Name: Harshbhai Solanki
SE-4
Roll. No: 62
Sub: OS

Assignment- I

1) In what ways is the modular kernel approch similar to layer approch? In what ways does it differ from layerred approch?

Ans The modular kernel approch requires subsystem to interact with each other through carefully construct interfaces that are typically narrow [in terms of functionality that is expossed to external modules.] The layered kernel approch 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 corresponding to the upper-layer subsystem. There are no such restriction in the modular-kernel approch, wherein modules are free to invoke each other without any constraints.

Q.2 consider the Processes whose arrival times and bust 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: Harshbhai solanki
SE-4-D
Roll.no: 62
SUB: OS

Page No. | Date

ANS (i) Preemptive Priority:

P1	P2	P3	P2	P1	P5	P4

0 2 4 16 21 33 49 53

Process ID	Arrival time	Burst Time	Priority	TAT	WT
P1	0	14	3	[33-0]=33	19
P2	2	7	2	[21-2]=19	12
P3	4	12	1	[16-4]=12	0
P4	6	4	5	[53-6]=47	43
P5	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

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

0 3 6 9 12 15 18 21 24 25 28 31 32 35 38 41 44 47 49 53

Name: Harshbhai Solanki
Roll No: 62
SE-4-D
SUB: OS

(ii) **Round Robin scheduling algorithm:**

Time Quantum = 3 units

Process Id	Arrival time	Burst time
P_1	0	~~14~~ ~~11~~ ~~8~~ ~~5~~ ~~2~~ 0
P_2	2	~~7~~ ~~4~~ ~~1~~ 0
P_3	4	~~12~~ ~~9~~ ~~6~~ ~~3~~ 0
P_4	6	~~4~~ ~~1~~ 0
P_5	8	~~16~~ ~~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	P_1	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	43	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

Name: solanki Harshbhai
SE-4-D
Roll. No: 62
Sub: OS

Page No.	
Date	

TAT = completion time - arrival time
WT = Turn around time - Burst time

where, TAT = Turn around Time
WT = waiting Time

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

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

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

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

Q3 consider 3 concurrent process 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 - 40	cnt = cnt + 20
:	:	:
:	:	:

name: Harshbhai solanki
SE-4-D
Roll.No: 62
SUB: OS

Page No.
Date

The Process are executed on uniprocessor system running a time shared operating system. find min and max value of cnt.

ANS minimum value of cnt :

1. P_2 reads cnt = 75 , Pre-empted
2. P_1 executes cnt = cnt + 15 = 90
3. P_3 executes cnt = cnt + 20 = 90 + 20 = 110
4. P_2 has cnt = 75 executes
   cnt = cnt - 40 = 75 - 40 = 35
   ∴ minimum of cnt is 35.

maximum value of cnt :

1. P_3 reads cnt = 75 , Pre-empted.
2. P_2 reads cnt = 75 , executes cnt = cnt - 40
   cnt = 35
3. now P_3 executes cnt = cnt + 20 = 95
4. P_1 reads cnt = 95 executes cnt = cnt + 15 = 110
   P_1 writes 110 final value.
   ∴ maximum value of cnt is 110

∴ minimum value of cnt = 35
maximum value of cnt = 110

name: Harshbhai solanki
SE-4-D
Roll no: 62
SUB: OS

Page No.
Date

Q-4

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 max is [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

Need ≤ Available

(iv) since Need (P_0) ≤ Available, we select.

P_0 [Available] = [Available] + [Allocation (P_0)].

Available = [1 5 2 0] + [0 0 1 2]

= [1 5 3 2]

Name: Harshbai Solanki
SE-4
Roll-no: 62
Sub: OS

Page No.
Date

- Need (P2) ≤ Available
  Available = [1 5 3 2] + [1 3 5 4] = [2 8 8 6]
- Need (P3) ≤ Available
  Available = [2 8 8 6] + [0 6 3 2] = [2 14 11 8]
- Need (P4) ≤ Available
  Available = [2 14 11 8] + [0 0 1 4] = [2 14 12 12]
- Need (P1) ≤ Available
  Available = [2 14 12 12] + [1 0 0 0] = [3 14 12 12]
- Safe sequence is < P0, P2, P3, P4, P1 >.

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

Ans
- First we check if req (P1) is less than Need (P1)
  [0 4 2 0] < [0 7 5 0] is true
- Now we check Req (P1) is less than Available.
  [0 4 2 0] < [1 5 2 0] is true
- So we update value as:
  Available = Available - Request = [1 5 2 0] - [0 4 2 0]
  = [1 1 0 0]
  Allocation = allocation (P1) + Request = [1 0 0 0] + [0 4 2 0]
  = [1 4 2 0]
  Need = Need (P1) - Request = [0 7 5 0] - [0 4 2 0]
  = [0 3 3 0]

Name: Harshbhai 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
P0	0	0	1	2	0	0	1	2	0	0	0	0	1	1	0	0
P1	1	4	2	0	1	7	5	0	0	3	3	0				
P2	1	3	5	4	2	3	5	6	1	0	0	2				
P3	0	6	3	2	0	6	5	2	0	0	2	0				
P4	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.