sum: 8

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

- rutuja@Rutuja:~$ vi que5.txt
- rutuja@Rutuja:~$ chmod u+x que5.txt
- rutuja@Rutuja:~$ cat que5.txt
    - #!/bin/bash
    - echo -n "Enter a number: "
    - read num
    - if [$((num % 2)) -eq 0]
    - then
    - echo "Even"
    - else
    - echo "Odd"
    - fi
- rutuja@Rutuja:~$ ./que5.txt
- Enter a number: 2
- Even
- rutuja@Rutuja:~$ ./que5.txt
- Enter a number: 1
- Odd

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

- rutuja@Rutuja:~$ vi que6.sh
- rutuja@Rutuja:~$ chmod u+x que6.sh
- rutuja@Rutuja:~$ ./que6.sh
    - 1
    - 2
    - 3
    - 4
    - 5

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

- rutuja@Rutuja:~$ vi que7.sh
- rutuja@Rutuja:~$ cat que7.sh
    - #!/bin/bash
    - i = 1

    - while [$i -le 5]
    - do
    - echo $i
    - i=$((i+1))
    - done
- rutuja@Rutuja:~$ ./que7.sh
    - 1
    - 2
    - 3
    - 4
    - 5

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If itdoes, print "File exists", otherwise, print "File does not exist".

- rutuja@Rutuja:~$ vi que8.sh
- rutuja@Rutuja:~$ chmod u+x que8.sh
- rutuja@Rutuja:~$ ./que8.sh
- File exists
- rutuja@Rutuja:~$ cat que8.sh
- #!/bin/bash
    - if [-f "test.txt"]
    - then
    - echo "File exists"
    - else
    - echo "File does not exits"
    - fi

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.

- rutuja@Rutuja:~$ vi que9.sh
- rutuja@Rutuja:~$ cat que9.sh
    - #!/bin/bash
    - echo -n "Enter a number: "
    - read num
    - if [$num -gt 10]
    - then
    - echo "The number is greater than 10"
    - else
    - echo "The number is not greater than 10"
    - fi
- rutuja@Rutuja:~$ chmod u+x que8.sh
- rutuja@Rutuja:~$ chmod u+x que9.sh

- rutuja@Rutuja:~$ ./que9.sh
  - Enter a number: 8
  - The number is not greater than 10
- rutuja@Rutuja:~$ ./que9.sh
  - Enter a number: 20
  - The number is greater than 10

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5.

- rutuja@Rutuja:~$ vi que10.sh
- rutuja@Rutuja:~$ chmod u+x que10.sh
- rutuja@Rutuja:~$ cat que10.sh
  - #!/bin/bash
  - for i in {1..5}
  - do
  - for j in {1..5}
  - do
  - printf "%4d" $((i * j))
  - done
  - echo
  - done
- rutuja@Rutuja:~$ ./que10.sh
  - 1 2 3 4 5
  - 2 4 6 8 10
  - 3 6 9 12 15
  - 4 8 12 16 20
  - 5 10 15 20 25

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.

- rutuja@Rutuja:~$ vi que11.sh
- rutuja@Rutuja:~$ chmod u+x que11.sh
- rutuja@Rutuja:~$ ./que11.sh
- Enter a number (negative to quit): 3
- Square: 9
- Enter a number (negative to quit): 9
- Square: 81
- Enter a number (negative to quit): -1
- Exiting...
- rutuja@Rutuja:~$ cat que11.sh
  - #!/bin/bash

```
while true
do
echo -n "Enter a number (negative to quit): "
read num
if [ $num -lt 0 ]
then
echo "Exiting..."
break
fi
echo "Square: $((num * num))"
done
```