SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, RAMAPURAM

DEPARTMENT OF ---------CSE--------

ANSWER KEY SUBMISSION

Date of Exam & Session	7/6/2022 & FN	**Category of Exam**	~~CLA1~~/CLA2/~~CLA3/SURPRISE TEST~~
Course Name	Operating Systems	**Course Code**	18CSC205J
Name of the Faculty submitting	Dr.C.G.Balaji	**Date of submission of Answer Key**	9/6/2022
Department to which the Faculty belongs to	CSE	**Total Marks**	50

	Part-A (20 x 1 = 20)	
1	Process synchronization can be done on _____ a) Physical b) Hardware c) Software **d) Both hardware and Software**	1
2	_____ is interprocess communication. a) communication within the process **b) communication between two process** c) communication between two threads of same process d) communication between devices	1
3	_____ is a classic software-based solution to the critical-section problem **a) Peterson's solution.** b) Synchronization hardware c) TestAndSet Instruction d) SetAndTest Instruction	1
4	Semaphore can be accessed via two operations namely _____ operations a) stop() and run() b) pause() and unpause() **c) proberen() and verhogen()** d) call() and wait()	1
5	In _____semaphore, the integer value can range over an unrestricted domain a) **counting** b) binary c) bounded d) unbounded	1
6	_____ is/are classical problem of synchronization. a) Bounded-Buffer Problem b) Readers and Writers Problem c) Dining-Philosophers Problem d) Bounded-Buffer Problem and Readers and Writers Problem e) **Bounded-Buffer Problem, Readers and Writers Problem and Dining-Philosophers Problem**	1
7	Which of the following scheduling algorithms is preemptive scheduling? a) FCFS Scheduling b) SJF Scheduling c) Network Scheduling **d) SRTF Scheduling**	1
8	CPU scheduling decisions may take place when a process a) switches from waiting to running state b) switches from terminate to ready state	1

	c) switches from waiting to ready state d) switches from waiting to new state	
9	Which module gives control of the CPU to the process selected by the short-term scheduler? a) **dispatcher** b) interrupt c) scheduler d) MMU	1
10	Deadlock can arise if following condition/s hold simultaneously a) Mutual Exclusion only b) Mutual Exclusion and Hold & wait c) Mutual Exclusion, Hold & wait, No preemption **d) Mutual Exclusion, Hold & wait, No preemption, Circular wait**	1
11.	Logical address is generated by the_____ a) memory manager **b) CPU** c) memory controller d) memory unit	1
12.	Copying a process from memory to disk to allow space for other process is known as _____ a) fragmentation b) paging **c) Swapping** d) Demand Paging	1
13.	Paging sets up a _____table to translate logical to physical addresses a) Segment b) Mapping c) Logical d) **Page**	1
14.	The page base contains the _____ a) starting logical address of the process **b) starting physical address of the page in memory** c) page length d) ending logical address of the process	1
15.	Solution to the problem of external fragmentation problem is to _____ **a) permit the logical address space of a process to be noncontiguous** b) permit smaller processes to be allocated memory at last c) permit larger processes to be allocated memory at last d) permit the logical address space of a process to be contiguous	1
16.	In segmentation, each address is specified by _____ a) a key & value b) an offset & value c) a value & segment number **d) a segment number & offset**	1
17.	Consider a computer with 8 Mbytes of main memory and a 128K cache. The cache block size is 4 K. It uses a direct mapping scheme for cache management. How many different main memory blocks can map onto a given physical cache block? a) 2048 b) 256 **c) 64** d) 8	1
18	The PTBR contains the _____ a) starting logical address of the process **b) starting address of the page table** c) page length d) ending logical address of the process	1
19	The 64-bit x86-64 architecture supports the following page _____ **a) Four levels of paging** b) Two levels of paging c) One level paging	1

	d) Five levels of paging	
20	In 32-bit ARM architecture, One-level paging is used for _____ sections a) 1-MB b) 4-KB and 16-KB pages **c) 1-MB and 16-MB** d) 16-KB	1
	Part-B (3 x 10 = 30)	
21.a.	**Define semaphore. Explain the use of semaphore in synchronization problem with an example.**	
	A semaphore is a variable or abstract data type used to control access to a common resource by multiple threads and avoid critical section problems in a concurrent system such as a multitasking operating system. Semaphores are a type of synchronization primitives.	2
	• Semaphores are used for mutual exclusions where the semaphore has an initial value of one, and P () and V () are called before and after the critical sections.	2
	• The following problems of synchronization are considered as classical problems: ○ Bounded-buffer (or Producer-Consumer) Problem, ○ Dining-Philosophers Problem, ○ Readers and Writers Problem, ○ Sleeping Barber Problem	4
	• For example, Suppose there are 4 processes P1, P2, P3, P4, and they all call wait operation on S (initialized with 4). If another process P5 wants the resource then it should wait until one of the four processes calls the signal function and the value of semaphore becomes positive.	2
21.b.	**Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use non-preemptive scheduling and base all decisions on the information you have at the time the decision must be made.**	

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

a) Find the average turnaround time for these processes with the FCFS scheduling algorithm?

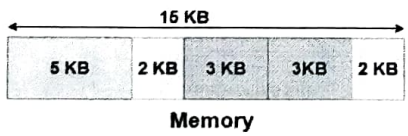
Avg. TAT = (8 + (12 - 0.4) + (13 - 1)) / 3 = 10.53 — 3

b) Find the average turnaround time for these processes with the SJF scheduling algorithm?

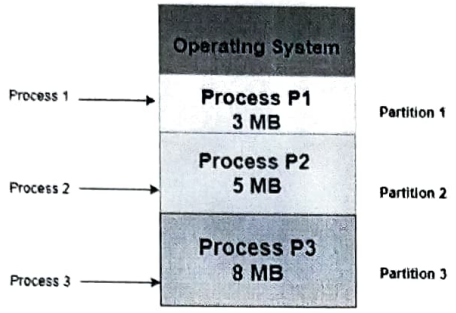
Avg. TAT = (8 + (9 - 1) + (13 – 0.4)) / 3 = 9.53 — 3

c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Find what is the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used.

Avg. TAT = ((2 - 1) + (6 – 0.4) + (14 - 0)) / 3 = 6.87 — 4

22.a.	**Illustrate contiguous memory allocation schemes with examples.**	
	In the Contiguous Memory Allocation, each process is contained in a single contiguous section of memory. The memory can be divided either in the **fixed-sized partition** or in the **variable-sized partition** in order to allocate contiguous space to user processes. (with Similar Example Figures)	2
		4
		4

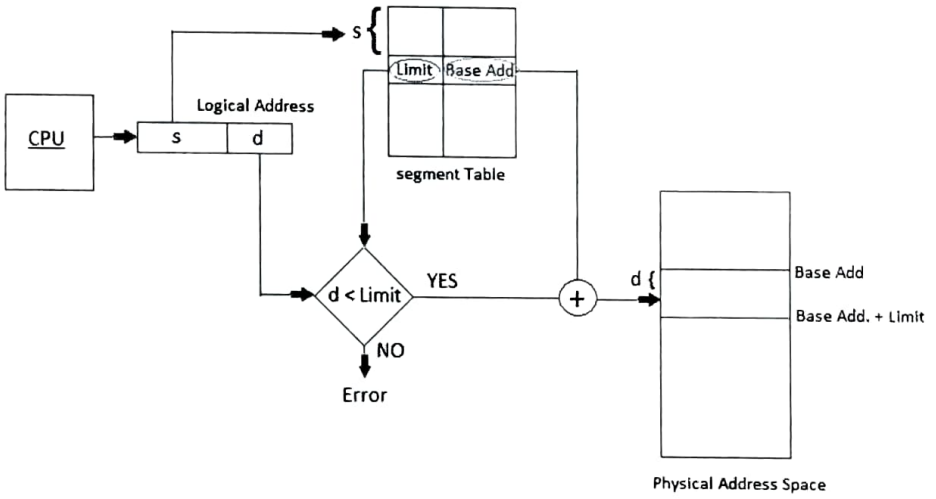
The size of each partition is **fixed** as indicated by the name of the technique and it cannot be changed. After allocation, if there is some wastage inside the partition then it is termed **Internal Fragmentation**. **Variable-sized partition** scheme is also known as Dynamic partitioning and is came into existence to overcome the drawback i.e internal fragmentation that is caused by Static partitioning.



Size of Partition =Size of Process

2

22.b. Draw the diagram of segmentation memory management scheme and summarize its principle.



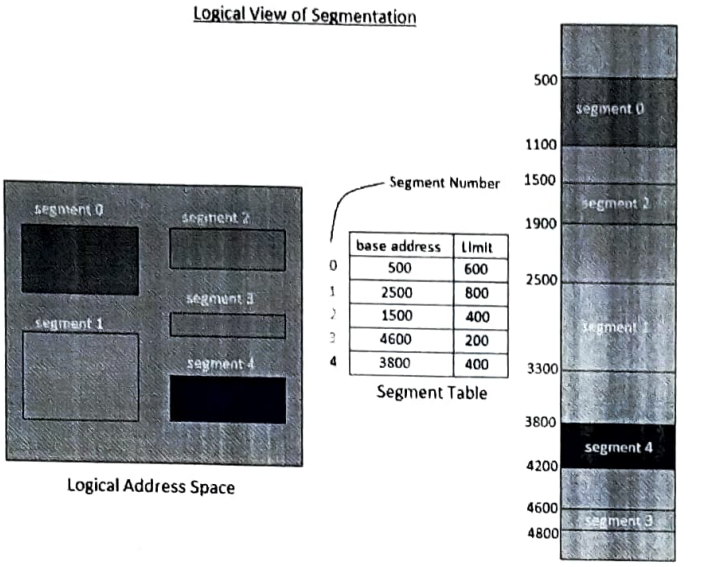
2

4

- A process is divided into Segments.
- Segmentation gives user's view of the process which paging does not give.
- There is no simple relationship between logical addresses and physical addresses in segmentation.
- A table stores the information about all such segments and is called Segment Table.

2

- **Segment Table** – It maps two-dimensional Logical address into one-dimensional Physical address.



It's each table entry has:

- **Base Address:** It contains the starting physical address where the segments reside in memory. 1
- **Limit:** It specifies the length of the segment.

Address generated by the CPU is divided into:

- **Segment number (s):** Number of bits required to represent the segment. 1
- **Segment offset (d):** Number of bits required to represent the size of the segment.

23.a. **Assume that there are three resources, A, B, and C. There are 5 processes P0 to P4. At T0 we have the following snapshot of the system**

Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	3	3	2
P1	2	0	0	3	2	2			
P2	3	0	2	9	0	2			
P3	2	1	1	2	2	2			
P4	0	0	2	4	3	3			

Answer the following based on banker's algorithm:

a) What is the content of need matrix? 3

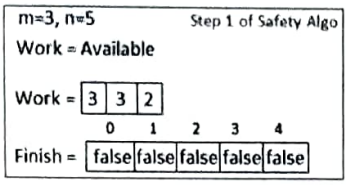
Need [i, j] = Max [i, j] – Allocation [i, j]

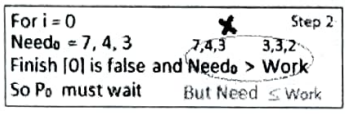
So, the content of Need Matrix is:

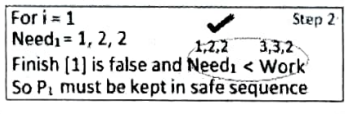
Process	Need		
	A	B	C
P_0	7	4	3
P_1	1	2	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

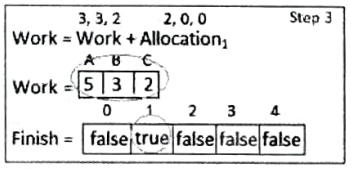
b) Is the system in a safe state?

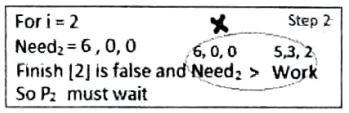
Applying the Safety algorithm on the given system,

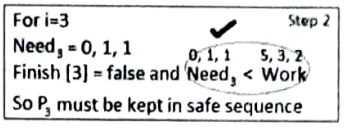


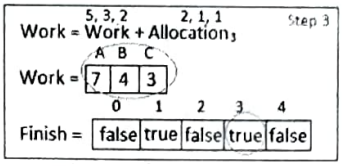


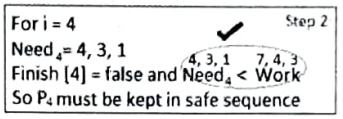


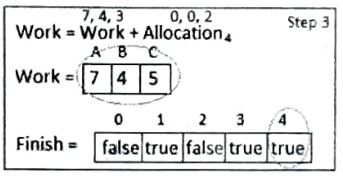


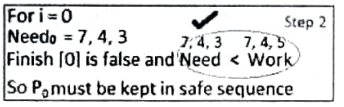


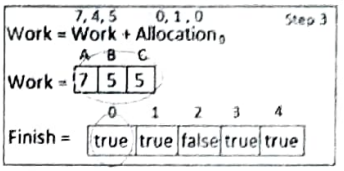


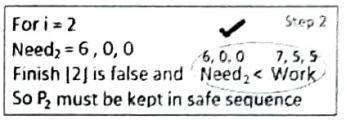


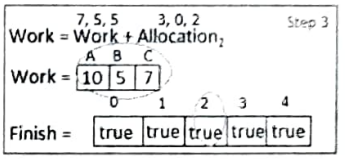


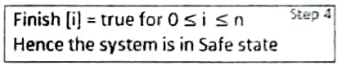












The safe sequence is P_1, P_3, P_4, P_0, P_2

4

C) If a request from process P1 arrives for (1,0,0) can the request be granted immediately?

A B C

$Request_1$ = 1, 0, 0

To decide whether the request is granted we use Resource Request algorithm

3

Step 1

1, 0, 0 ≤ 1, 2, 2

$Request_1 \leq Need_1$

Step 2

1, 0, 0 ≤ 3, 3, 2

$Request_1 \leq Available$

Step 3

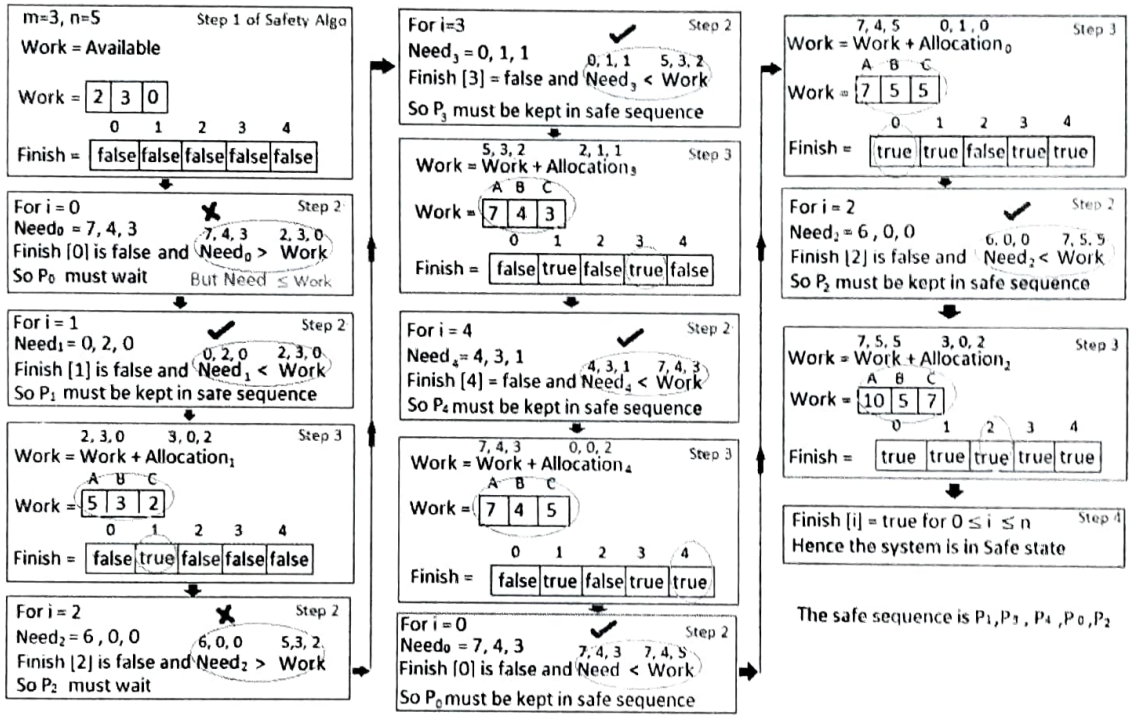
$Available = Available - Request_1$

$Allocation_1 = Allocation_1 + Request_1$

$Need_1 = Need_1 - Request_1$

Process	Allocation	Need	Available
	A B C	A B C	A B C
P_0	0 1 0	7 4 3	2 3 2
P_1	3 0 0	0 2 2	
P_2	3 0 2	6 0 0	
P_3	2 1 1	0 1 1	
P_4	0 0 2	4 3 1	

We must determine whether this new system state is safe. To do so, we again execute Safety algorithm on the above data structures.



Hence the new system state is safe, so we can immediately grant the request for process P_1.

23.b. Describe the various techniques for structuring the page table in a page memory management scheme.

A page table is the data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses. **2**

The following are the most common techniques for structuring the page table – Hierarchical Paging, Hashed Page Tables, and Inverted Page Tables. **1**

Hierarchical Paging: **3**

Two level

	Three level **Hashed Page Table** **Inverted Page Table**	 **2** **2**