2C303 41

D D M M Y Y Y Y

Module 2

2) using FCFS

Process	A.T	B.T	completion time	T.A.T	W.T
P_1	0	9	9	9	0
P_2	1	4	13	12	8
P_3	2	9	22	20	11
P_4	3	5	27	24	19

Gantt chart

P_1	P_2	P_3	P_4
0	9	13	22 27

average T.A.T = $\frac{9+12+20+24}{4} = \frac{65}{4} = 16.25$ ms

average W.T = $\frac{0+8+11+19}{4} = \frac{38}{4} = 9.5$ ms

priority

Process	A.T	B.T	priority	Comp time	T.A.T	W
P_1	0	9 80	3	22	22	13
P_2	1	4 80	2	14	13	9
P_3	2	9 80	1	11	9	0
P_4	3	5 0	4	27	24	19

Gantt chart

P_1	P_2	P_3	P_3	P_2	P_1	P_4
0	1	2	3	11	14	22 27

Running queue : P_1 P_2
P_1 P_2 P_3
P_1 P_2 P_3 P_4

D D M M Y Y Y Y

Avg T.A.T = $68/4$ = 17 MS

Avg W.T = $41/4$ = 10.25 MS

Round Robin

q = 2ms

Process	A.T.	B.T.
P_1	0	9 ~~7 5 3~~
P_2	1	~~4~~ 2
P_3	2	9 ~~7 5~~
P_4	3	~~5~~ 3

Gantt chart

P_1	P_2	P_3	P_1	P_4	P_3	P_1	P_4
0	2	4	6	8	10	12	14

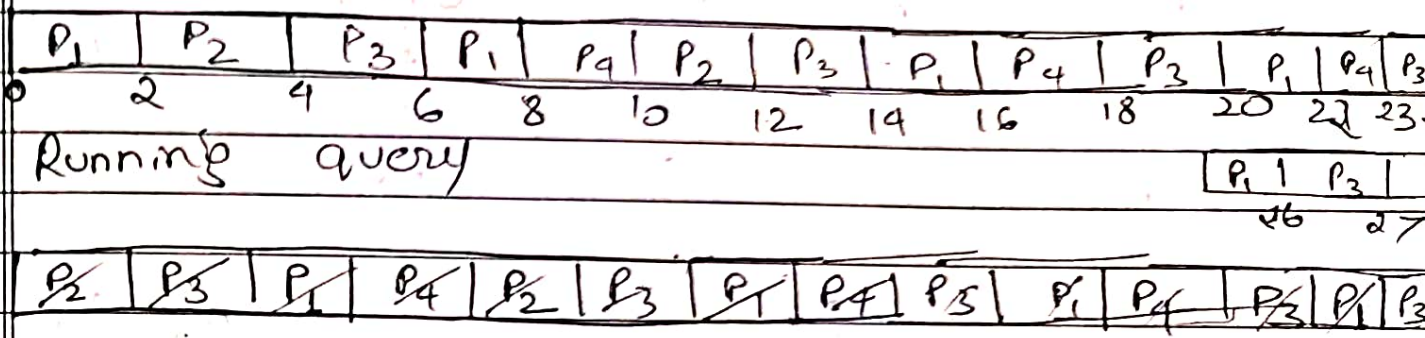
Running query

P_2	P_3	P_1	P_4	P_3	P_1	P_4	P_3	P_1

Process	A.T	B.T
P_1	0	~~9 7 5 3 1 0~~
P_2	1	~~4 2~~ 0
P_3	2	~~9 7 5 3 1~~ 0
P_4	3	~~5 3~~ 1 0

D D M M Y Y Y Y

Gantt chart



Running queue

	comp time	T.A.T	W.T
P_1	26	26	17
P_2	12	11	7
P_3	27	25	16
P_4	23	20	15

Avg. T.AT = $82/4$ = 20.5 ms

Avg. W.T = $55/4$ = 13.75 ms

SJF

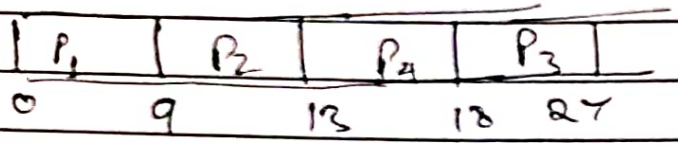
DD MM YYYY

SJF

Process	A.T.	B.T.	Completion time	T.A.T	W.T.
P_1	0	9	9	9	0
P_2	1	4	13	12	8
P_3	2	9	18	16	7
P_4	3	5	27	24	19

Avg = 34/4 = 8.5 ms

Avg = 34/4 = 8.5 ms



b) Threads are unit of CPU utilization They facilitate multitasking using single processor. Thus making system more:-

- Responsive
- Scalable
- Efficient & Fast
- Easier maintability

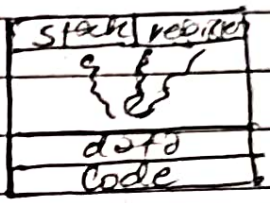
Also interruption in a random thread anywhere doesn't hinder the flow of program & other programs keeps on execution without getting disturbed. There are two types of thread: user & kernel

D D M M Y Y Y Y

Ex:- Java Multi-threading
win 32 thread

It achieves this by distributing separate stack & storage whereas keeping the same code and data

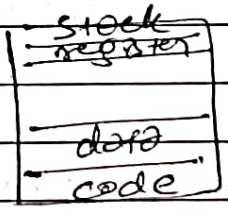
Ex:



thread.

Supposedly we have a word document open, so each section like: menu bar, Format bar etc. will have its own thread.

Process is traditional concept of resource distribution where single storage is allocated to each program. So for execution of each & every program we need processor. Thus it is slower & less scalable compare to thread

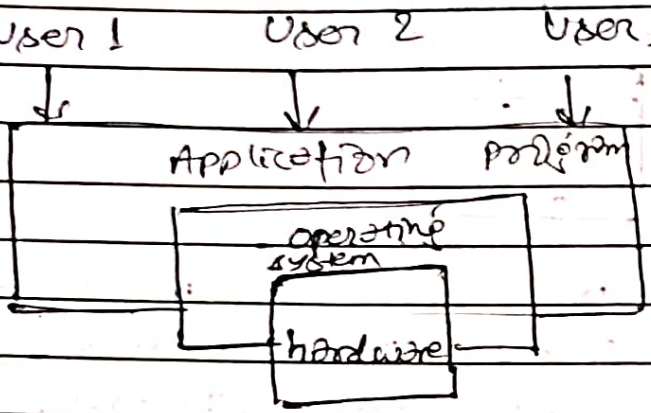


process.

DD MM YYYY

Q.

a. Operating system is system program that provides interface to user to interact with hardware.



Thus it acts a middle interface that help access the hardware other services of OS include:

- Resource Allocation & distribution
  by ensuring that resources available (processor, cache, registered) are properly distributed according to necessity and demand of program
- Facilitate in accessing system:
  It makes user-friendly approach to help access the system
- Prevention & Security:

D D M M Y Y Y Y

It secures the system by its ~~feature~~ events such as privileged instruction & other networking mishaps.

User view of OS:-

» depending on performance requirement & ease of use: handheld system, standalone system

» depending on resource allocation: terminal, ~~workstation~~.
System view: resource allocation

Q.

1 → b) ✗

2 → c) ✗

3 → b ✓

4 → b) ✓

5 → b) ✓

New
Ready
Wait
Terr

Q.

b)

i) Multi-processor
System designed on process (instead of thread) which allocate 2 processes for each task are multi-processor. They are high-in demand of resource and cost high. Distinctive points:-
» They share common memory.
» Communication is faster
» limited scalability

D D M M Y Y Y Y

Clustered - System.

System : like peer-to-peer or web-based which are interconnected to form network

→ They are independent
→ Higher latency
→ Higher scalability : More system can be added.

(ii)

Multi-programming	Multi-tasking
→ focus on efficient CPU utilization	user-centric
• aims to maximize processor resource	focus on responsiveness & interactivity
• involve rapid & dynamic searching of program	switching between program after task completion