OS Assignment 2 28-02-2025 Pratik Wable

#Part A

// What will the following commands do?

1.echo "Hello, World!"

Ans: It will print the Hello, world!.
    echo is used to display string or text.

2.name="Productive"

Ans: To set a variable name called Productive.

3.touch file.txt

Ans: It will create a empty file called file.txt

4.ls -a

Ans: It will list out the "ALL" files from that directory.

5.rm file.txt

Ans: It will remove the file.txt from directory.

6. cp file1.txt file2.txt

Ans: Content of file1.txt will be copied into file2.txt.

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

Ans: To move file.txt to another destination directory.

8. chmod 755 script.sh

Ans: To change permission of script.sh file

9. grep "pattern" file.txt

Ans: It will search pattern word in file.txt.

10. kill PID

Ans: kill is used to terminate the processes.
here Process with Process id PID will be terminated.

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

Ans: crete mydir directory.
Go to the path of mydir directory.
create a empty file called file.txt.
Write Hello, World! in file.txt.
And lastly show the content of file.txt

12. ls -l | grep ".txt"

Ans: This will filter out the files with the extension of .txt from that directory
And will show the permissions, Modification & Access

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

Ans: It will concatenate the file1.txt and file2.txt
Then output will be sort out alphabetically.
and finally Duplicate lines will be Removed.

14. ls -l | grep "^d"

Ans: This will filter out the files with the extension of ^d from that directory
And will show the permissions, Modification & Access

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

Ans: This will search pattern recursively in all files and subdirectories from the mentioned path.

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

Ans: It will concatenate the file1.txt and file2.txt
 Then output will be sort out alphabetically.
 And only the Duplicate lines will be Displayed from the file.

17. chmod 644 file.txt

Ans: chmod used to change the permission of file.txt
 for User to read and write only.
 for group to read only.
 for other to read only.

18. cp -r source_directory destination_directory

Ans: This will copy the entire content like directory, sub-directory and files to the source dire

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

Ans: to find the files with extension of *.txt in mentioned path.

20. chmod u+x file.txt

Ans: we can use symbols also instead of numbers.
 here u+x means user is permitted to only execute only.

21. echo $PATH

Ans: This is used to display the current value of the PATH environment variable.

// 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

3. cd is used to copy files and directories. // False

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

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, // True
and read and execute permissions to group and others.

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 Wrong | chmod is correct to change file permissions

2. cpy is wrong | cp is correct to copy files and directories.

3. mkfile is wrong | touch is correct to create a new file.

4. catx is wrong | cat is correct to concatenate files

5. rn is wrong | mv is correct to rename files.

// Part C

Que1: Write a shell script that prints "Hello, World!" to the terminal.

```
nano

echo Hello, World!

save the file helloworld.sh
chmod +x helloworld.sh
./helloworld.sh
```

Que2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

nano

```
name="CDAC Mumbai"
echo $name
```

saved the file name.sh

```
chmod +x name.sh
./name.sh
```

Que3: Write a shell script that takes a number as input from the user and prints it.

nano

```
echo "Enter the number"
read num
echo "You have entered: $num"
```

saved the file readnum.sh

```
chmod +x readnum.sh
./readnum.sh
```

Que4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

nano

```
echo "Enter first Number"
read FirstNumber

echo "Enter second Number"
read SecondNumber

sum=$((FirstNumber + SecondNumber))
echo "The addition is: $sum"
```

saved the file addition.sh

```
chmod +x addition.sh
./addition.sh
```

Que5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

nano

```
echo "Please enter the number"
read Num

if [ $(($Num % 2)) -eq 0 ]; then
   echo "$Num is even number"
else
   echo "$Num is odd number"
fi
```

```
saved the file evennum.sh
chmod +x evennum.sh
./evennum.sh
```

Que6: Write a shell script that uses a for loop to print numbers from 1 to 5.

```
nano

for i in {1..5}
do
   echo $i
done
```

```
saved the file forloop.sh
chmod +x forloop.sh
./forloop.sh
```

Que7: Write a shell script that uses a while loop to print numbers from 1 to 5.

```
nano

i=1

while [ $i -le 5 ]
do
  echo $i
  i=$((i+1))
done
```

```
saved the file while.sh
chmod +x while.sh
./while.sh
```

Que8: 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".

```
nano
```

```
if [ -f "file.txt" ]; then
  echo "File Do Exist"
else
  echo "Not Exist"
fi
```

saved the file doesexist.sh
chmod +x doesexist.sh
./doesexist.sh

Que9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

nano

```
echo "Enter the  number"
read Num

if [ $Num -ge 10 ]; then
  echo "$Num is greater than 10"
else
  echo "$Num is less than 10"
fi
```

saved the file numten.sh
chmod +x numten.sh
./numten.sh

Que10: 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.

nano

```
for i in {1..5}
do
  for j in {1..10}
  do
    echo "$i X $j = $((i * j))"
  done
done
```

Saved the file Multi.sh

```
chmod +x Multi.sh
./Multi.sh
```

Que11: 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.

```
nano

while true
do
  echo "Please enter a number:"
  read number

  if [ $number -lt 0 ]; then
    echo "Negative number entered. Exiting..."
    break
  fi

  square=$((number * number))
  echo "The square of $number is $square"
done
```

#Part D

Que. 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?

Ans: Final value of Parents(x) is 6.also Final value of Child also 6.

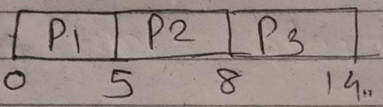
Page No.
Date

Q1 fcfs Avg waiting Time ?

→

Process	AT	BT
P1	0	5
P2	1	3
P3	2	6

Gant chart



* waiting Time = START − Arrival

P1 = 0 − 0 = 0

P2 = 5 − 1 = 4

P3 = 8 − 2 = 6

∴ 6 + 4 + 0 = 10

∴ Avg Waiting Time = $\frac{10}{3}$ ≈ 3.33

Q2 SJF , Avg TAT

→ Process Arrival Burst

Process	Arrival	Burst
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Gant chart

P1	P1	P1	P3	P4	P4	P4	P4	P2	P2	P3	P2	P2

0 1 2 3 4 5 6 7 8 9 10 11 12 13

∴ TAT = completion − Arrival

$P_1 = 3 - 0 = 3$

$P_2 = 13 - 1 = 12$

$P_3 = 4 - 2 = 2$

$P_4 = 8 - 3 = 5$

Avg TAT ⇒ $12 + 2 + 3 + 5 = \frac{22}{4} = \underline{5.5}$

Page No.

Date

Q3 priority schedulling Algor

	AT	BT	priority
P1	0	6	3 ×
P2	1	4	1 ×
P3	2	7	4
P4	3	2	2 ×

Gant chart

P1	P2	P~~3~~4	P3

0 6 10 12 16

× waiting Time = ST - AT

P1 = 0 - 0 = 0

P2 = 6 - 1 = 5 Avg = $\frac{22}{4}$ = 5.5

P3 = 12 - 2 = 10

P4 = 10 - 3 = 7

Q4 Round Robin Sche, ATAT = ?

→

	Process	AT	BT
2	P1	0	4
2	P2	1	5
0	P3	2	2
1	P4	3	3

Time unit = 2 units

Gant chart

~~P1~~	~~P1~~	~~P2~~	~~P2~~	~~P3~~	~~P3~~	~~P4~~	~~P4~~	~~P4~~	
~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~

P1	P1	P2	P2	P3	P3	P4	P4	P1	P1	P2	P2	P4	
0	1	2	3	4	5	6	7	8	9	10	11	12	13

P2	
13	14

TAT = CT − AT

∴ $P_1 = 10 - 0 = 10$

$P_2 = 14 - 1 = 13$

$P_3 = 6 - 2 = 4$

$P_4 = 13 - 3 = 10$

∴ Avg TAT = 10 + 10 + 13 + 4

$= \frac{37}{4} = 9.25$