CDAC MUMBAI

Concepts of Operating System

Assignment 2

Part A

What will the following commands do?

- echo "Hello, World!"
  prints Hello, World!

- name="Productive"
  assigns Productive to “name” variable

- touch file.txt
  creates a file named “file.txt”

- ls –a
  use to list all the files and directories including the hidden one

- rm file.txt
  removes the file named “file.txt”

- cp file1.txt file2.txt
  copies the content of “file1.txt” to “file2.txt”

- mv file.txt /path/to/directory/
  move “file.txt” to the given path

- chmod 755 script.sh
  changes the permissions of the file “script.sh”

- grep "pattern" file.txt
  grep is used to search for “pattern” in the texts of file named “file.txt”

- kill PID
  use to sends signal to “PID” to teminate the process id

- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
  It creates a directory named “mydir” then it creates a file named “file.txt” prints and displays Hello, World! in the file using echo and then displays the content of the “file.txt”

- ls -l | grep ".txt"
  is use to display the files with .txt extension

- cat file1.txt file2.txt | sort | uniq
  is use to concatenate these two files and then sort it alphabetically and with “uniq” command removes the duplicate line

- ls -l | grep "^d"

is use to filter the alphabet “d” in the list of files and folders

- grep -r "pattern" /path/to/directory/
  recursive pattern in directory

- cat file1.txt file2.txt | sort | uniq –d
  is use to concatenate the content of two files, sort it and with uniq –d removes the repeated lines

- chmod 644 file.txt
  provides read and write permission to owner and only read permission to group and other users

- cp -r source_directory destination_directory
  copy content from source to destination directory

- find /path/to/search -name "*.txt"
  is use to search .txt within the “/path/to/search”

- chmod u+x file.txt
  use to give execute permission to the user within “file.txt”

- echo $PATH
  is use to print the value of “PATH”

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, 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.
   chmod

2. **cpy** is used to copy files and directories.
   cp

3. **mkfile** is used to create a new file.
   touch filename.txt

4. **catx** is used to concatenate files.
   cat filename1.txt > cat filename2.txt

5. **rn** is used to rename files.
   mv f1.txt f2.txt, this renames the file f1.txt to f2.txt

Part C

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

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.
vi f1.sh

#!/bin/bash
name="CDAC Mumbai"
echo "$name"

sh f1.sh

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

#!/bin/bash
read num
echo "enter a number: $num"

sh n.sh

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

result.

```
#!/bin/bash
read X
read Y

echo "$(( $X + $Y ))"
```

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

```
#!/bin/bash

echo "Please enter a number:"

read num

if [ $((number % 2)) -eq 0 ]; then
   echo "$num is Even"
else
   echo "$num is Odd"
fi
```

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

```
#!/bin/bash

for((i = 1; i<6; i++))
do
    echo $i
done
```

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

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".

```
#!/bin/bash

if [-e "file.txt"]; then
echo "file exists"
else
echo "file does not exists"
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.

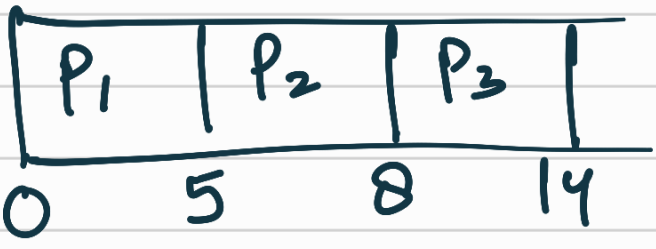
```
#!/bin/bash

echo "Enter a num: "
read num

if [ $num -gt 10 ]; then
echo "It greater than 10."
else
echo "It is not greater than 10."
fi
```

PART-E

1. First come, First served (FCFS)

Process	Arrival time	Burst time	Comp time	TAT	WT
P_1	0	5	5	5	0
P_2	1	3	8	7	4
P_3	2	6	14	12	6

Gantt chart



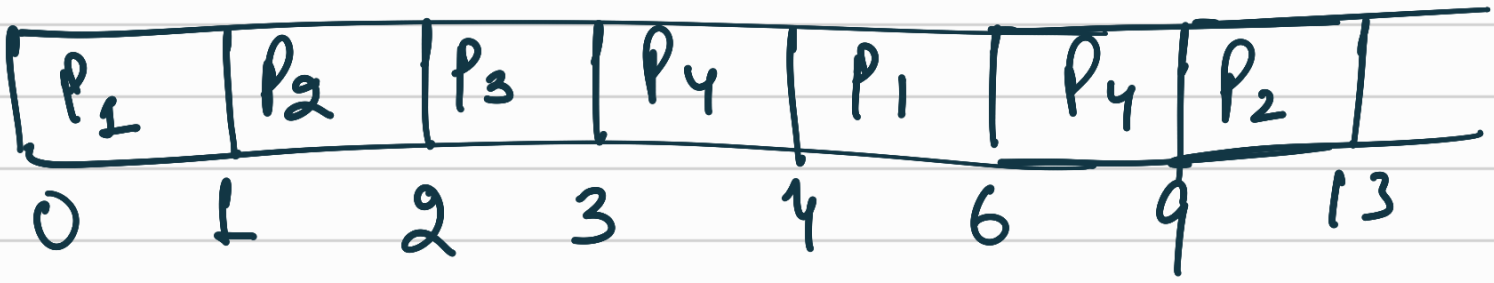
Average WT = $\frac{0+4+6}{3}$

$= \frac{10}{3} = 3.333$

2. Shortest Job First (SJF)

Process	Arrival T.	Burst T.	Comple T.	TAT
P_1	0	~~8~~ ~~2~~ 0	6	6
P_2	1	~~8~~ 4	13	12
P_3	2	~~1~~ 0	3	1
P_4	3	~~4~~ ~~3~~ 0	9	6

Gantt chart

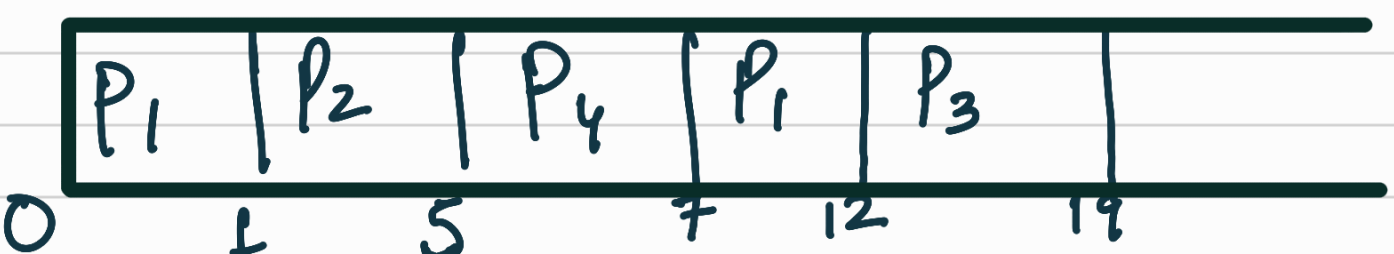


$$\text{Avg TAT} = \frac{6+12+1+6}{4} = \frac{25}{4} = 6.25.$$

Q3 Priority schedule

Lower number = Higher Priority.

Process	Arrival Time	Burst Time	Priority	Comp T.	TAT	WT
P_1	0	$\cancel{6}\ 5$	3	12	12	6
P_2	1	4	1	5	4	0
P_3	2	7	4	19	17	10
P_4	3	2	2	7	4	2



Avg waiting Time :- $\frac{6+10+2+0}{4}$

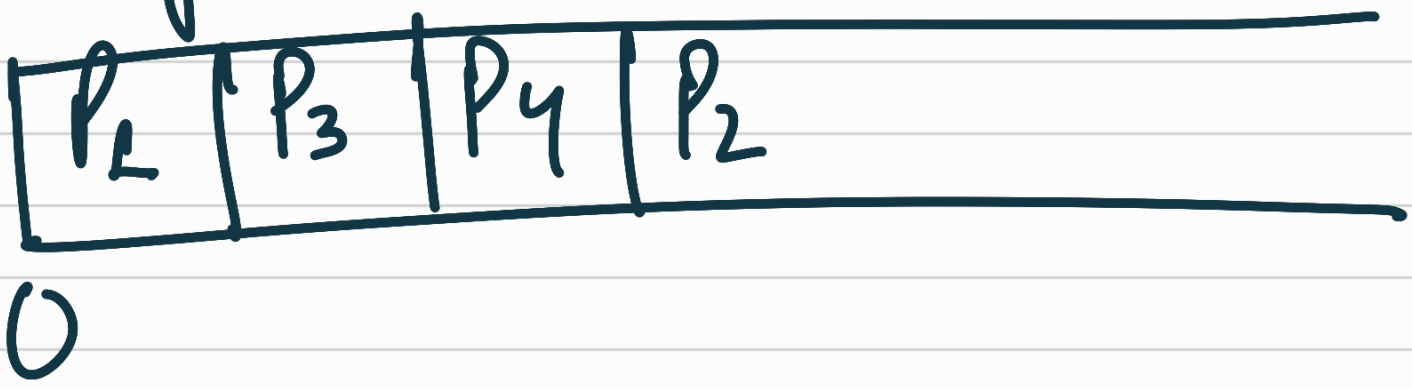
$= \frac{18}{4} = 4.5$

Ques 4. Round Robin

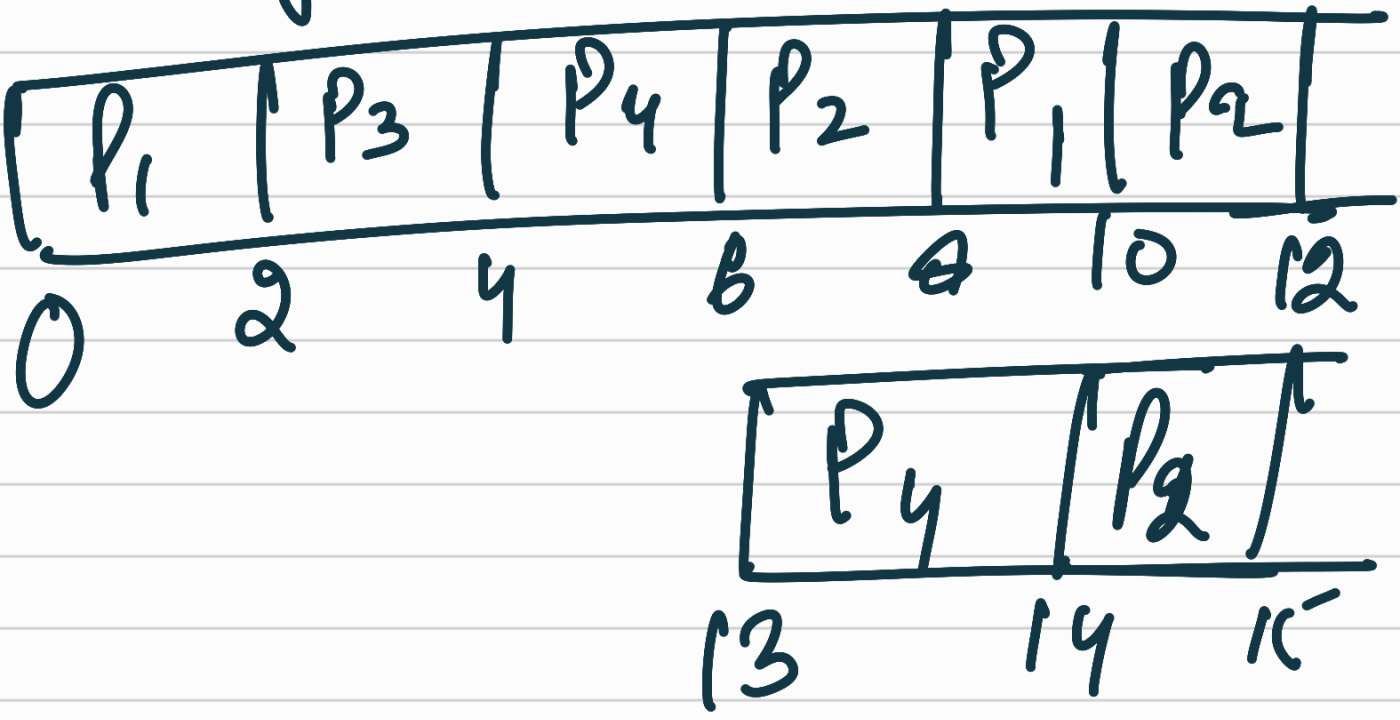
Time Quantum :- 2

Process	Arrival Time	Burst time	Compl Time	TAT	WT
P_1	0	~~4~~ 2 0	10	10	
P_2	1	~~5~~ 3 1	15	14	
P_3	2	~~2~~	4	2	
P_4	3	~~3~~ 1	14	11	

Ready Queue



Running Queue



$$\text{Avg TAT} = \frac{10+14+2+11}{4} = \frac{37}{4}$$

$$= 9.25$$

Ques 5