CMPT 300: Assignment #4
Virtual Memory Questions

Authors: Van Khai Nguyen (301538378), Duc Manh To (301425895)

Question 1:

- A certain computer provides its users with a virtual-memory space of 2^{32} bytes
- The computer has 2^{18} bytes of physical memory
- Virtual memory is implemented by paging, and the page size is 4096 bytes = 2^{12} bytes

A user process generates the virtual address 11123456 (hexadecimal).
Explain how the system establishes the corresponding physical location.
Distinguish between software and hardware operations.

Convert the hexadecimal virtual address to binary:
0x11123456 ⇒ 0001 0001 0001 0010 0011 0100 0101 0110

- **Page number** (upper 32 - 12 = 20 bits of the virtual address): 0001 0001 0001 0010 0011
- **Offset** (Least significant 12 bits): 0100 0101 0110

Mapping Virtual Pages to Physical Frames
The mapping from virtual pages to physical frames is managed by the page table. The page table stores the frame number for each page, if it present in physical memory
Software Operations:

- **Page Table Lookup**: The OS uses the page number as an index into the page table to find the corresponding frame number. If the page is not in memory (page fault), the OS must load it from disk → a physical frame, and updating the page table

Hardware Operations:

- **Translation Lookaside Buffer (TLB)**: A cache that stores recent translations of virtual page numbers to physical frame numbers to speed up the translation process
- **Physical Address Calculation**: Once the frame number is obtained (either from the TLB cache hit or a page table lookup following a TLB miss), the physical address is compute by concatenating the frame number and the offset

For example, assuming the frame number for the virtual page 0x11123 is 0x0001 (in reality this would be looked up from the page table or loaded into memory on a TLB miss):

- **Physical Address**: Frame number 0x0001 with offset 0x456
- Convert frame number and offset to binary and concatenate:
    - Frame 0x0001 ⇒ 0000 0000 0000 0001
      ⇒ Physical Address: 0000 0000 0000 0001 0100 0101 0110

Question 2: Assume 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 page is available or if the replaced page is not modified, but 20 milliseconds if the replaced page was modified.
- Memory access time is 100 nanoseconds.
- Assume that the page to be replaced is modified 70% of the time.

What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

- Effective Access Time(EAT) = (1 - p) x Memory Access Time + p x Page Fault Time
- Page Fault Time = (1 - 0.7) x 8 + 0.7 x 20 = 16.4 milliseconds = 16,400,000 nanoseconds
  => 200 = 100 + p(16,400,000 - 100)
  P = 100 / 16,399,900 ≈ 6.097 x 10^(-6)

So the maximum acceptable page-fault rate to achieve an EAT of no more than 0.0006097%