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

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu **SET-3**

Academic Year: 2022-23 (EVEN)

Test: CLA-T1 Date: 28-Mar-2023 Course Code & Title: 18CSC205J & Operating Systems Duration: 2 Hours **Year & Sem: II Year / V Sem 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
1	CO1	3		3									
2	CO2	2	1	3									
3	CO3	3	2	2									
4	CO4	3	2	2									
5	CO5	3		2	2								

Q. No	Question

1) Consider the methods used by processes P_1 and P_2 for accessing their critical sections whenever needed, as given below. The initial values of shared Boolean variables S_1 and S_2 are randomly assigned. Discuss the execution of the processes for all possible values of S_1 and S_2 **Process P_1** **Process P_2**

while ($S_1 < S_2$);
while ($S_1 > S_2$);
<<critical section>>
<<critical section>>
swap(S_1 , S_2);
swap(S_1 , S_2);

It can be easily observed that the Mutual Exclusion requirement is satisfied by the above solution, P1 can enter critical section only if S1 is greater than S2, and P2 can enter critical section only if S1 is lesser than S2. But here Progress Requirement is not satisfied. Suppose when s1=1 and s2=0 and process p1 is not interested to enter into critical section but p2 want to enter critical section. P2 is not able to enter critical section in this as only when p1 finishes execution, then only p2 can enter. Progress will not be satisfied when any process which is not interested to enter into the critical section will not allow other interested process to enter into the critical section.

b) Compare monitor with semaphore. What are the advantages of using monitors? (Any two 2

S. No.	Semaphore	Monitor
1.	It is an integer variable.	It is an abstract data type.
2.	The value of this integer variable tells about the number of shared resources that are available in the system.	It contains shared variables.
3.	When any process has access to the shared resources, it performs the 'wait' operation (using wait method) on the semaphore.	It also contains a set of procedures that operate upon the shared variable.
4.	When a process releases the shared resources, it performs the 'signal' operation (using signal method) on the semaphore.	When a process wishes to access the shared variables in the monitor, it has to do so using procedures.
5.	It doesn't have condition variables.	It has condition variables.

M)

Advantages of Monitors (Any 1) 1 M

Monitors have built-in mutual exclusion.

Monitors are easier to set up than semaphores.

2	

a. Compare Priority Scheduling with Shortest Job First Scheduling policy. At what situation, the Priority Scheduling will be same as SJF Scheduling?

Any Two -2M

Shortest job first (SJF)	Priority scheduling
Shortest Job First (SJF) executes the processes based upon their burst time i.e. in ascending order of their burst times.	Priority scheduling executes the processes based upon their priorities i.e. in descending order of their priorities. A process with higher priority is executed first.
SJF is also non-preemptive but its preemptive version is also there called Shortest Remaining Time First (SRTF) algorithm.	Priority scheduling is both preemptive and non preemptive in nature.
The average waiting time for given set of processes is minimum.	There is no idea of average waiting time and response time.
The real difficulty with SJF is knowing the length of the next CPU request or burst.	It is quite easy to implement and best for real time operating systems.
A long process may never get executed and the system may keep executing the short processes.	The problem of blocking of a process can be solved through aging which means to gradually increase the priority of a process after a fixed interval of time by a fixed number.

SJF works similar to priority - The smaller the expected burst, the higher the priority. 1 M

b. Consider the set of 5 processes whose arrival time and burst time are given below. If the CPU scheduling policy is priority preemptive, calculate the average waiting time and average turnaround time. (Higher number represents higher priority).

Process ID Arrival Time Burst Time Priority

P1 0 4 2

P2 1 3 3

P3 2 1 1

P4 3 5 4

P5 4 2 5

	AT	BT	Priority	CT	TAT	WT
P1	0	4	2	14	14	10
P2	1	3	3	11	7	4
P3	2	1	1 (Low)	15	13	12
P4	3	5	4	10	7	2
P5	4	2	5 (High)	6	2	0

Gantt chart

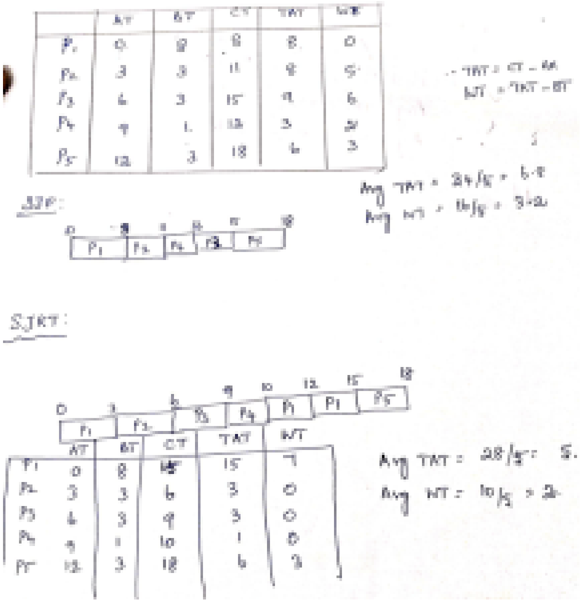
0–1	1–2	2–3	3–4	4–6	6–10	10–11	11–14	14–15
P1	P2	P2	P4	P5	P4	P2	P1	P3

Avg TAT = 43/5 = 8.6

Avg WT = 28/5 = 5.6

3

Apply Shortest Job First scheduling and Shortest remaining time next (SRTN) with preemptive approach using the given arrival time. The system has 5 processes entering in a ready queue with the time interval of 3ms starts with 0ms respectively. The running time of the process required is 8,3,3,1,3 respectively. Draw the Gantt chart and find average waiting time and average turn-around time for the scheduling



	AT	BT	CT	TAT	WT
P1	0	8	8	8	0
P2	3	3	11	8	5
P3	6	3	15	9	6
P4	9	1	12	3	2
P5	12	3	18	6	3

∴ TAT = CT − AT
WT = TAT − BT

Avg TAT = 34/5 = 6.8
Avg WT = 16/5 = 3.2

SJF:

0 8 11 12 15 18
| P1 | P2 | P4 | P3 | P5 |

SJRT:

0 3 6 9 10 12 15 18
| P1 | P2 | P3 | P4 | P1 | P1 | P5 |

	AT	BT	CT	TAT	WT
P1	0	8	15	15	7
P2	3	3	6	3	0
P3	6	3	9	3	0
P4	9	1	10	1	0
P5	12	3	18	6	3

Avg TAT = 28/5 = 5.
Avg WT = 10/5 = 2.

4	A system has 4 processes and 5 allocatable resources. The current allocation and maximum needs are as follows **Allocated Maximum** **A** 1 0 2 1 1 1 1 2 1 3 **B** 2 0 1 1 0 2 2 2 1 0 **C** 1 1 0 1 1 2 1 3 1 1 **D** 1 1 1 1 0 1 1 2 2 0 If Available = [0 0 X 1 1], what is the smallest value of x for which this is a safe state? Solution: <table><tr><td></td><td colspan="5">Need</td></tr><tr><td>A</td><td>0</td><td>1</td><td>0</td><td>0</td><td>2</td></tr><tr><td>B</td><td>0</td><td>2</td><td>1</td><td>0</td><td>0</td></tr><tr><td>C</td><td>1</td><td>0</td><td>3</td><td>0</td><td>0</td></tr><tr><td>D</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table> **Case-01: For X = 0** If X = 0, then Available= [0 0 0 1 1] With the instances available currently, the requirement of any process can not be satisfied. So, for X = 0, system remains in a deadlock which is an unsafe state.

	Case-02: For X = 1
	If X = 1, then
	Available= [0 0 1 1 1]
	Step-01:
	With the instances available currently, only the requirement of the process D can be satisfied. So, process D is allocated the requested resources. It completes its execution and then free up the instances of resources held by it.
	
	Then
	Available= [0 0 1 1 1] + [1 1 1 1 0]= [1 1 2 2 1]
	With the instances available currently, the requirement of any process can not be satisfied. So, for X = 1, system remains in a deadlock which is an unsafe state.
	
	Case-02: For X = 2
	If X = 2, then
	Available= [0 0 2 1 1]
	Step-01:
	With the instances available currently, only the requirement of the process D can be satisfied. So, process D is allocated the requested resources.It completes its execution and then free up the instances of resources held by it.
	Then
	Available= [0 0 2 1 1] + [1 1 1 1 0]= [1 1 3 2 1]
	Step-02:
	With the instances available currently, only the requirement of the process C can be satisfied. So, process C is allocated the requested resources.It completes its execution and then free up the instances of resources held by it.
	Then-Available= [1 1 3 2 1] + [1 1 0 1 1]= [2 2 3 3 2]
	Step-03:
	With the instances available currently, the requirement of both the processes A and B can be satisfied. So, processes A and B are allocated the requested resources one by one.They complete their execution and then free up the instances of resources held by it.
	Then-Available= [2 2 3 3 2] + [1 0 2 1 1] + [2 0 1 1 0]= [5 2 6 5 3]
	Thus, There exists a safe sequence in which all the processes can be executed.
	So, the system is in a safe state.Thus, minimum value of X that ensures system is in safe state = 2.

5	a) A system shares 9 tape drives. The current allocation and maximum requirement of tape drives for 3 processes are shown below. Check whether the system is safe. Process Current Allocation Maximum requirement P1 3 7 P2 1 6 P3 3 5 Need of p1,p2,p3 is 4,5,2 Available = 9-(3+1+3)=2 2 Resources will be satisfied by P3 Now Available = 2+3=5 5 ca be used by p2 0r p1 Case 1: If ps is executed before p1, Available = 5+1=6 6 resources can satisfy the need of p1. Safe sequence will be p3,p1,p2 or p3,p2,p1. No Deadlock.

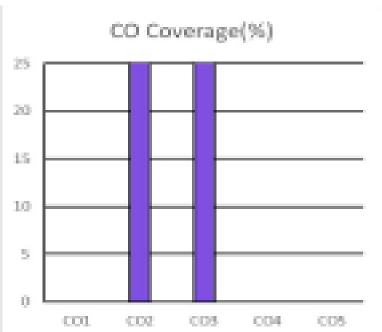
	b) Differentiate Internal fragmentation problem with External Fragmentation problem. Discuss the solution for solving these fragmentation problems with example. Any two – 2M Example 3M

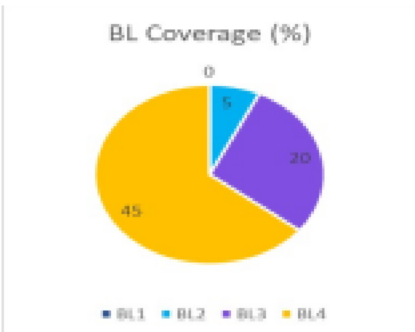
S.NO	Internal fragmentation	External fragmentation
1.	In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to the method.
2.	Internal fragmentation happens when the method or process is smaller than the memory.	External fragmentation happens when the method or process is removed.
3.	The solution of internal fragmentation is the best-fit block.	The solution to external fragmentation is compaction and paging.
4.	Internal fragmentation occurs when memory is divided into fixed-sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
5.	The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, which is called External fragmentation.
6.	Internal fragmentation occurs with paging and fixed partitioning.	External fragmentation occurs with segmentation and dynamic partitioning.
7.	It occurs on the allocation of a process to a partition greater than the process's requirement. The leftover space causes degradation system performance.	It occurs on the allocation of a process to a partition greater which is exactly the same memory space as it is required.
8.	It occurs in worst fit memory allocation method.	It occurs in best fit and first fit memory allocation method.

6	(a) Draw the multi programming fixed size partitioning of size 7KB, 20KB, 25KB, 3KB, 10KB, 5KB in partitioning in the main memory and in that allocate the processor of 6KB, 15KB, 20KB, 1KB, 2KB, 3KB. Find the wastage in the given partitioning and determine the total wastage in the main memory. **Solution** 6KB process is allocated to 7KB partition, 15KB to 20KB partition and so on.. So the wastage of space is (7-6) + (20-15) + (25-20) + (3-1) + (10-2) + (5-3) = 1 + 5 + 5 + 2 + 8 + 2 **= 23** (b) Draw the multi programming fixed size partitioning of size 5KB, 5KB, 5KB, 5KB, 5KB, 5KB in partitioning in the main memory and in that allocate the processor of 6KB, 15KB, 20KB, 1KB, 2KB, 3KB. Find the wastage in the given partitioning and determine the total wastage in the main memory. Find which memory allocation algorithm determines less wastage by comparing (a) and (b). **Solution:** 6KB, 15KB and 20KB cannot be stored in memory as there is no partition with its size. So remaining 3 processes will be assigned to memory like 1KB to 5KB partition, 2KB to 5KB and 3KB to 5KB. So the wastage of space is (5-1) + (5-2) + (5-3) = 4 + 3 +2 **= 9** **Conclusion :** Solution (a) has more wastage than (b) but solution (a) accommodate all the processes.
7	Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size is 4 KB, what is the approximate size of the page table? (10 Marks) $\text{Physical memory} = 64MB = 2^{26}B$ $\text{Size of frame} = 4KB = 2^{12}B$ $\text{No. of frames} = \text{physical memory/size of frame} = \frac{2^{26}B}{2^{12}B} = 2^{14}$ $\text{Frame number} = 14bits$ $\text{Virtual memory} = 32bits = 2^{32}B$ $\text{Size of page} = \text{size of frame} = 4KB = 2^{12}B$ $\text{No. of pages} = \text{virtual memory/size of page} = \frac{2^{32}B}{2^{12}B} = 2^{20}$ $\text{size of page table} = \text{Number of pages} \times \text{Size of each entry}$ $\text{size of page table} = \text{Number of pages} \times \text{Page table entry}$ $\text{size of page table} = \text{Number of pages} \times \text{Frame number}$ $\text{size of page table} = 2^{20} \times 14bits$ $\text{assume Frame number} = 16bits$ $\text{size of page table} = 2^{20} \times 16bits \approx 2MB$

***Program Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

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





Approved by the Audit Professor/Course Coordinator