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

Mode of Exam
OFFLINE
SET B

DEPARTMENT OF COMPUTING TECHNOLOGIES

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2021-2022 (Even)

Test: CLAT-2 **Date: 26-05-2022**
Course Code & Title: 18CSC205J: Operating Systems **Duration:** 2 Period
Year & Sem: 2022 & IV Semester **Max. Marks:** 50 Marks

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

Part - A
(10 x 1 = 10 Marks)

Instructions: Answer all

Q. No	Question	Marks	BL	CO	PO	PI Code
1	b. I and II	1	3	2	3	3.6.1
2	c. II and IV	1	3	2	3	3.6.1
3	d. execution of wait() and signal()	1	1	2	3	3.6.1
4	c. i and iii only	1	3	2	3	3.6.1
5	d. 12	1	3	2	3	3.6.1
6	a. the number of partitions	1	1	3	1	3.6.1
7	d. Compilation, loading, and execution	1	1	3	1	1.6.1
8	c. Relocatable register	1	1	3	1	1.6.1
9	a. context switching	1	1	3	1	1.6.1
10	a. frames, pages	1	1	3	1	1.6.1

Part - B
(4 x 5 = 20 Marks)

Instructions: Answer Any 5

11	**Explain why the following lock does not work? Locked is a Boolean which indicates the lock is currently locked. Also describe a hardware assisted solution, how does it solve this problem?** **while (locked) {** **// do nothing** **}** **locked = TRUE;** ANSWER If two threads retrieve the value of "locked" in the while statement at the same time and it is false then they would both set it to true and carry on to the critical section. The problem is that reading the value of "locked" is separate from setting it. By providing an atomic instruction on a processor	5	4	2	3	3.7.1

	such as testandset the value of the “locked” variable is returned at the same time as it is set. This means there is no way two threads can find the lock free at the same time.					
12	**The semaphores wait and signal operations are defined as indivisible (or atomic) operations. Why do they have to be indivisible? Because they are indivisible does that mean that all other processes running on a multiprocessor system must stop when a wait or signal operation is executed? Explain why or why not**. Answer : If they weren't indivisible it would be possible for multiple waits and signals to occur simultaneously (on a multiprocessor) or at least interleaved with each other. This would corrupt the value of the semaphore and prevent it working properly. No. Strictly only processes accessing the same semaphore at that time need to stop. Any processes not wanting to change or access the semaphore should be allowed to continue.	**5**	**4**	**2**	**3**	**3.7.1**
13	**Sharing segments among processes without requiring that they have the same segment number is possible in a dynamically linked segmentation system. a. Define a system that allows static linking and sharing of segments without requiring that the segment numbers be the same. b. Describe a paging scheme that allows pages to be shared without requiring that the page numbers be the same.** Both of these problems reduce to a program being able to reference both its own code and its data without knowing the segment or page number associated with the address. MULTICS solved this problem by associating four registers with each process. One register had the address of the current program segment, another had a base address for the stack, another had a base address for the global data, and so on. The idea isthat all references have to be indirect through a register that maps to the current segment or page number. By changing these registers, the same code can execute for different processes without the same page or segment numbers	**5**	**4**	**3**	**3**	**3.6.4**

14	**Why is that, on a system with paging, a process cannot access memory it does not own? How could the operating system allow access to other memory? Why should it or should it not?** An address on a paging system is a logical page number and an offset. The physical page is found by searching a table based on the logical page number to produce a physical page number. Because the operating system controls the contents of this table, it can limit a process to accessing only those physical pages allocated to the process. There is no way for a process to refer to a page it does not own because the page will not be in the page table. To allow such access, an operating system simply needs to allow entries for non-process memory to be added to the process' page table. This is useful when two or more processes need to exchange data - they just read and write to the same physical address (which may be at varying logical addresses) . This makes for very efficient interprocess communication.	**5**	**3**	**3**	**1**	**1.7.1**
15	**Fill in the blanks(priority of the processes)** Process CPU burst time Priority 1 5 2 4 ------ 3 8 ------ 4 7 ------ 5 3 4 **Out of five processes, process1 takes priority "1" and process 5 takes priority "4". Find the suitable priority for the remaining processes among 2,3 and 5, in order to achieve the average waiting time "11".** Answer: Process 1= Priority 1 Process 2 = Priority 3 Process 3 = Priority 2 Process 4 = Priority 5 Process 5 = Priority 4 Avg WT = 55/5=11	**5**	**5**	**2**	**3**	**3.6.2**
16.a	**There are 5 processes P0, P1, P2, P3 & P4 in a system and 4 resource types A, B, C & D. There are 5 instance of A, 12 instance of B, 11 instance of C and 10 instance of D are available. The resources allocated to each process and the**	**10**	**5**	**2**	**3**	**3.6.2**

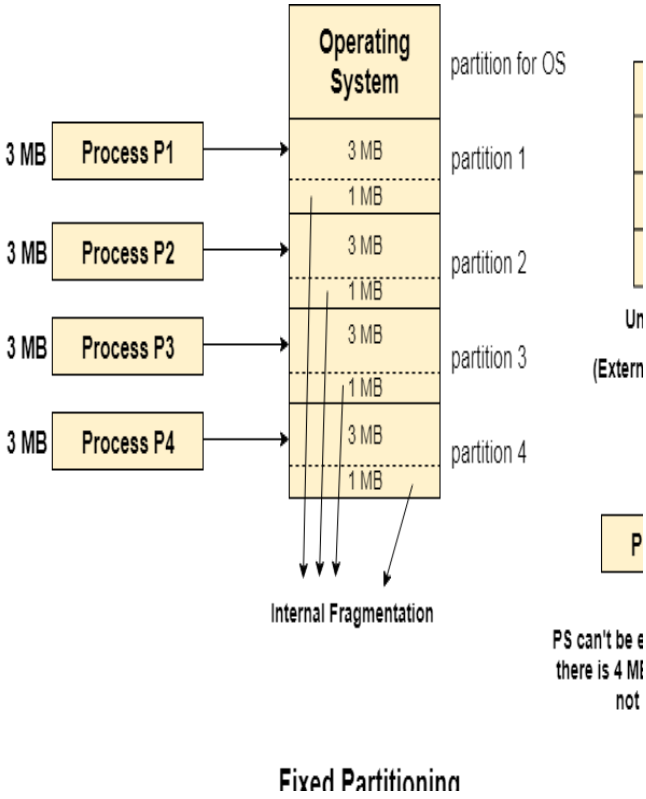
	maximum resources required by each process are given in in the following table. **Using Bankers algorithm generate the sequence of process for which the system is in safe state.** **Soln** <P0,P2,P1,P4.P3> satisfies the safety sequence									
16.b	**Consider a scenario where the following processes are arriving in the given time and their CPU burst is given in the milliseconds** **Which of the following algorithm would be more suitable for this given scenario to yield best average waiting and average turnaround time?** **1. Shortest remaining time first** **2. Shortest job first** **Justify your answer whether the preemptive scheduling or non-preemptive scheduling work well for this case. And also specify the difference in the average waiting time by those two algorithms.** Answer:	**10**	**5**	**2**	**3**	**3.6.4**

	Allocation				Maximum				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	1	1	1	2	1	1	0	0	3	7	1	1
P1	0	2	0	0	0	7	3	0	0	5	3	0				
P2	1	3	5	2	2	3	7	4	1	0	2	2				
P3	0	0	3	1	4	0	5	8	4	0	2	7				
P4	1	0	1	4	5	5	5	5	4	5	4	1				

	Allocation				Maximum			
	A	B	C	D	A	B	C	D
P0	0	0	1	2	1	1	1	2
P1	0	2	0	0	0	7	3	0
P2	1	3	5	2	2	3	7	4
P3	0	0	3	1	4	0	5	8
P4	1	0	1	4	5	5	5	5

Process	Time	Burst time
1	0	5
2	2	8
3	4	7
4	5	10
5	6	4

	SRTF is SJF with Pre-emption Gantt Chart for both SJF and SRTF SJF : Avg WT =8 Avg TAT = 14.8 SRTF : Avg WT = 7.6 Avg TAT = 14.4 SRTF is preferred for these processes.					
17.a	**Explicate the simplest technique used to load more than one processes into the main memory, which helps to split into equal or different sizes of the memory. The operating system always resides in the first half or split while the other partitions can be used to store user processes. The memory is assigned to the processes in contiguous way; demonstrate with suitable examples and diagrams.** In fixed partitioning, 1. The partitions cannot overlap. 2. A process must be contiguously present in a partition for the execution. There are various cons of using this technique. **Internal Fragmentation** If the size of the process is lesser then the total size of the partition then some size of the partition get wasted and remain unused. This is wastage of the memory and called internal fragmentation. As shown in the image below, the 4 MB partition is used to load only 3 MB process and the remaining 1 MB got wasted. **External Fragmentation** The total unused space of various partitions cannot be used to load the processes even though there is space available but not in the contiguous form. As shown in the image below, the remaining 1 MB space of each partition cannot be used as a unit to store a 4 MB process. Despite of the fact that the sufficient space is available to load the process,	**10**	**2**	**3**	**1**	**1.7.1**

	process will not be loaded. **Limitation on the size of the process** If the process size is larger than the size of maximum sized partition then that process cannot be loaded into the memory. Therefore, a limitation can be imposed on the process size that is it cannot be larger than the size of the largest partition. **Degree of multiprogramming is less** By Degree of multi programming, we simply mean the maximum number of processes that can be loaded into the memory at the same time. In fixed partitioning, the degree of multiprogramming is fixed and very less due to the fact that the size of the partition cannot be varied according to the size of processes.  Fixed Partitioning					
17.b	**Associate the technique used for Structuring the page table,** **Case 1: There might be a case where the page table is too big to fit in a contiguous space may have a hierarchy with several levels.**	**10**	**2**	**3**	**2**	**2.6.2**

Case 2: Identify the approach is used to handle address spaces that are larger than 32 bits.

Some of the common techniques that are used for structuring the Page table are as follows:

1. Hierarchical Paging
2. Hashed Page Tables

Hierarchical Paging

Another name for Hierarchical Paging is multilevel paging.

- There might be a case where the page table is too big to fit in a contiguous space, so we may have a hierarchy with several levels.
- In this type of Paging the logical address space is broke up into Multiple page tables.
- Hierarchical Paging is one of the simplest techniques and for this purpose, a two-level page table and three-level page table can be used.

Two Level Page Table

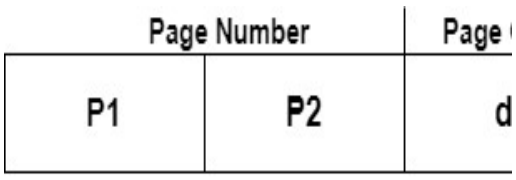
Consider a system having 32-bit logical address space and a page size of 1 KB and it is further divided into:

- Page Number consisting of 22 bits.
- Page Offset consisting of 10 bits.

As we page the Page table, the page number is further divided into :

- Page Number consisting of 12 bits.
- Page Offset consisting of 10 bits.

Thus the Logical address is as follows:



In the above diagram,

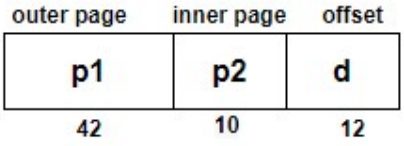
P1 is an index into the **Outer Page** table.

P2 indicates the displacement within the page of the **Inner page** Table.

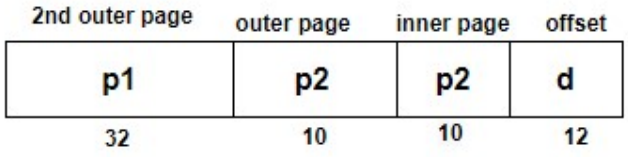
As address translation works from outer page table inward so is known as **forward-mapped Page Table**.

Below given figure below shows the Address Translation scheme for a two-level page

level scheme then the addresses will look like this:



Thus in order to avoid such a large table, there is a solution and that is to divide the outer page table, and then it will result in a **Three-level page table:**



Hashed Page Tables

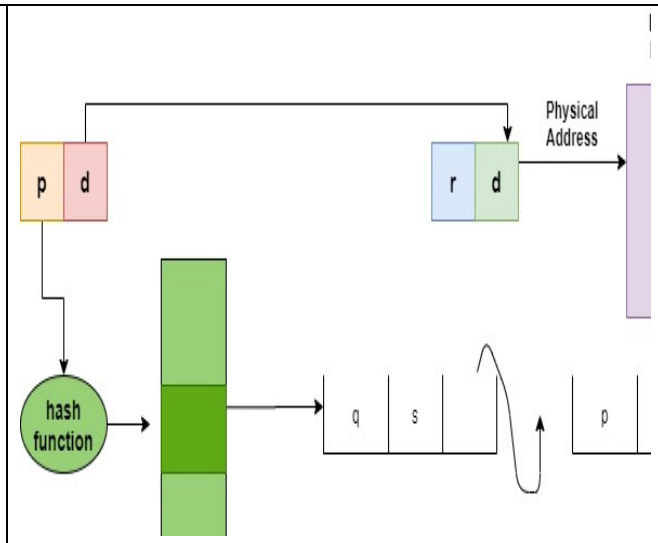
This approach is used to handle address spaces that are larger than 32 bits.

- In this virtual page, the number is hashed into a page table.
- This Page table mainly contains a chain of elements hashing to the same elements.

Each element mainly consists of :

1. The virtual page number
2. The value of the mapped page frame.
3. A pointer to the next element in the linked list.

Given below figure shows the address translation scheme of the Hashed Page Table:

Question Paper Setter

Approved by Audit Professor/ Course Coordinator