

Assignment - I

- Q. In what ways is the modular kernel approach similar to layered approach? In what ways does it differ from layered approach?

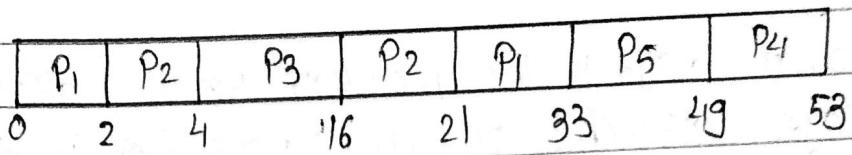
Ans The modular kernel approach realizes subsystems to interact with each other through carefully constructed interfaces that are typically narrow [in terms of functionality] that is exposed to external modules. The layered kernel approach is similar in that respect. However, the layered kernel imposes a strict ordering of subsystems such that subsystem at the lower layers are not allowed to invoke operations which correspond to the upper-layer subsystem. There are no such restriction in the modular kernel approach, wherein modules are free to invoke each other without any constraints.

- Q. 2 Consider the processes whose arrival times and burst time are given. Draw Gantt chart and find the average waiting time and average turn around time using Priority Preemptive and Round Robin (time quantum = 3)

Name: Harshhai Solanki
 SE-4th
 Roll.no: 62
 Sub: OS

Page No.	
Date	

ANS (i) Preemptive Priority:

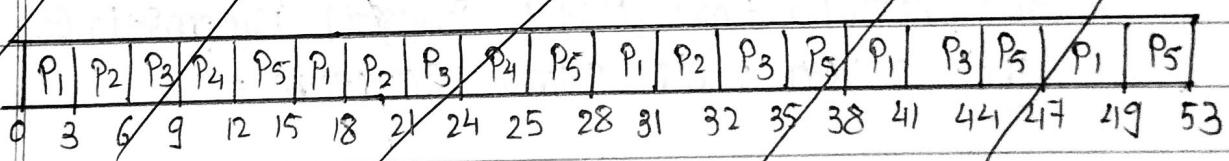


Process ID	Arrival Time	Burst Time	Priority	TAT	WT
P ₁	0	14	3	[33-0] = 33	19
P ₂	2	7	2	[21-2] = 19	12
P ₃	4	12	1	[16-4] = 12	0
P ₄	6	4	5	[53-6] = 47	43
P ₅	8	16	4	[49-8] = 41	25
Average				30.4	19.8

Average Turn around Time (ATAT) = 30.4.

Average waiting Time (AWT) = 19.8

(ii) Round Robin



(iii) Round Robin scheduling algorithm:

Time Quantum = 3 units

Process ID	Arrival time	Burst time
P ₁	0	14 14 8 8 2 0
P ₂	2	4 4 10
P ₃	4	12 9 8 3 0
P ₄	6	4 10
P ₅	8	18 13 10 7 0

Gantt chart:

P ₁	P ₂	P ₁	P ₃	P ₄	P ₂	P ₅	P ₁	P ₃	P ₄	P ₂	P ₅	P ₁	P ₃	P ₅	R	P ₃	P ₅	
0	3	6	9	12	15	18	21	24	27	28	29	32	35	38	41	43	46	53

Process ID	Arrival Time	Burst Time	Completion Time	TAT	WT
P ₁	0	14	23	43	29
P ₂	2	7	29	27	20
P ₃	4	12	46	42	30
P ₄	6	4	28	22	18
P ₅	8	16	53	45	29

TAT = completion time - arrival time

WT = Turn around time - Burst time.

where, TAT = Turn Around Time

WT = waiting time.

$$\therefore \text{Average turn around time} = \frac{43 + 27 + 42 + 22 + 45}{5}$$

$$\therefore \boxed{\text{Avg TAT} = 35.8}$$

$$\therefore \text{Average waiting time} = \frac{29 + 20 + 30 + 18 + 29}{5}$$

$$\boxed{\text{Avg WT} = 25.2}$$

Q3

Consider 3 concurrent processes P_1 , P_2 and P_3 as shown below which access shared variable CNT that has been initialized to 75.

Process P_1	Process P_2	Process P_3
..
..
$CNT = CNT + 15$	$CNT = CNT - 20$	$CNT = CNT + 20$
..
..

The processes are executed on uniprocessor system running a time shared operating system. Find min and max value of CNT.

Ans minimum value of CNT :

1. P₂ reads CNT = 75 , PDE-emptied

2. P₁ executes CNT = CNT + 15 = 90

3. P₃ executes CNT = CNT + 20 = 90 + 20 = 110

4. P₂ has CNT = 75 executes

$$CNT = CNT - 40 = 75 - 40 = 35$$

∴ minimum value of CNT is 35.

maximum value of CNT :

1. P₃ reads CNT = 75 , PDE-emptied

2. P₂ reads CNT = 75 , executes CNT = CNT - 40
CNT = 35

3. Now P₃ executes CNT = CNT + 20 = 95

4. P₁ reads CNT = 95 executes CNT = CNT + 15 = 110

P₁ writes 110 final value.

∴ maximum value of CNT is 110

∴ minimum value of CNT = 35

maximum value of CNT = 110

Q4

	Processes Allocation				MAX				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P ₀	0	0	1	2	0	0	1	2	1	5	2	0
P ₁	1	0	0	0	1	7	5	0				
P ₂	1	3	5	4	2	3	5	6				
P ₃	0	6	3	2	0	6	5	2				
P ₄	0	0	1	4	0	6	5	6				

(i) What is the content of NEED matrix?

⇒ Need matrix is calculated by subtracting allocation matrix from max matrix

Processes	Need (MAX - Allocation)			
	A	B	C	D
P ₀	0	0	0	0
P ₁	0	7	5	0
P ₂	1	0	0	2
P ₃	0	0	2	0
P ₄	0	6	4	2

(ii) Is the system in safe state?

Ans (i) The available matrix [1 5 2 0]

(ii) A process after it has finished execution is supposed to free up all the resources it hold.

(iii) we need to find a safety sequence such that it satisfied the criteria need
 $\text{need} \leq \text{Available}$

(iv) since $\text{need}(P_0) \leq \text{Available}$, we select.

$$P_0 [\text{Available}] = [\text{Available}] + [\text{Allocation}(P_0)]$$

$$\text{Available} = [1 5 2 0] + [0 0 1 2]$$

$$= [1 5 3 2]$$

- Need(P2) \leq Available

$$\text{Available} = [1532] + [1354] = [2886]$$

- Need(P3) \leq Available

$$\text{Available} = [2886] + [0632] = [214118]$$

- Need(P4) \leq Available

$$\text{Available} = [214118] + [0014] = [2141212]$$

- Need(P1) \leq Available

$$\text{Available} = [2141212] + [1000] = [3141212]$$

- Safe sequence is $\langle P_0, P_2, P_3, P_1 \rangle$

(iii) If the request from process P1 arrives for (0, 4, 2, 0) can request be granted immediately.

- Ans • First we check if $\text{req}(P1)$ is less than $\text{need}(P1)$

$$[0420] < [0750] \text{ is true}$$

- Now we check $\text{Req}(P1)$ is less than Available.

$$[0420] < [1520] \text{ is true}$$

- So we update value as:

$$\begin{aligned}\text{Available} &= \text{Available} - \text{Request} = [1520] - [0420] \\ &= [1100]\end{aligned}$$

$$\begin{aligned}\text{Allocation} &= \text{Allocation}(P1) + \text{Request} = [1000] + [0420] \\ &= [1420]\end{aligned}$$

$$\begin{aligned}\text{Need} &= \text{Need}(P1) - \text{Request} = [0750] - [0420] \\ &= [0330]\end{aligned}$$

Name: Harshhai Solanki

SE-4

ROLL NO: 62

SUB: OS

Page No.

Date

Processes	Allocation				MAX				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P ₀	0	0	1	2	0	0	1	2	0	0	0	0	1	1	0	0
P ₁	1	4	2	0	1	7	5	0	0	3	3	0				
P ₂	1	3	5	4	2	3	5	6	1	0	0	2				
P ₃	0	6	3	2	0	6	5	2	0	0	2	0				
P ₄	0	0	1	4	0	6	5	6	0	6	4	2				

- This is the modified table.
- On verifying we see that the safe sequence is still remain same. The system continue to remain in a safe state.