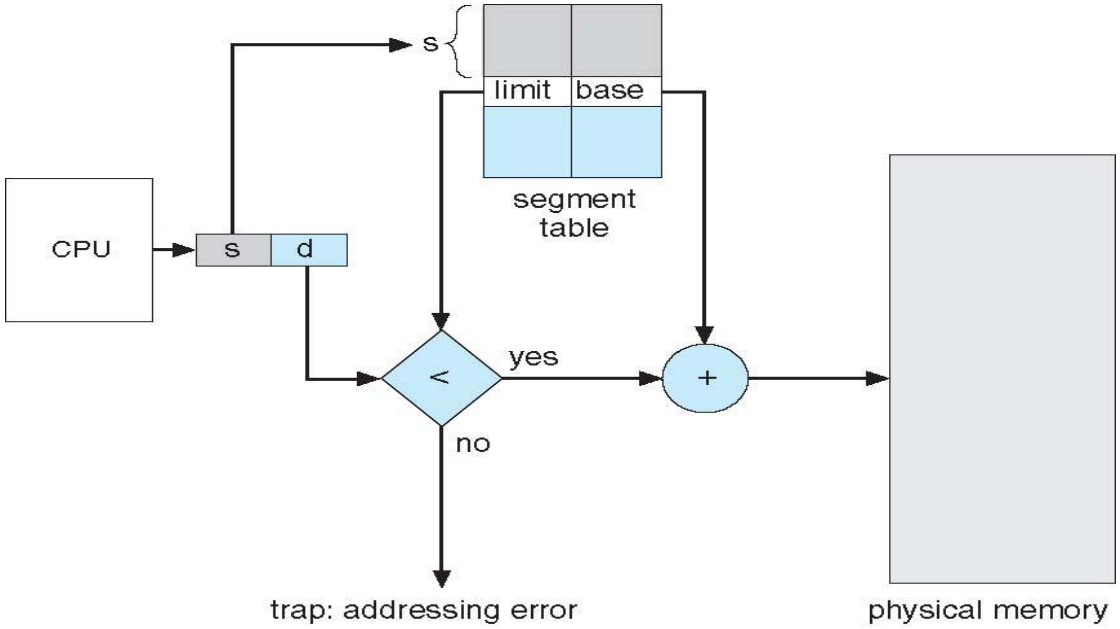
KONGU ENGINEERING COLLEGE, PERUNDURAI 638 060
CONTINUOUS ASSESSMENT TEST – 3
Regulations 2020

Month and Year : Nov 2023	Roll Number :
Programme : B.Tech Branch : IT Semester : V	Date : 15.11.23 Time : 2.30pm to 4pm
Course Code : 20ITT52 Course Name : Operating systems	Duration : 1 ½ Hours Max. Marks : 50

PART - A (10 × 2 = 20 Marks)		
ANSWER ALL THE QUESTIONS		
1.	**External Fragmentation** 1. 212KB is put in 500KB partition. 2. 417KB is put in 600KB partition. 3. 112KB is put in 288KB partition (new partition 288K = 500K - 212K). 4. 426K must wait. **Total External fragmentation** is (100+200+300) = 600KB **Internal Fragmentation** 1. 176 KB in 500KB partition (500-212-112) . 2. 183 KB in 600KB partition(600-417) . **Total Internal fragmentation** is (176+183) = 359KB	[CO4,K3]
2.	The page size is 4KB = 2^{12} bytes. So, the offset is 12. The inner page table size is 4KB = 2^{10}x2^2 bytes. So it needs 10 bits (every entry is of 4 Bytes = 2^2). The first outer page table size is 4MB = 2^{20} x 2^2 bytes. So it needs 20 bits. The second outer page table size is 16MB = 2^{22} x 2^2 bytes. So it needs 22 bits.	[CO4,K3]
3.	Here, a three-level paging scheme is used. Therefore, the EAT is $0.8 \times 100 + (1 - 0.8) \times (4 \times 100) = 160$ns.	[CO4,K3]
4.	The number of entries in the hash table is 17 (0 to 16, since the mod 17 is used). Each entry in the hash page table is the address of a physical page. Since the physical address space is 12 bits, so it need 12-bit entries in the hash page table.	[CO4,K3]
5.	The virtual address is 1101000101111111. Therefore, pid is 1101. The virtual page number is 000101, and the offset is 111111. The pid will give the index in the page table. Here, the pid is 1101, so it indicates the index is 101. The physical address is **<index || offset>,** and so 101111111 is the physical address.	[CO4,K3]
6.	Size of the disk block = 4KB= 4096 bytes = 2^{12} Disk block address = 32bits = 4 bytes Number of addresses per block = Block size/ space occupied by each address = 4096/4 = 1024 = 2^{10} Maximum size of file = (12 direct pointers ×4KB) + (1 single Indirect pointer ×4KB) + (1 Double Indirect pointer ×4KB) =(12×2^{12}) + $(2^{10} \times 2^{12})$ + $(2^{10} \times 2^{10} \times 2^{12})$= $2^{15} + 2^{22} + 2^{32} = 2^{32}$ bytes = $2^2 \times 2^{30}$ = 4GB	[CO5,K3]
7.	▪ **Name** – only information kept in human-readable form ▪ **Identifier** – unique tag (number) identifies file within file system ▪ **Type** – needed for systems that support different types ▪ **Location** – pointer to file location on device ▪ **Size** – current file size ▪ **Protection** – controls who can do reading, writing, executing ▪ **Time, date, and user identification** – data for protection, security, and usage monitoring	[CO5,K1]
8.	In UNIX/Linux operating systems, the 'x' permission allows the execution of a directory.	[CO5,K3]
9.	The free-space bitmap is 00111100111111000110000001110000 ...	[CO5,K3]

10.	FCB contain many details about the file such as ▪ File permissions ▪ Ownership ▪ Size ▪ Location of data block **File permissions** **File dates (create, access, write)** **File owner, ACL** **File size** **File data blocks**		[CO5,K1]
	PART – B (3 × 10 = 30 Marks) ANSWER ANY THREE QUESTIONS		
11.	Consider the following page reference string: 1, 5,2, 3, 4, 5,1, 2, 5, 1, 2, 3, 4, 5,4 i) When frame size is three, the number of page faults are FIFO – 11, Optimal- 8, LRU-11 ii) When frame size is four, the number of page faults are FIFO – 10, Optimal- 6, LRU-10 No, this example does not exhibit Belady's anomaly.	(10)	[CO4,K3]
12.	**i) Segmentation architecture:** ▪ Memory-management scheme that supports user view of memory ▪ Segmentation is another non-contiguous memory allocation like paging ▪ Unlike paging, in segmentation , the processes are not divided into fixed size pages ▪ Instead , the processes are divided into several modules called segments • A segment is a logical unit such as:main program, procedure, function, method, object, local variables, global variables, common block, stack , symbol table, arrays etc So, both secondary memory and main memory are divided into partitions of unequal sizes.  		

- Protection
  - With each entry in segment table associate:
    - validation bit = 0 ⇒ illegal segment
    - read/write privileges
- Protection bits associated with segments
- Since segments vary in length, memory allocation is a dynamic storage-allocation problem

(5) [CO4,K2]

ii) There are four ways to implement LRU ;

1. Using Counters 2. Using Stack 3. Additional Reference Bits Algorithm
4. Second-chance algorithm

- **Counter implementation**

With each page table entry, associate a time of use field and add logical clock or counter to the CPU. The clock/counter is incremented for every memory reference. Whenever a reference to a page is made, the contents of the clock/counter are copied to the time of use field in the page table entry for that page. So, always have the 'time/count' value of the last reference to each page.

When a page needs to be changed, look at the counters to find smallest value- Search through table needed

Stack implementation

- Maintain a stack of Page Numbers
- Whenever a page is referenced, it is removed from the stack and put on the top
- So, the most recently used page is always at the top of the stack and the least recently used page is always at the bottom
- Since, entries must be removed from the middle of the stack, it is best to implement by using a doubly linked list with head and tail pointer

Additional Reference Bits Algorithm

- Additional ordering information by recording the reference bits at regular intervals.
- Keep an 8-bit byte for each page in a table in memory.
- At regular intervals (say, every 100 milliseconds), a timer interrupt transfers control to the operating system.
- The operating system shifts the reference bit for each page into the high-order bit of its 8-bit byte, shifting the other bits right by 1 bit and discarding the low-order bit.
- These 8-bit shift registers contain the history of page use for the last eight time periods

(5)

Page No	Shift Register Content	What it means
P1	00000000	This page has not been used for eight time periods
P2	11111111	The page has been used atleast once in each period
P3	11000100	The page has been used more recently than P4
P4	01110111	The page has been used less recently than P3

Second-chance algorithm

- Basic algorithm of second-chance replacement is a FIFO replacement algorithm.
- When a page has been selected, check its reference bit. If the value is 0, proceed to

	replace this page; ➢ But if the reference bit is set to 1, give the page a second chance and move on to select the next FIFO page. ➢ When a page gets a second chance, its reference bit is cleared, and its arrival time is reset to the current time. ➢ Thus, a page that is given a second chance will not be replaced until all other pages have been replaced (or given second chances). ➢ In addition, if a page is used often enough to keep its reference bit set, it will never be replaced.	(10)	[CO5,K3]
13	**a. FCFS service order : 7081 cylinders** **b. SSTF service order : 1745 cylinders** **c. SCAN service order : 9769 cylinders** **d. LOOK service order : 3319 cylinders** **e. C-SCAN service order : 9985 cylinders** **f. C-LOOK service order : 3363 cylinders**	(10)	[CO5,K1]
14	File allocation : An allocation method refers to how disk blocks are allocated for files **Various allocation Methods:** • Contiguous Disk space allocation • Linked Disk space allocation • File Allocation Table (FAT) • Indexed Allocation Method **Contiguous Allocation Method:** Each file occupies set of contiguous blocks on the disk. ▪ Contiguous allocation of a file is defined by the disk address and length(in block units) [Eg] – If a file is 'n' blocks and starts at location 'b' , then it occupies the blocks b,b+1,b+2,---- b+n-1. ▪ The directory entry for each file indicates the address of the starting block and length of the area allocated for this file. Advantages: Best performance in most cases • Simple – only starting location (block #) and length (number of blocks) are required For Sequential Access: The file system remembers the disk address of the last block referenced. Then it reads the next block. For Direct Access: For accessing block 'i' of a file that starts at block 'b'. Then immediately access block b + i block **Linked Allocation:** Each file is a linked list of blocks which may be scattered anywhere on the disk ▪ The directory contains a pointer to the first and last blocks of the file ▪ Each block contains a pointer to the next block and File ends at nil pointer . If each block is 512 bytes in size, and a disk address(the pointer) requires 4 bytes, then the user see the block of 508 bytes Advantages: ▪ No external fragmentation and No compaction ▪ Each block contains pointer to next block ▪ Free space management system called when new block needed		

14.	▪ Improve efficiency by clustering blocks into groups Dis Advantages: 1. It can be used effectively only for sequential access not for direct access. To find the i^{th} block , start at beginning of that file and follow the pointers until to get the i^{th} block . 2. Space required for the pointers - If a pointer requires 4bytes out of a 512 byte block, then 0.78 percent of the disk is used for pointers 3. Reliability problem - If the pointers are used for linking the files gets damaged or lost or wrong pointers are picked up leads to problems **FAT Allocation Method:** This method is used by MS-DOS ▪ Beginning of volume has table, the table has one entry for each disk block and indexed by block number ▪ The directory entry contains the block number of the first block of the file ▪ The table entry indexed by that block number contains the block number of the next block in the file. The chain continues until the last block, which has a special end –of- file value as the table entry Unused blocks are indicated by a '0' table value **Advantage : Random access time is improved** **Dis Advantage : Not suitable for Direct access**		
	Indexed Allocation Method: The main disadvantage of linked disk allocation is, it can not support for direct access. ▪ Indexed allocation method solves this problem by bringing all the pointers together into one location called as index block ▪ Each file has its own index block- an array of disk block addresses. The i^{th} entry in the index block points to the i^{th} block of the file ▪ The directory contains the address of the index block ▪ To find and read the i^{th} block, use the pointer in the i^{th} index block entry Create a new file : Step1: All pointers in the index block are set to null. Step2 : When the i^{th} block is first written, a block is obtained from the free space manager. Step3 : Its address is put in the i^{th} index block entry **Advantages:** 1. Support direct access, without suffering external fragmentation 2. Any free block on the disk can satisfy a request for more space **DisAdvantages:** 1. More wasted space 2. Pointer overhead of the index block is generally greater than the pointer overhead of linked allocation		

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	20	20	60	-	-	-