1. Give estimates for the following values:

(a) cache access time \_\_\_1 ns\_\_\_

(b) main memory access time \_\_\_100 ns\_\_\_\_

(c) disk access time \_\_\_10 ms\_\_\_\_

2. Label the following steps required in processing a page fault in their proper sequential order, 1-4.

\_4\_\_ OS eventually dispatches to process 1. Load instruction in process 1 is retried and succeeds.

\_3\_\_ I/O completion interrupt for the page-in; OS sets the presence bit in the PTE; and, OS changes process 1

back to ready.

\_1\_\_ Load instruction in process 1 has a page fault. CPU saves information about the fault and context switches

to an OS fault handler. OS checks the PTE to see if requested access is permitted. Since access is allowed,

OS does not terminate process 1.

\_2\_\_ OS finds a free page frame or selects which page frame should be replaced, and schedules an I/O operation

to bring in the missing page from disk to the selected page frame. OS suspends process 1. OS dispatches to

another ready process (e.g., process 2).

3. Fill in the three empty cells in the table below with the appropriate bit values for the on-demand clock algorithm when only the use bit is considered.

|  |  |  |
| --- | --- | --- |
| **Current Use Bit** | **Action Taken When a Replacement is Needed** | **New Use Bit Value** |
| 0 | Replace this page, move pointer forward so next search starts one page beyond the new page | 0 |
| 1 | Skip this page for now, move pointer forward and continue search for replacement | 0 |

4. Explain why a presence bit in a page table could be off but the page still be in physical memory in VAX/VMS. (see slides)

The page is in the victim buffer and ready for fast reclaim if there is any reuse.

5. How does page fault frequency (PFF) attempt to prevent page thrashing? (see slides)

Tracks fault rate for all processes and attempts to steal pages from low-fault-rate processes to high-fault-rate processes.