SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2024-25 (ODD)

SET B

Test: FJ2
Course Code & Title: 21CSC202J - Operating Systems
Year & Sem: II Year / III Sem

Date: 01.10.2024
Duration: 100 Minutes
Max. Marks: 50

Course Articulation Matrix: ***(to be placed)***

S.No	Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	CO1	3	3	2	2								3			
2	CO2	3	3	3	2								3			
3	CO3	3	3	3	2								3			
4	CO4	3	3	3	2								3			
5	CO5	3	2	3	2								3			

Questions with Answer Key

Part – A
(10 x 1 = 10 Marks)

Instructions: Answer all

Q. No	Answer with choice variable	Marks	BL	CO	PO	PI Code
1	**Program is -------- entity stored on disk and process is ---------** a) passive, active b) active, passive c) passive, passive d) active,active Answer: a) passive, active	1	L1	2	2	1.2.2
2	**A process can be ____________** a) both single threaded and multithreaded b) multithreaded c) single threaded d) stack Answer : a) both single threaded and multithreaded	1	L1	2	2	1.6.1

3	**. ---------- Scheduller reduces the degree of multiprogramming** a. Mid Term Scheduller b. CPU Scheduller c. Short Term Scheduller d. Long Term Scheduller Answer : a) Mid Term Scheduller	1	L2	2	3	1.5.1
4	**Which of the following two operations are provided by the IPC facility?** a) write & delete message b) delete & receive message c) send & delete message d) send & receive message Answer: d) send & receive message	1	L1	2	2	1.5.1
5	The ________ keeps track of pending and blocked bit vectors within each process's context. a) CPU b) Memory c) Process d) Kernel Answer: d) Kernel	1	L2	2	3	1.6.1
6	What does the term "throughput" mean in scheduling terms? a) The amount of time the CPU is idle b) The number of processes completed per unit time c) The total time a process spends in the system d) The total time taken from process submission to completion Answer: b) The number of processes completed per unit time	1	2	3	1	1.6.1
7	Identify the deadlock handling technique in the Banker's Algorithm. A) Deadlock Prevention B) Deadlock Avoidance C) Deadlock Detection and Recovery	1	2	3	2	2.1.3

	D) Deadlock Ignorance Answer: b) Deadlock Avoidance					
8	**What does interrupt latency refer to in real-time operating systems?** A) The time taken for a process to complete its execution. B) The time from the arrival of an interrupt to the start of the interrupt service routine (ISR). C) The time from the completion of an interrupt service routine (ISR) to the start of the next interrupt. D) The time from the arrival of an interrupt to the completion of the interrupt service routine (ISR). **Answer:** B) The time from the arrival of an interrupt to the start of the interrupt service routine (ISR).	**1**	**2**	**3**	**1**	1.6.1
9	Deadlock detection algorithms differ from deadlock prevention algorithms in that they ___________ A. Prevent deadlocks from occurring in the first place. B. Avoid deadlocks by pre-allocating resources. C. Detect and recover from deadlocks after they occur. D. Use priority scheduling to avoid resource conflicts. **Answer: C)** Detect and recover from deadlocks after they occur.	**1**	**1**	**3**	**1**	1.3.1
10	Which statement is true about the Non-Preemptive scheduling algorithm? a) Resources are allocated to a process for a limited time b) Process can be interrupted in between c) High Priority process frequently arrive in the ready Queue d) Process hold the resources till it complete its burst time or switching to wait state Answer: d) Process hold the resources till it complete its burst time or switching to wait state	**1**	1	3	1	1.3.1
	Part – B **(4 x 5 = 20 Marks)**					
11	**Elaborate the actions taken by the kernel to context-switch between processes. Sketch the steps in CPU Switching From Process to Process.** **Answer:** 1. Saving the Context of the Current Process Program Counter (PC): The address of the next instruction to be executed. Processor Registers: Includes general-purpose registers, status registers, and instruction registers. Stack Pointer (SP) and Frame Pointer (FP): These pointers need to be saved to restore the stack correctly	**5**	**L2**	**2**	**2**	**2.6.2**

later.

Memory Management Information: Page tables or segment descriptors, depending on the memory management scheme.

I/O State (if necessary): The state of I/O operations or any in-progress I/O activities.

2. Updating the Process Control Block (PCB)

Process State: The state of the process is updated to reflect that it is no longer running (e.g., to "waiting" or "ready").

CPU Registers: The saved context (registers, PC, SP, etc.) is stored in the process's PCB.

Accounting Information: CPU time used, scheduling parameters, and other relevant metrics are updated.

3. Choosing the Next Process to Run

Scheduling Algorithm: The kernel uses a scheduling algorithm (e.g., round-robin, priority-based, etc.) to select the next process to run.

Ready Queue: The next process is selected from the ready queue, which holds all processes that are ready to execute.

4. Loading the Context of the Next Process

Restore CPU Registers: The CPU registers, PC, SP, FP, and other necessary context information are loaded from the PCB of the next process.

Memory Management: The memory management unit (MMU) is updated with the memory map of the next process (e.g., updating the page table base register).

Set Process State: The state of the process is updated to "running."

5. Switching to the Next Process

Update Hardware Registers: This includes updating the PC to point to the instruction to be executed next in the new process.

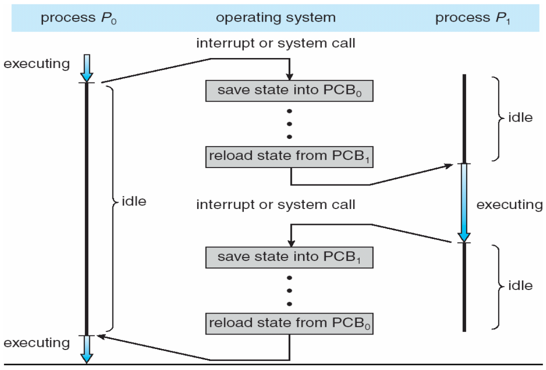
Execution Transfer: The CPU starts executing the new process from where it left off, according to the restored context.

6. Miscellaneous Operations

Cache Management: In some architectures, the CPU caches might need to be flushed or updated.

TLB (Translation Lookaside Buffer) Management: The TLB might need to be flushed or updated to reflect the new process's memory mappings.

Handling Special Hardware Contexts: If the system supports features like SIMD registers, floating-point registers, or other special hardware contexts, these must be managed accordingly.



12	**Write the pseudocode for the Bounded Buffer- producer-consumer problem.**	5	L2	2	3	2.6.4

Answer:

Producer

```
while (true) {
        /* produce an item in next produced */

        while (counter == BUFFER_SIZE) ;
                /* do nothing */
        buffer[in] = next_produced;
        in = (in + 1) % BUFFER_SIZE;
        counter++;   }
```

Consumer

```
while (true) {
        while (counter == 0)
                ; /* do nothing */
        next_consumed = buffer[out];
        out = (out + 1) % BUFFER_SIZE;
        counter--;
        /* consume the item in next consumed */ }
```

<table>
<tr>
<td>13</td>
<td>Consider the processing and its arrival time given in the table; predict the scheduling scheme among SJF and FCFS which gives lowest average turnaround time.

<table>
<tr><th>Process</th><th>Arrival Time</th><th>Processing Time</th></tr>
<tr><td>A</td><td>0</td><td>3</td></tr>
<tr><td>B</td><td>1</td><td>6</td></tr>
<tr><td>C</td><td>4</td><td>4</td></tr>
<tr><td>D</td><td>6</td><td>2</td></tr>
</table>

Ans :

Shortest job first:
Turn around time
A = 3
B=8
C=11
D=5
Average TAT = (3+8+11+5)/4 = 6.75

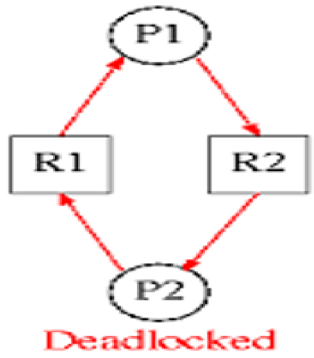
Shortest job first- 6.75(Average turnaround time)

FCFS:
Turn around time
A = 3
B=8
C=9
D=9
Average TAT = (3+8+9+9)/4 = 7.25

FCFS- 7.25 (Average turnaround time)

Shortest job first gives lowest average turnaround time</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2.1.3</td>
</tr>
<tr>
<td>14</td>
<td>In a system with two processes P1 and P2 and two resources R1 and R2, the following situation arises:
• P1 holds R1 and requests R2.
• P2 holds R2 and requests R1.
Construct the Resource Allocation Graph for the above situation and check whether deadlock has occurred. If a deadlock is detected, suggest a method to resolve the deadlock and justify your answer.

Answer:</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3.2.3</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

To resolve the deadlock, we can use one of the following methods:
1. Resource Preemption:
○ Preempt R1 from P1 or R2 from P2. Since both processes need one more resource to proceed, preempting one resource may break the cycle. For instance, preempt R2 from P2 and allocate it to P1, allowing P1 to complete and release both R1 and R2.
2. Process Termination:
○ Terminate one of the processes. For example, terminating P2 will release R2, allowing P1 to complete its execution. This is a more drastic measure but ensures the system proceeds.
Justification:
● Resource preemption is preferred in this case because it is less disruptive than process termination. By preempting a resource, we can allow one process to complete and release all its resources, potentially resolving the deadlock without losing any progress.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td colspan="7">Part – C
Either OR Choice Questions
(2 X 10 = 20 Marks)</td>
</tr>
<tr>
<td>15a</td>
<td>Let us consider a scenario in which a process P1 tries changing data in a particular memory location. At the same time another process P2 tries reading data from the same memory location.
a. Mention the rules to be following for preserving data consistency?
Answer:
Mutual Exclusion - If process P_i is executing in its critical section, then no other processes can be executing in their critical sections

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 processes that will enter the critical section next cannot be postponed</td>
<td>10</td>
<td>L4</td>
<td>2</td>
<td>3</td>
<td>2.6 .3</td>
</tr>
</table>

indefinitely

Bounded Waiting - A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted

- Assume that each process executes at a nonzero speed
- No assumption concerning **relative speed** of the ***n*** processes

b. Explain the classical solutions and Synchronization Hardware for preserving data consistency?

Answer:

Peterson's Solution

```
do {
    flag[i] = true;
    turn = j;
    while (flag[j] && turn = = j);
            critical section
    flag[i] = false;
            remainder section
} while (true);
```

Compare and swap

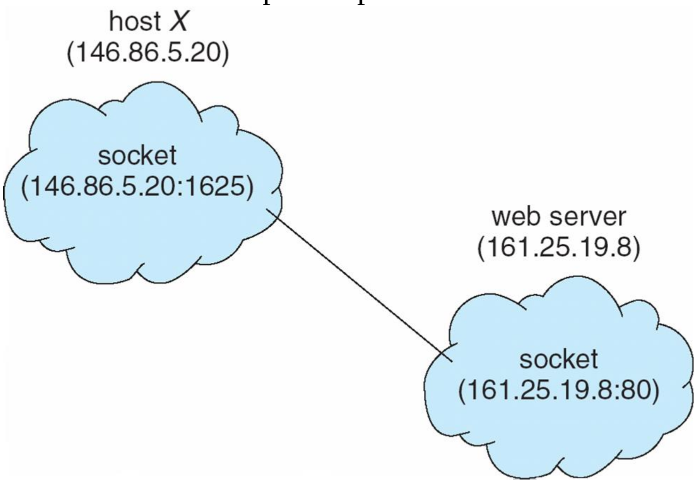
```
do {
    while (compare_and_swap(&lock, 0, 1) != 0)
      ; /* do nothing */
    /* critical section */
  lock = 0;
    /* remainder section */
} while (true);
```

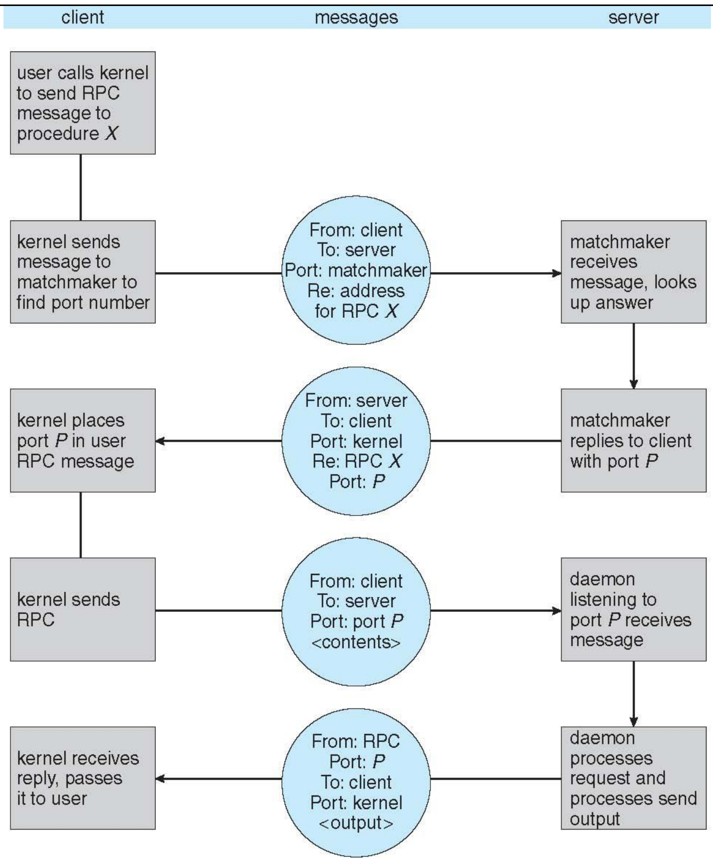
Test and set

```
do {
    while (test_and_set(&lock))
        ; /* do nothing */
            /* critical section */
    lock = false;
            /* remainder section */
} while (true);
```

Lock and Unlock

```
do {
    acquire lock
         critical section
    release lock
         remainder section
} while (TRUE);
```

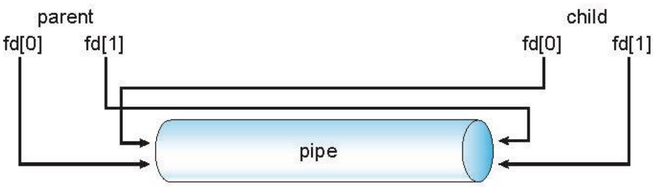
	(or)					
15b	Explain the three methods of communication in Client- Server system with the necessary diagrams. Answer : i)Socket Communication • Three types of sockets – Connection-oriented (TCP) – Connectionless (UDP) – MulticastSocket class– data can be sent to multiple recipients  **ii) Remote procedure call (RPC) abstracts procedure calls between processes on networked systems** • Again uses ports for service differentiation • Stubs – client-side proxy for the actual procedure on the server • The client-side stub locates the server and marshalls the parameters • The server-side stub receives this message, unpacks the marshalled parameters, and performs the procedure on the server • On Windows, stub code compile from specification written in Microsoft Interface Definition Language (MIDL)	**10**	**L4**	**2**	**2**	**2.6.3**



iii)Pipes

- Acts as a conduit allowing two processes to communicate

- Ordinary pipes – cannot be accessed from outside the process that created it. Typically, a parent process creates a pipe and uses it to communicate with a child process that it created.
- Named pipes – can be accessed without a parent-child relationship.
- Ordinary Pipes allow communication in standard producer-consumer style
- Producer writes to one end (the write-end of the pipe)
- Consumer reads from the other end (the read-end of the pipe)
- Ordinary pipes are therefore unidirectional
- Require parent-child relationship between communicating processes
-



- Named Pipes are more powerful than ordinary pipes

- Communication is bidirectional
- No parent-child relationship is necessary between the communicating processes
- Several processes can use the named pipe for communication
- Provided on both UNIX and Windows systems

16a	**Consider the following example with seven processes and implement Preemptive Priority Scheduling**	**10**	3	3	3	3.2.3

PID	Priority	Arrival Time	Burst Time
P1	2(low)	0	4
P2	4	1	2
P3	6	2	3
P4	10	3	5
P5	8	4	1
P6	12(high)	5	4
P7	9	6	6

Draw the Gantt Chart and Calculate the following for all the processes,

i) **Completion time**
ii) **Turn Around Time**
iii) **Waiting Time**

Answer :

Gantt Chart :

P1	P2	P3	P4	P6	P4	P7	P5	P3	P2	P1

0 1 2 3 5 9 12 18 19 21 22 25

PID	Priority	Arrival Time	Burst Time	Completion Time(CT)	Turn Around Time(TAT)	Waiting Time (WT)
P1	2(low)	0	4	25	25	21
P2	4	1	2	22	21	19
P3	6	2	3	21	19	16
P4	10	3	5	12	9	4
P5	8	4	1	19	15	14
P6	12(high)	5	4	9	4	0
P7	9	6	6	18	12	6

(Or)

16b	**Consider the following snapshot of a system:**	**10**	3	3	3	3.2.3

	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	

Answer the following questions 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 P1 arrives for (0,4,2,0), can the request be granted immediately

Answer:

a. The values of Need for processes P0 through P4 respectively are (0, 0, 0, 0), (0, 7, 5, 0), (1, 0, 0, 2), (0, 0, 2, 0), and (0, 6, 4, 2).

b. The system is in a safe state? Yes. With Available being equal to (1, 5, 2, 0), either process P0 or P3 could run. Once process P3 runs, it releases its resources, which allow all other existing processes to run.

c. The request can be granted immediately? This results in the value of Available being (1, 1, 0, 0). One ordering of processes that can finish is P0, P2, P3, P1, and P4.

Course Outcome (CO) and Bloom's level (BL) Coverage in Questions