Date :

UIT2503 - PRINCIPLES OF OPERATING SYSTEMS.

ASSIGNMENT - 2

BY: R. NITHYASRI
REG NO: 3122 22 5002 086
CLASS: IT - 'B'

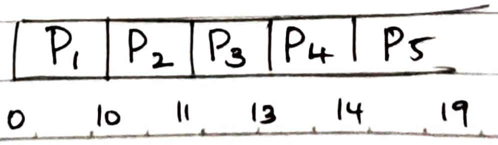
1)

Process	Burst Time	Priority	Arrival Time
P1	10	5	0.0
P2	1	11	0.2
P3	2	9	0.4
P4	1	3	0.5
P5	5	7	0.6

FCFS

Process	Burst Time	A.T	C.T	TAT	WT
P1	10	0.0	10	10	0
P2	1	0.2	11	10.8	9.8
P3	2	0.4	13	12.6	10.6
P4	1	0.5	14	13.5	12.5
P5	5	0.6	19	18.4	13.4
				Avg: 13.06	Avg: 9.26

Gantt chart :-



Date : ____________

SJF :-

Gantt chart

P1	P2	P4	P3	P5

0 10 11 12 14 19

Process	A.T	B.T	C.T	TAT	W.T
P_1	0.0	10	10	10	0
P_2	0.2	1	11	10.8	9.8
P_3	0.4	2	14	13.6	11.6
P_4	0.5	1	12	11.5	10.5
P_5	0.6	5	19	18.4	13.4
			Avg :	12.86	9.06

Non-pre-emptive priority :-

P_1 : 5	P_2 : 11	P_3 : 9	P_4 : 3	P_5 : 7

Gantt chart :-

P1	P4	P5	P3	P2

0 10 11 16 18 19

Process	A.T	B.T	C.T	TAT	W.T
P_1	0.0	10	10	10	0
P_2	0.2	1	19	18.8	17.8
P_3	0.4	2	18	17.6	15.6
P_4	0.5	1	11	10.5	9.5
P_5	0.6	5	16	15.4	10.4
			Avg :	14.46	10.66

Date :

Round Robin quantum = 1

Gantt chart

P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P1	P1	P1	P1

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Process	A.T	B.T	C.T	TAT	W.T
P_1	0.0	10	19	19	9
P_2	0.2	1	2	1.8	0.8
P_3	0.4	2	7	6.6	4.6
P_4	0.5	1	4	3.5	2.5
P_5	0.6	5	14	13.4	8.4
			Avg :	8.86	5.06

d) Minimal average W.T is for Round Robin :

e) SJF - preemptive

Gantt chart

P1	P2	P4	P3	P5	P1

0 0.2 1.2 2.2 4.2 9.2 19

Process	A.T	B.T	C.T	TAT	W.T
P_1	0.0	10	19	19	9
P_2	0.2	1	1.2	1	0
P_3	0.4	2	11.2	3.8	1.8
P_4	0.5	1	2.2	1.7	0.7
P_5	0.6	5	9.2	8.6	3.6
			Avg :	6.82	3.02

2)

```
monitor AlarmClock

{ int  currentTick = 0;
  queue. waitingTasks;

  procedure delay (int ticks)

  {
    int wakeupTime = currentTick + ticks;
    enqueue ( waitingTasks, (WakeUpTime,
                             CurrentTask));

    while ( currentTick < WakeUpTime)

    {  wait () ;

     }
   }

 procedure tick()
  {
    currentTick = currentTick + 1;

    while (! isEmpty (waitingTasks) &&
      peek ( waitingTasks). wakeUpTime
             <=  currentTick)

     {
        task = dequeue (waitingTasks);
        notify( task);
      }
   }
{
 current Tick = 0;
 waiting Tasks = empty Queue();
 }
}
```

3) BANKER'S ALGORITHM :-

	Allocation	Max	Available	Need
P_0	0 0 1 2	0012	1520	0000
P_1	1 0 0 0	1750		0750
P_2	1 3 5 4	2356		1002
P_3	0 6 3 2	0632		0000
P_4	0 0 1 4	0656		0642

Work = Available.

P_0 : Need ≤ Work. then work = work + allocation

0000 < 1520 (P_0 completed)

work = 1520
0012
1532

P_1 : 0750 > 1532

P_2 : 1002 < 1532 (Can complete)

1532
1354 / 2886 /

P3 : 0000 < 2 8 8 6 (can complete)

$$\begin{array}{lcccc} \text{work} = & 2 & 8 & 8 & 6 \\ & 0 & 6 & 3 & 2 \\ \hline & 2 & 14 & 11 & 8 \end{array}$$

P4 : 0 6 4 2 < 2 14 11 8 (can complete)

$$\begin{array}{lcccc} \text{work} = & 2 & 14 & 11 & 8 \\ & 0 & 0 & 1 & 4 \\ \hline & 2 & 14 & 12 & 12 \end{array}$$

P1 : 0 7 5 0 < 2 14 12 12 (can complete)

$$\begin{array}{lcccc} \text{work} = & 2 & 14 & 12 & 12 \\ & 0 & 7 & 5 & 0 \\ \hline & 2 & 21 & 17 & 12 \end{array}$$

∴ The system is in safe state

$P_0 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$

b) req from P_1 for (0, 4, 2, 0)

if req < need :

0 4 2 0 < 0 7 5 0

Date :

if req < available.

0 4 20 < 15 20

then for P_i Allocation :

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 4 & 2 & 0 \\ \hline 1 & 4 & 2 & 0 \\ \hline \end{array}$$

Available :

$$\begin{array}{cccc} 1 & 5 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ \hline 1 & 1 & 0 & 0 \\ \hline \end{array}$$

Need :

$$\begin{array}{cccc} 0 & 7 & 5 & 0 \\ 0 & 4 & 2 & 0 \\ \hline 0 & 3 & 3 & 0 \\ \hline \end{array}$$

req = Available : req = 1100

P_0 : 0000 < 1100, then req = $\begin{array}{c} 1100 \\ 0012 \end{array}$ = 1112

P_1 : 0330 > 1112

P_2 : 1002 < 1112, then req $\begin{array}{c} 1112 \; \oplus \\ 1354 \end{array}$ = 2466

P_3 : 0000 < 2466, then req = $\begin{array}{c} 2466 \\ \oplus\, 0632 \end{array}$ = 2698

P_4 : 0 0 4 2 < 2 10 9 8 , then req,

$$\begin{array}{rrrr} 2 & 10 & 9 & 8 \\ 0 & 0 & 1 & 4 \end{array} \oplus$$

2 10 10 12

P_1 : 0 3 3 3 < 2 10 10 12 , then req

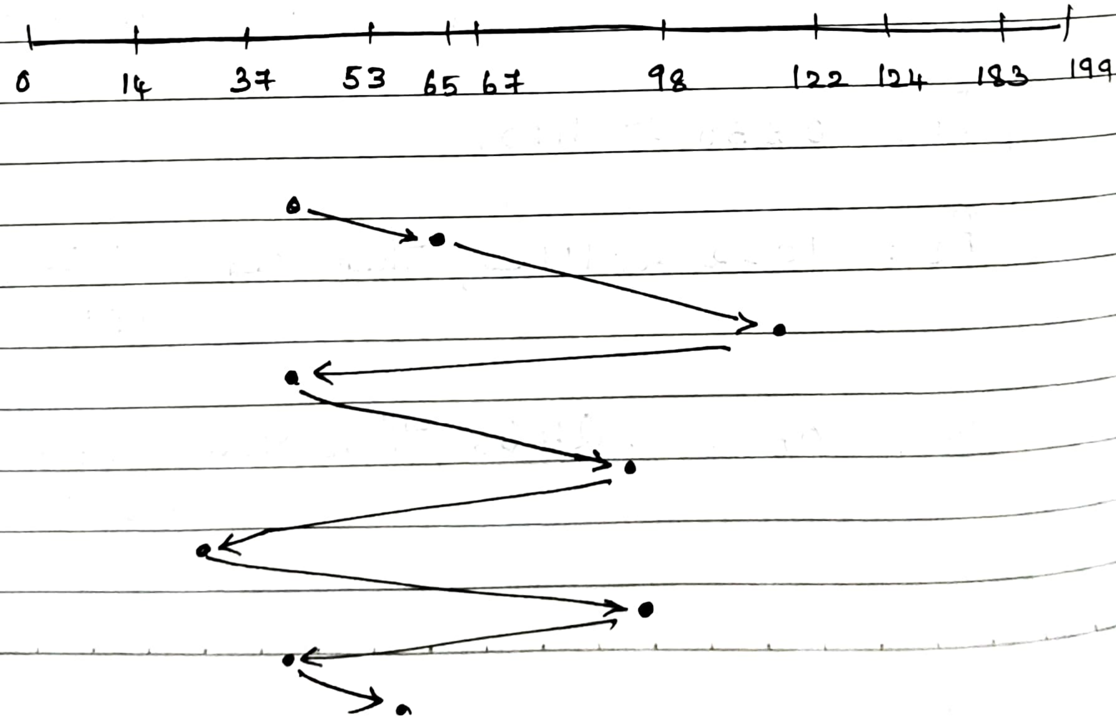
$$\oplus \begin{array}{rrrr} 2 & 10 & 10 & 12 \\ 1 & 4 & 2 & 0 \end{array} = 3\ 14\ 12\ 12$$

∴ The request can be granted immediately and system remains safe.

11) Disk drive has 200 cylinders, numbered from 0 to 199.

Head pointer : 53
Request queue: 98, 183, 37, 122, 14, 124, 65, 67

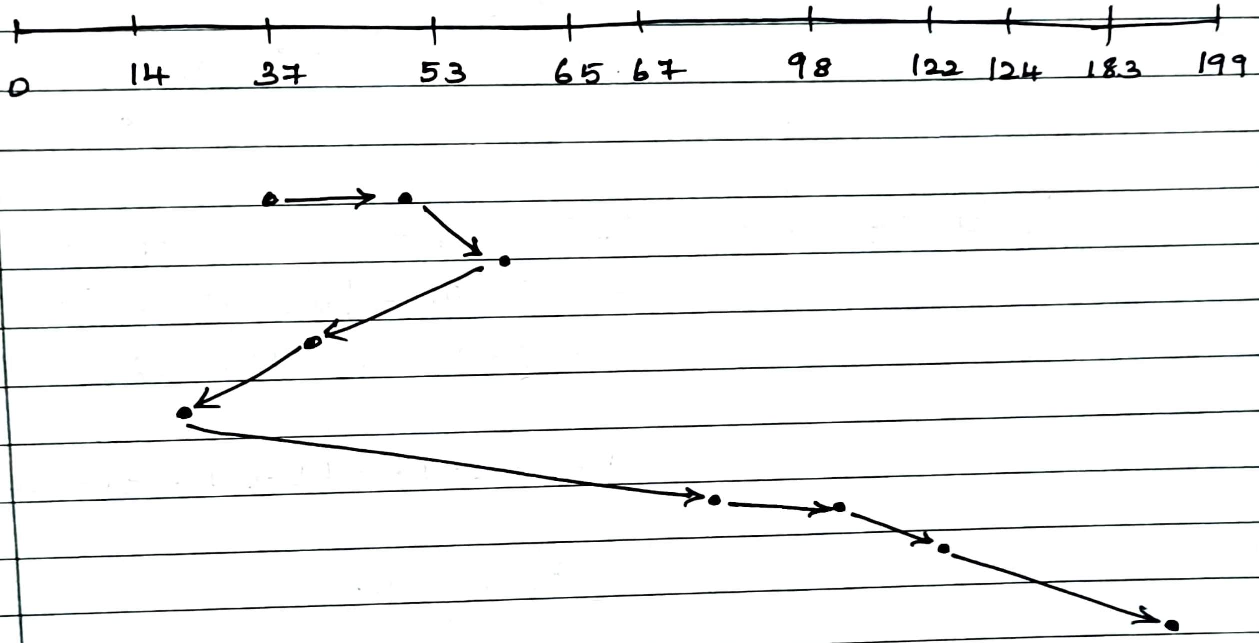
(i) FCFS



Total distance = |98 - 53| + | 183 - 98| + |37 - 183|
+ |122 - 37| + |14 - 122| + |124 - 14|
+ | 65 - 124| + |67 - 65|

= 45 + 85 + 146 + 85 + 108 + 110 + 59 + 2
= 640 cylinders.

(ii) SSTF (Shortest Seek Time First)

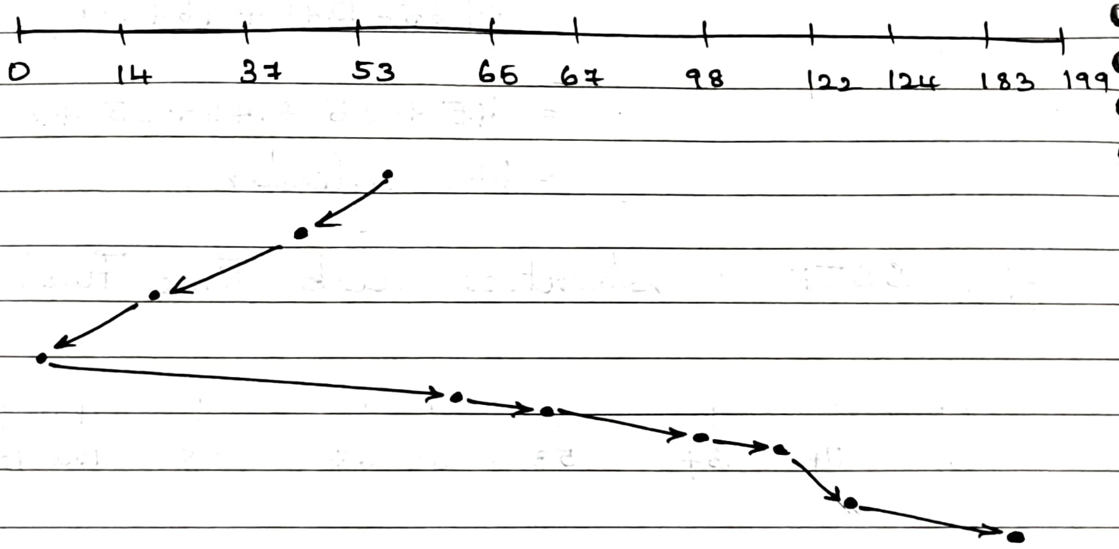


total distance = |65 - 63| + |67 - 65| + |37 - 67| +
|14 - 37| + |98 - 14| + |122 - 98| +
|124 - 122| + |183 - 124|

= 12 + 2 + 30 + 23 + 84 + 24 + 2 + 59

= 236 cylinders.

(iii) SCAN

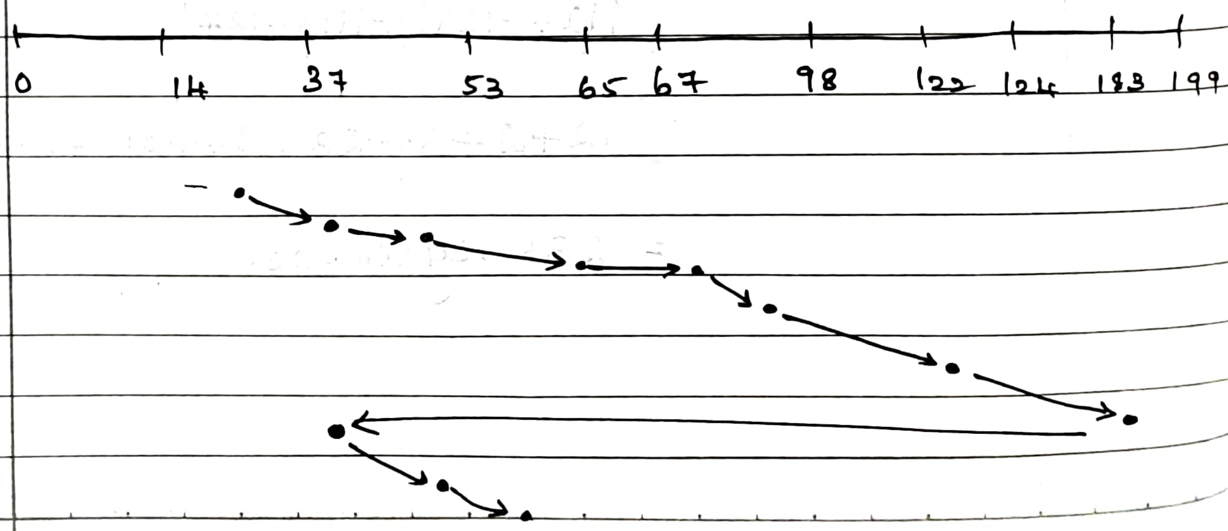


total cylinders :- |37 - 53| + |14 - 37| + |0 - 14|
+ |65 - 0| + |67 - 65| + |98 - 67| +
|122 - 98| + |124 - 122| + |183 - 124|

= 16 + 23 + 14 + 65 + 2 + 31 + 24 + 2 + 59

= 236 cylinders.

(iv) C - SCAN



Date :

$$\text{total distance} = |65 - 53| + |67 - 65| + |98 - 67|$$

$$+ |122 - 98| + |124 - 122| +$$

$$|183 - 124| + |199 - 183| + |0 - 99|$$

$$+ |14 - 0| + |37 - 14|$$

$$= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 199$$

$$+ 14 + 23$$

$$= 382 \text{ cylinders}$$