Name:- Sumit Kumar Yadav Roll No.: 18CS30042

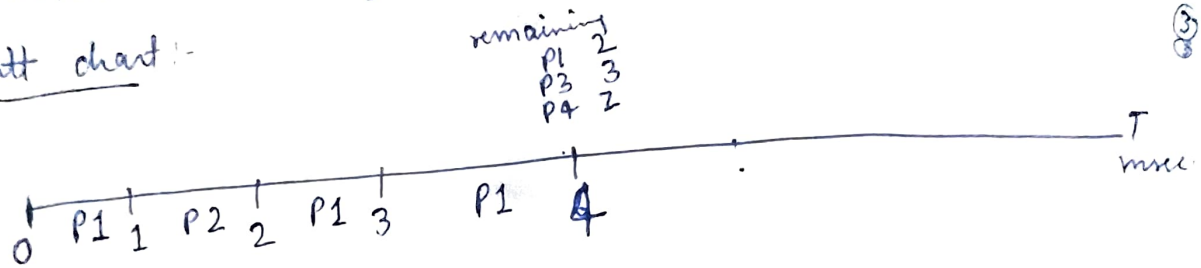
Ans(3:3)

Process	Arrival time	CPU burst time
P1	0	5
P2	1	1
P3	3	3
P4	4	2

average waiting time = 2 msec

∵ shortest remaining time first scheduling algo.

Grantt chart :-



$\because 2 = \frac{(\Sigma \text{ wait time})}{4} \Rightarrow \Sigma \text{ Wait time} = 8 \text{ msec}$

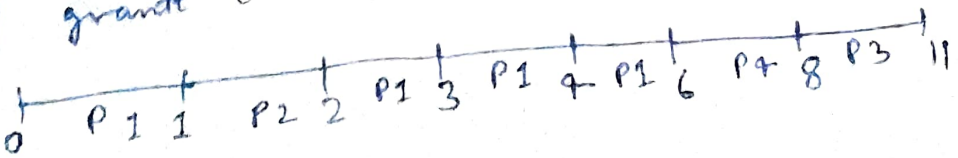
∴ before 4 sec., waiting time = 1 + 1 = 2 msec
only 6 waiting time is remaining and either we assign job to P1, P3 or P4 we get waiting assign duration 4-6 to P1 now at 6 msec P3 has 3 msec remaining and waiting time increases by 2(2) = 4.

∴ current waiting time = 6 msec.

now for 8 msec waiting time z must be 2 so that process P3 has 2 waiting time

∴ $\boxed{z = 2}$

final grantt chart :-



verification, waiting time = $\frac{P1 \quad P2 \quad P3 \quad P4}{1 + 0 + 5 + 2}{4}$

$= 2$ msec.

Ans(3:2) total 100 processes in ready queue.
let time quantum be x.

$\therefore (100-1)(x + 5\text{ msec}) \leq 1$ sec.

$99(x + 5 \times 10^{-6}) \leq 1$

$x + 5 \times 10^{-6} \leq \frac{1}{99}$

$x \leq \frac{1}{99} - 5 \times 10^{-6}$

$x \leq \frac{10^6}{99} - 5$ micro sec.

$x \leq 10101.00 - 5$

$x \leq 10096$ microsec

$x \leq 10.096$ msec.

$\therefore$ $x = 10$ msec (time quantum)

millisecond

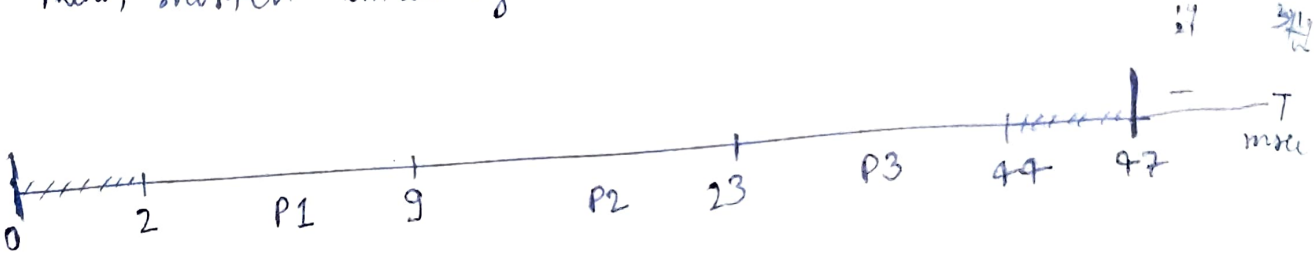
```
  099
10101
    5
  96
```

Ans 3:1

		arrival time	burst time	20 %.	70 %.	10 %.
3 process ,	P1	0	10	2	7	1
	P2	0	20	4	14	2
	P3	0	30	6	21	3

20 %. I/O → 70 %. CPU → 10 %. I/O

Now, shortest remaining time first scheduling :-



∴ total time = 47 msec

CPU remain idle for 2 + 3

= 5 msec

∴ % of time CPU remain idle

$$= \frac{5}{47} \times 100$$

$$= 10.638\ \%.$$

~~Ans 1:-~~

Ans 2 :-

④ (i) Running to Ready :-

When process is running, the time interrupt occur and process move to ready.

(ii) Waiting to ready :-

Waiting occur when process need I/O or other event. when this event completed then Kernel moves to ready state.

Ans 1:-

(2) True : when user program call syscall then user change to supervisor mode which affect the system settings.

(4) False : preemptive CPU scheduling can result in shorter average waiting time compared to non preemptive because in non preemptive no interrupt occur & whole process complete while in preemptive interrupt occur & then min. remaining process scheduler next.

(7) ~~False~~ True : yes because address of the child process created by fork() is same as parent process.

(8) False : pid will return many value either >0, <0 or 0 then in child case fork() call 3 times while in parent case fork() call 2 times so 6 4 times is not printed.

(6) False : because different threads perform different functions and therefore no data shared between 2 execution threads of a process.

Ans 2:- (1) ~~Kernel is necessary to~~ It is necessary because kernel provide basic functionalities like access to I/O and other system functions

Kernel provide essential functionalities of OS ie managing memory, network, file, processes, system call.

(2) unnamed pipe() can be used to communicate btn only child & parent processes.
ex: child writes in writing end & parent reads that work in the reading end.