office copy.

Roll Number: _____

Thapar Institute of Engineering & Technology, Patiala
(Deemed to be University)
Department of Electrical and Instrumentation Engineering
Mid Semester Test (MST)

BE- 2nd Year (EEC) | **Course Code: UCS303**
Date: 11/03/2023 | **Subject: Operating System**
Time: 02:00 Hours | **Name of Faculty: Dr. Alok Kumar Shukla**
Max Marks: 25

Note: (1) Assume suitable values for missing data if any and mention in the answer, (2) Answer the Questions as per the allotted marks, (3) In case of any doubt in question try to solve as per your own understanding and write a note or remark about the assumption you have taken for solving the question.

Q1. **(2+3)**

a. The following program consists of 3 concurrent processes and 3 binary semaphores. The semaphores are initialized as S0=1, S1=0, S2=0.

Process P0	Process P1	Process P2

while (true) {
wait (S0);
print (0);
release (S1);
release (S2);
}
	wait (S1);
	Release (S0);
		wait (S2);
		release (S0);

In the above scenario, how many times will process P0 print '**0**'?

b. Define when you should use fork () over the thread. Also, discuss what happens when you call fork () in thread?

Q2. **(2.5+2.5)**

a. Does the following code suffice to protect a critical section with two processes p0 and p1 running on separate processors on a two-processor shared-memory machine, explain in detail. That is, does it meet all of the requirements for protecting a critical section? The below-mentioned code for process p0. The code for p1 is similar, with each occurrence of 0 replaced by 1. Variable 'turn' is a shared variable.

```
... /* non-critical-section code */
while(turn != 0) {}
 critical section
turn = 0;
... /* non-critical-section code */
```

b. What are the three conditions of mutual exclusion? Consider Peterson's algorithm for mutual exclusion between two concurrent processes i and j. The program executed by process is shown below:

```
repeat
    flag [i] = true;
    turn = j;
    while ( P ) do no-op;
    Enter critical section, perform actions, then exit critical
```

section
flag [i] = false;
Perform other non-critical section actions
until false;

For the program mentioned above to guarantee mutual exclusion, write the condition using *flag* and *turn* respect to processes i and j to the predicate ***P*** in the while loop.

Q3. a. What is round robin CPU scheduling algorithm? Consider *n* processes sharing the CPU in a round-robin fashion. Assuming that each process switch takes *s* seconds, what must be the quantum size *q* such that the overhead resulting from process switching is minimized but, at the same time, each process is guaranteed to get its turn at the CPU at least every t seconds? **(3+2)**

b. What is the effect of time quantum on number of context switches, explain with a neat diagram.

Q4. a. A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column ***Alloc*** denotes the number of units of each resource type allocated to each process, and the column ***Request*** denotes the number of units of each resource type requested by process in order to complete execution. **Show the order of execution of processes (P0, P1, P2) completion. Also, find which of the process will finish LAST?** **(3+2)**

	Alloc			Request		
	X	Y	Z	X	Y	Z
P0	1	2	1	1	0	3
P1	2	0	1	0	1	2
P2	2	2	1	1	2	0

b. Which necessary condition for deadlock states that a process must be holding one resource and waiting to acquire additional help by using a resource allocation graph to illustrate your answer?

Q5. Consider the following set of processes, with the length of the CPU burst given in milliseconds: **(3+1+1)**

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a. Draw the Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: First Come First Serve (FCFS), shortest job first (SJF), and non-preemptive priority (a larger priority number implies a higher priority) scheduling.

b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

c. Which of the algorithms results in the minimum average waiting time (over all processes) in part a?

Roll Number: _____

Thapar Institute of Engineering & Technology, Patiala

(Deemed to be University)

Operating System (UCS303)

Mid Semester Test (MST)

Group: MEC(MT1) ***Course Code: UCS303***

Date: 11/03/2023 ***Subject: Operating System***

MT: 02:00 Hrs, MM: 25 ***Name of Faculty: Dr. Garima Singh***

Note: Attempt all questions and assume missing data if any.

Q1	***a) Define need of dual mode of operating system with transition procedure.***	***2***
	b) Compare cache with register memory and explain cache coherence with a suitable example .	***3***
Q2	***For the processes listed in the following table, which of the following scheduling schemes will give the lowest average turnaround time.***	***5***

Process	*Arrival Time*	*Burst Time*
A	*0*	*3*
B	*1*	*6*
C	*4*	*4*
D	*6*	*2*

a) First come first serve.

b) Non-preemptive shortest job first.

c) Shortest remaining time.

d) Round robin with quantum value 2.

Q3 — ***5***

An operating system uses the Banker's algorithm for deadlock avoidance and managing the allocation of three resource types X, Y, and Z to five processes P0, P1, P2, P3 and P4. There are 10 instances of X, 7 instance of Y, 8 instances of Z. The data given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation	*MAX*
	X Y Z	*X Y Z*
P0	*1 1 2*	*5 4 4*
P1	*2 1 2*	*4 3 3*
P2	*3 0 1*	*9 1 3*
P3	*0 2 0*	*8 6 4*
P4	*1 1 2*	*2 2 3*

	a) *Calculate the need matrix?* b) *What will happen if P0 request (2, 2, 1), can the system accept this request immediately? If yes, what will be the safe sequence.* c) *What will happen if P1 request (2, 2, 1), can the system accept this request immediately? If yes, what will be the safe sequence.*	
Q4	*Consider the set of 4 processes whose arrival time and burst time are given below-* *If the CPU scheduling policy is Priority Scheduling, calculate the average waiting time. (Lower number means higher priority) and CPU utilization.*	*5*
Q5	*Explain following:* a) *Circular wait elimination rule.* b) *Deadlock detection using RAG with a suitable example.* c) *Process states.* d) *Aging.*	*5*

Process No.	*Arrival Time*	*Priority*	*Burst Time*		
			CPU Burst	*I/O Burst*	*CPU Burst*
P1	*0*	*2*	*1*	*5*	*3*
P2	*2*	*3*	*3*	*3*	*1*
P3	*3*	*1*	*2*	*3*	*1*