DECCAN EDUCATION SOCIETY'S

KIRTI M. DOONGURSEE COLLEGE (AUTONOMOUS)

DADAR WEST, MUMBAI-400028.

ASSIGNMENT

NAME:- Ajay Kumar Gupta.

ROLL NO.:- 33 **CLASS:-** fyBsc.cs **DIV:-** _____

SUBJECT Operating System

PAPER NO. _____ **SEM.** _____ **Year:-** _____

Date:- _____	Mark Obtained
	___ / 20
	Name of Examiner _____
Student's Signature Ajay.	Signature _____

Q.1. Outline requirements should be satisfied to resolve the Critical Section Problem.

→ Mutual Exclusion: If process P_i is executing in its Critical Section, then no other processes can be executing in their critical sections.
Only one process can execute its critical sectionat a time

Progress. If no process is executing in its critical section and there exist some processes that wish to enter their Critical section, then the selection of the process that will enter the critical section next cannot be postponed indefinitely.

Bounded Waiting: A bound must exist on the number of times that other processes are allowed to enter their Critical sections after a processe has made a request to enter its critical section and befor that request is granted.

* Assume that each process executes at a nozero speed
* No assumption Concerning relative speed of the n processes.

Q.2 Round-Robin Scheduling behave differently depending on the its time quantum. Can the time quantum be set to make round robin behave the same as any of the following algorithm? If so how?
i) first-come-first served ii) shortest Job first

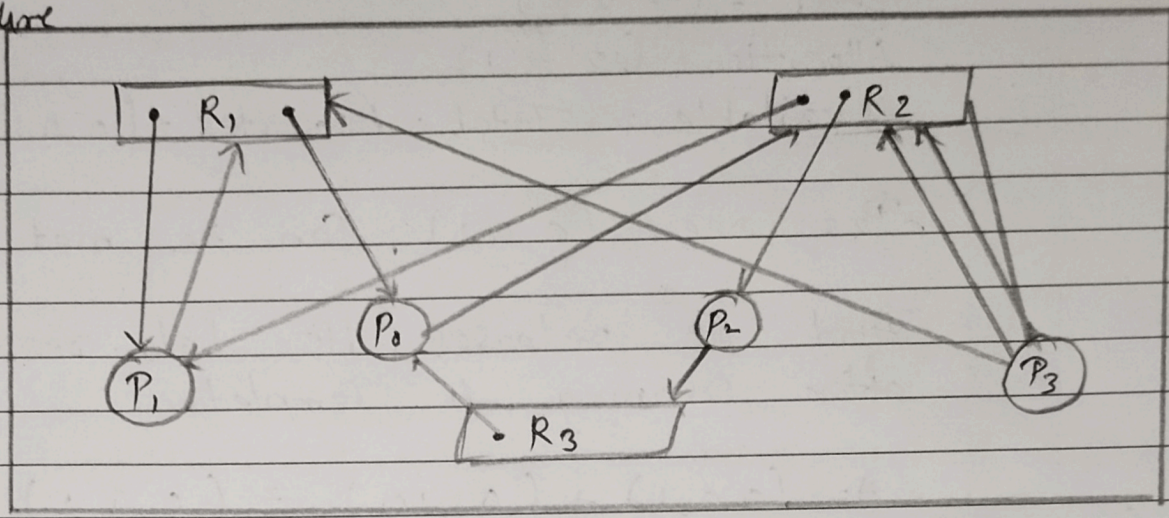
Ans The period of Time for which a process is allowed to run in a premptive multitasking System is generally called the time slice or quantum.

1) first come first serve (fcfs): Fcfs Scheduling behaves Similar to Round-Robin because each process to Round-Robin because each process run to Completion befor the next one Start.

2) Shortest Job first :- Conversely, Setting the time quantum to a very small value makes Round Robin before behave more like a preemptive Shortest Job first (SJF) Scheduling algorithm where the process with the shortest burst time is selected to run next.

Round Robin Scheduling Can behave the same as first-served (fcfs) Scheduling by Setting the time quantum to be equal to the time required for the longest Job. However, it Cannot behave the Same as Shortest Job first (SJF) Scheduling each process is given a fixed time quantum where in SJF Scheduling the Process with the Shortest burst time is always chosen first.

Q.8] Consider the resource allocation graph in the figure



1] find if the system is in deadlock state.
2] If not find a safe sequence.

— To determin id the system is in a deadlock state, let's dirst calculate the available resources:

Here's the Allocation & future Need matrix

Allocation

	r_1	r_2	r_3
P_0	1	0	1
P_1	1	1	0
P_2	0	1	0
P_3	0	1	0

future Need

	r_1	r_2	r_3
P_0	0	1	1
P_1	1	0	0
P_2	0	0	1
P_3	1	2	0

Total - (2 3 2)
Allocation = (2 3 1)
Available = Total - Allocated = (0 0 1)

P_2s need (0 0 1) Can be met

And it releases its held resources after running to completion.

A = (0 0 1) + (0 1 0) = (0 1 1)
P_0's need (0 1 1) can be met and it releases

A - (0 1 1) + (1 0 1) = (1 1 2)

P_1's need can be met (1 0 0) and it releases

A = (1 1 2) + (1 0 0) = (2 2 2)

P_3's need can be met so, the safe sequence will be.
$P_2 - P_0 - P_1 - P_3$

Q.4] Consider the following set of processes with the length of the CPU burst given in milliseconds.

Process	Burst time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

The processes are assumed to have arrived in the P_1 P_2 P_3 P_4 P_5 all at time 0.

1) Draw four Gantt charts that illustrate the executing of these processes. using the following Scheduling algorithm : FCFS and Priority

2) Compute the average waiting time and average turnaround time for the given scenario.

3) Compare the outcome of the algorithms and suggest the best Suited one for the given scenario.

Ans 1) Gantt Chart : FCFS Scheduling

P_1	P_2	P_3	P_4	P_5
0	10	11	13	14	19

Waiting time for $P_1 = 0$
Waiting time for $P_2 = 10$
Waiting time for $P_3 = 11$
Waiting time for $P_4 = 13$
Waiting time for $P_5 = 14$

Average waiting time
= Sum of waiting time for individual Process / No. of process

$= \frac{0+10+11+13+14}{5} = \frac{48}{5} = 9.6 = 9.6$

Average execution time
= Sum of Burst Time / No. of process $= \frac{10+1+2+1+5}{5}$

$= \frac{19}{5} = 3.8 = 3.8$

Average turn-around time
= Average waiting time + Average execution time

$= 9.6 + 3.8 = 13.4$

ii) Gantt Chart : Priority

P_2	P_5	P_1	P_3	P_4

0 1 6 16 18 19

Waiting time for $P_1 = 6$
Waiting time for $P_2 = 0$

Waiting time for $P_3 = 10$
Waiting time for $P_4 = 18$
Waiting time for $P_5 = 1$

$$AWT = \frac{6 + 0 + 16 + 18 + 1}{5} = \frac{41}{5} = 8.2 = 8.2$$

$$AET = \frac{10 + 1 + 2 + 1 + 5}{5} = \frac{19}{5} = 3.8$$

$$ATT = AWT + AET = 8.2 + 3.8 = 12$$

(ii) In Both Scheduling Algorithm :-

a) The Average execution time are Same
b) The Average waiting times are different
c) The Average waiting times of FCFS is 9.6 & priority is 8.2

FCFS > Priority

∴ Priority Consume less waiting time for execution than the FCFS.

d) So, priority is best suited for this Scenario.

Q.5 Consider the following snapshot of a system.

Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	0	0	0	0	0	0
P_1	2	0	0	2	0	2			
P_2	3	0	3	0	0	0			
P_3	2	1	1	1	0	0			
P_4	0	0	2	0	0	2			

Answer the following question using the banker's algorithm

a) What is the content of the matrix Need?
b) Is the system in a safe state?
c) If a request from process P_1 arrive for (0,0,1) Can the request be granted immediately?

→ Need Matrix :-

Need Matrix = Max - Allocation

	A	B	C
P_0	0	0	0
P_1	0	0	2
P_2	0	0	0
P_3	1	0	0
P_4	0	0	0

The Need matrix is negative, it indicates that the process has already been allocated

Safe State or not!-

Available ≤ Need

P_0 :- 0 0 0 ≤ 0 -1 0

This is false

∴ Process do not execute

P_1 :- 0 0 0 ≤ 0 0 2

This is false

P_1 execute.

New Allocation - Available + Allocation

= 000 + 2 00

= 200

P_2 :- 2 0 0 ≤ -3 0 -3

This is false

∴ Process do not execute.

P_3 :- 2 0 0 ≤ -1 -1 -1

This is false

∴ Process do not execute.

P_4 :- 2 0 0 ≤ 0 0 0

This is false

∴ Process do not execute.

Starting Available = 0 0 0 No process Can Complete because all processes have atleast one resourse they still need Therefore the system is not in a safe state.

c) i] request ≤ need

P_1 :- 0 0 1 ≤ 0 0 2

This Condition is True

∴ The process P_1 execute

ii) Since 00 process P_1 Can Complete because all processes have atleast one resources they still need, the System is not in a safe State after granting the request from process P_1 for (0 0 1)

1) a] Available = available − Request

= 0 0 0 − 0 0 1 = 0 0 − 1

b] Allocation $[P_1]$ − Allocation $[P_1]$ + Request

= 2 0 0 + 0 0 1 = 2 0 1

c] Need $[P_1]$ = Need $[P_1]$ − Request

= 0 0 2 − 0 0 1 = 0 0 1

iii) But there is not in a Safe State Therefore the request Cannot be granted immediately