Name: Nimra Nasir
Reg No: 19-CP-35

Operating System (Lab)

Final Exam

Questions:

Question: 2

Part: 5

Preemptive Scheduling	Non-preemptive Scheduling
• The resources are allocated to a process for a limited time.	• once resources are allocated to a process, the process holds it till it completes its burst time or switches to waiting state.
• process can be interrupted in between.	• Process cannot be interrupted till it terminates or switches to waiting state.
• If a high priority process frequently arrives in the ready queue, low priority process may starve.	• If a process with long burst time is running CPU, then another process with less CPU burst time may starve.
• preemptive scheduling has overheads of scheduling the processes.	• Non-preemptive scheduling does not have overheads.
• preemptive scheduling is flexible.	• Non-preemptive scheduling is rigid.
• Preemptive scheduling is cost associated.	• Non-preemptive scheduling is not cost associative.
• **Example:** SRTF, LRTF, RR etc.	• **Example:** FCFS, SJF, HRRN etc.

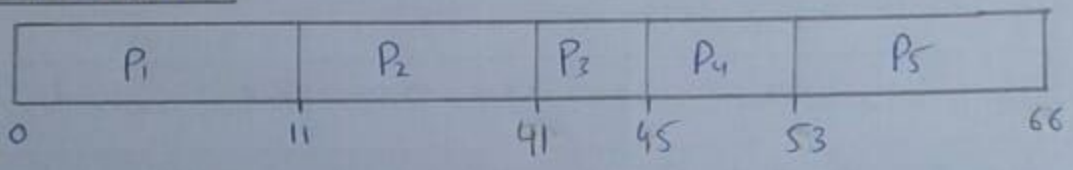
19-CP-35

Process	Burst Time (ms)	Priority
P_1	11	3
P_2	30	1
P_3	4	3
P_4	8	4
P_5	13	2

- Turn Around time = completion − Arrival
- waiting time = Turn around − Burst time

1) First come First serve (FCFS):

Gantt chart:



Process	Burst time	Completion time	Arrival time	Turn Around Time	waiting time
A	11	11	0	11	0
P_2	30	41	0	41	11
P_3	4	45	0	45	41
P_4	8	53	0	53	45
P_5	13	66	0	66	53

Average waiting time = $\frac{0+11+41+45+53}{5}$

$= 150/5$

$= 30$

Average Turn around time = $\frac{11+41+45+53+66}{5}$

$= 216/5$

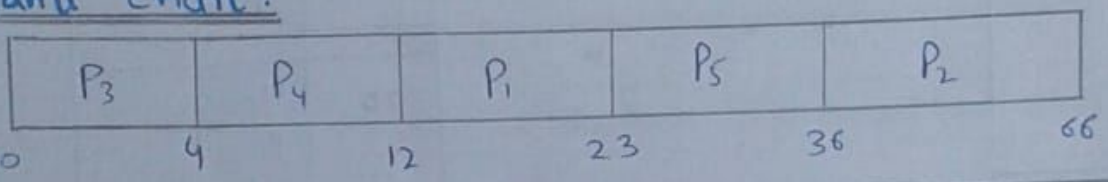
$= 43.2$

19-CP-35

Process	Burst time (ms)	Priority
P_1	11	3
P_2	30	1
P_3	4	3
P_4	8	4
P_5	13	2

2) Shortest Job First (SJF Non-preemptive):

Gantt chart:



Process	Burst time	Completion time	Arrival Time	Turn Around Time	waiting time
P_1	11	23	0	23	12
P_2	30	66	0	66	36
P_3	4	4	0	4	0
P_4	8	12	0	12	4
P_5	13	36	0	36	23

Average Turn Around time $= \frac{23+66+4+12+36}{5}$

$= 141/5$

$= 28.2$

Average waiting Time $= \frac{12+36+0+4+23}{5}$

$= 75/5$

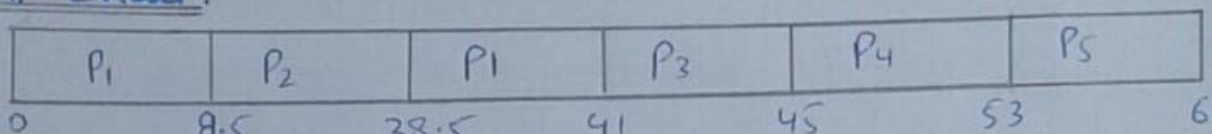
$= 15$

Process	Burst time (ms)	Priority
P_1	11	3
P_2	30	1
P_3	4	3
P_4	8	4
P_5	13	2

19-CP-35

3) Round Robin (R.R):

Gantt chart:



Process	Burst time	Priority	Completion time	Turn Around time	waiting time	Arrival time
P_1	11	3	41	41	30	0
P_2	30	1	38.5	38.5	8.5	0
P_3	4	3	45	45	41	0
P_4	8	4	53	53	45	0
P_5	13	2	66	66	53	0

Average Turn Around time $= \frac{41 + 38.5 + 45 + 53 + 66}{5}$

$= 48.7$

Average waiting time $= \frac{30 + 8.5 + 41 + 45 + 53}{5}$

$= 35.5$

Part: 4

The average waiting time of FCFS Algorithm = 30

The average waiting time of SJF Algorithm = 15

The average waiting time of RR = 35.5

Thus, the SJF algorithm has minimum waiting time.

19-CP-35

Question: 3

Answer:

0	a b c d
1	e f g h
2	i j k l
3	m n o p
4	q r s t
5	u v

Logical map

Page Number	Frame Number
0	7
1	26
2	52
3	20
4	55
5	6

Page table

	Contents
6	u v
7	a b c d
20	m n o p
26	e f g h
52	i j k l
53	q r s t

physical map

The physical address = page size x frame number + offset

Physical address of m = 4 x 20 + 0 = 80

Physical address of d = 4 x 7 + 3 = 31

Physical address of v = 4 x 6 + 1 = 25

Physical address of r = 4 x 55 + 1 = 221

The external fragmentation = 0

The internal fragmentation = 2

19-cp-35

Question: 1

Answer:

Process	Allocation					Request					Available				
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
P_0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1
P_1	1	1	0	0	0	0	0	1	0	1					
P_2	0	0	0	1	0	0	0	0	0	1					
P_3	0	0	0	0	0	1	0	1	0	1					

1) Need matrix:

Process	Need matrix					Available				
	A	B	C	D	E	A	B	C	D	E
P_0	0	1	0	0	1	0	0	0	0	1
P_1	0	0	1	0	1					
P_2	0	0	0	0	1					
P_3	1	0	1	0	1					

P_2 will be released first.

2) Total number of resources:

Total = Available + Allocation

For A = 0 + 2

= 2

For B = 0 + 1

= 1

For C = 0 + 1

= 1

19-CP-35

Now, total number of resources

A	B	C	D	E
2	1	1	2	1

3) **Resource type:**

The available resource type are A, B, C, D, E. The sum of total resources are 7.

Some are allocated some are not.

4) **The system is in safe state or not:**

Since available resources

A	B	C	D	E
0	0	0	0	1

only P_2 execute completely because E requires only 1 resource.

we know that, it should be available resource > need resource.

And we can see from need matrix that except P_2, all other processes P_0, P_1, P_3 have larger resource values than the available resource.

So, P_0, P_1, P_3, are in deadlock situation and also in unsafe state.

The processes cannot execute completely.

5) **can request for (1,1,0,0,0) by P_2 granted:**

No, it is not granted as available resources are (0,0,0,0,1) and if we grant for P_2, as (1,1,0,0,0) then

Request > available resource

Thus, no granted

c) Can request for (0,1,0,1,0) by P_0 ?

Since, available resources are (0,0,0,0,1) and is requested for (0,1,0,1,0), then again

Requested > available

Thus not granted