National University of Computer and Emerging Sciences, Lahore Campus

Course:	Operating Systems	Course Code:	CS205
Program:	BS (Computer Science)	Semester:	Spring 2019
Due Date:	23-4-2019	Total Marks:	5
Section:	D	Weight	5
Exam:	Quiz 3	Page(s):	1
Name:		Roll #:	

1. Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference, if the page-table entry is in the associative memory. This implies that the overhead of associative memory is negligible and can be ignored. Assume that 80 percent of the accesses are in the associative memory. Out of the remaining accesses (20%), 10% (or 2% of the total) cause page faults. What is the effective memory access time?

Answer:

HIT not HIT but in mem page fault

EAT = (0.8 x 1 µs) + (0.18 x 2 µs) + (0.02 x 20002 µs)
= 0.8 + 0.36 + 400.04
= 401.2 µs

2. A computer with a 32-bit address uses a two level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-level page table field, and an offset. How large are the pages and how many are there in the virtual address space ?

- How large are the pages?	4KB
- How many pages are there in the virtual address space?	2^20 pages.

3. Consider the two-dimensional array A: int A[][] = new int[100][100]; where A[0][0] is at location 200, in a paged memory system with pages of size 200 bytes. Each int type needs 4 bytes and A is stored in row-major order (as in C/C++). A small process is in page 0 (locations 0 to 199) for manipulating the array; thus, every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array-initialization loops, using LRU replacement, and assuming frame 0 has the process in it and the other two frames are initially empty and later used to store the array A?

- for (int j=0; j < 100; j++) for (int i=0; i< 100; i++) A[i][j]=0;	100x100
- for (int i=0; i< 100; i++) for (int j=0; j < 100; j++) A[i][j]=0;	200