



(8) Explain the following briefly

(8.a) Thrashing (**4pts**)

(8.b) Belady's anomaly (**4pts**)

(8.c) Seek time (**4pts**)

(8.d) Inverted page table (**4pts**)

(9) Consider the following snapshot of a system and answer the question using the banker's algorithm

	<u>Allocation</u>				<u>Max</u>				<u>Available</u>			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

(9.a) What is the content of the matrix *Need*? (5pts)

(9.b) Is the system in a safe state? Explain why. (5pts)

(9.c) If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately? Explain why. (5pts)

(10) Suppose that we have a paged virtual memory where the virtual address is 32 bits and the physical address is 24 bits. The lower 12 bits of each address provides the page offset and the memory is byte addressable.

(10.a) What is the page size? (3pts)

(10.b) What is the page table size if we use a one-level page table? Note that each page table entry has the starting address of the corresponding physical page frame. (3pts)

(10.c) The page table is stored in the main memory and no page fault happens for page table reference. It takes 10 ms to service a page fault if an empty frame is available or if the replaced page is not modified, and 20 ms if the replaced page is modified. Page fault ratio is 1% and the page to be replaced is modified 50% of the time. The access to read from or write to the main memory is 100ns. What is the effective access time? (5pts)

(11) In a file allocation scheme, there are several ways that disk blocks can be allocated to a file. We could have: (i) a linked allocation where the last word of a block is a disk address of another block, and (ii) a indexed allocation where the first block, called index block, has a set of pointers to the other blocks for data. Assume that a disk block is 10 bytes and a word, a disk address, is 1 byte.

(11.a) If you want to access the 40th byte in a file, how many disk accesses are required for linked allocation and indexed allocation respectively? Also, explain why. **(4pts)**

(11.b) A variation of linked allocation is a File Allocation Table (FAT) used by MS-DOS operating system, where a section of disk at the beginning of each partition is set aside to contain the table about linked list. Assume that FAT is initially loaded into main memory. What would be the advantage of FAT over linked allocation? For accessing 30th byte in a file, how many disk accesses are required? Also explain why. **(4pts)**

(11.c) Unix operating system uses i-node that has direct blocks, single indirect blocks, double indirect blocks, and triple indirect blocks. Explain how this works briefly and calculate the maximum file size using i-nodes. **(6pts)**

(12) Two basic I/O schemes for interaction between CPU and the controllers are polling and interrupts. Explain and compare them briefly. **(5pts)**

(13) Suppose that there is a disk that has 200 cylinders. Consider a disk queue with requests for I/O to disk blocks on cylinders 100, 180, 40, 120, 10, 120, 60, 70. Currently disk head is at cylinder 50. What is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk scheduling algorithms? Show how you get the answer.

FCFS (First Come First Served) (2 pts)

SSTF (Shortest Seek Time First) (2 pts)

C-LOOK (Circular LOOK) (2 pts)



(14.b) Although LRU page replacement algorithm is theoretically realizable, it required special hardware to fully implement LRU to maintain the linked list of the least recently used pages and to update the list for every memory reference. LRU can be implemented in Software that approximates LRU replacement. Suppose that the system provides 8 reference bits for each page. Explain how we can use these reference bits to approximate LRU algorithm. (5pts)