Assigment-9

1.

Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.)

• TLB miss with no page fault

• TLB miss and page fault

• TLB hit and no page fault

• TLB hit and page fault

A:

1. page has been removed from the TLB but is still in memory, possible memory contents that are not frequently used

2. A page fault has occurred

3. Frequently used memory content, both in memory and has indexed in the TLB

4. No such situation exists

2.

Assume that we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory-access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

A: (time units are in microseconds)

(1 - P) \* 0.1 + o.3P \* 8000 + 0.7P \* 20000 < 0.2

16400.1P < 0.1

P < 1/164001 = 6E-6

Therefore, the maximum page fault rate P is 6E-6.

3.

When a page fault occurs, the process requesting the page must block while waiting for the page to be brought from disk into physical memory. Assume that there exists a process with five user-level threads and that the mapping of user threads to kernel threads is one to one. If one user thread incurs a page fault while accessing its stack, would the other user threads belonging to the same process also be affected by the page fault—that is, would they also have to wait for the faulting page to be brought into memory? Explain.

A: No, because the stack is independent between threads and does not affect each other.

4.

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. Assume that 80 percent of the accesses are in the associative memory and that, of those remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time?

A:

E(T) = 80% \* 1 + 18% \* 2 + 2% \* (2 + 20000)

= 401.2 ms