OS paper

Section -A

MCQ’s

1) interrupt
2) Memmory related information
3) Wait
4) Asymmetric
5) ps
6) fg
7) jobs
8) Ctrl-C
9) all of these
10) usability
11) /lib
12) long term
13) do not
14) semaphore
15) spin lock
16) false
17) true
18) bounded waiting
19) Firmware based solution
20) Medium term schedular

Section – B

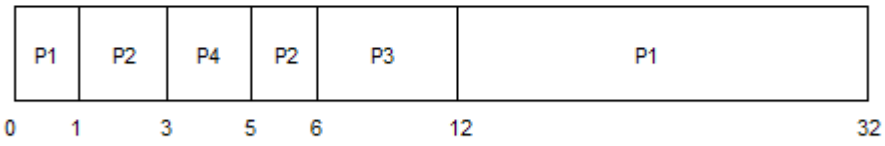
Q1:

Ans:

In Preemptive Shortest Job First Scheduling, jobs are put into ready queue as they arrive, but as a process with **short burst time** arrives, the existing process is preempted or removed from execution, and the shorter job is executed first.

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3

The GANTT chart for Preemptive Shortest Job First Scheduling will be,



The average waiting time will be, ((5-3) + (6-2) + (12-1))/4 = 4.25 ms

The average waiting time for preemptive shortest job first scheduling is less than both, non-preemptive SJF scheduling and FCFS scheduling.

As you can see in the **GANTT chart** above, as **P1** arrives first, hence it's execution starts immediately, but just after 1 ms, process **P2** arrives with a **burst time** of 3 ms which is less than the burst time of **P1**, hence the process **P1**(1 ms done, 20 ms left) is preemptied and process **P2** is executed.

As **P2** is getting executed, after 1 ms, **P3** arrives, but it has a burst time greater than that of **P2**, hence execution of **P2** continues. But after another millisecond, **P4** arrives with a burst time of 2 ms, as a result **P2**(2 ms done, 1 ms left) is preemptied and **P4** is executed.

After the completion of **P4**, process **P2** is picked up and finishes, then **P2** will get executed and at last **P1**.

The Pre-emptive SJF is also known as **Shortest Remaining Time First**, because at any given point of time, the job with the shortest remaining time is executed first.

Q2:

Ans:

If a process exits, then all of its threads are terminated as well, so then it is not possible for them to keep running.

Q3:

Ans:

As we know that by default threads share common code, data, and other resources. I think it is an advantage because it allows multiple tasks to be performed simultaneously in a single address space.