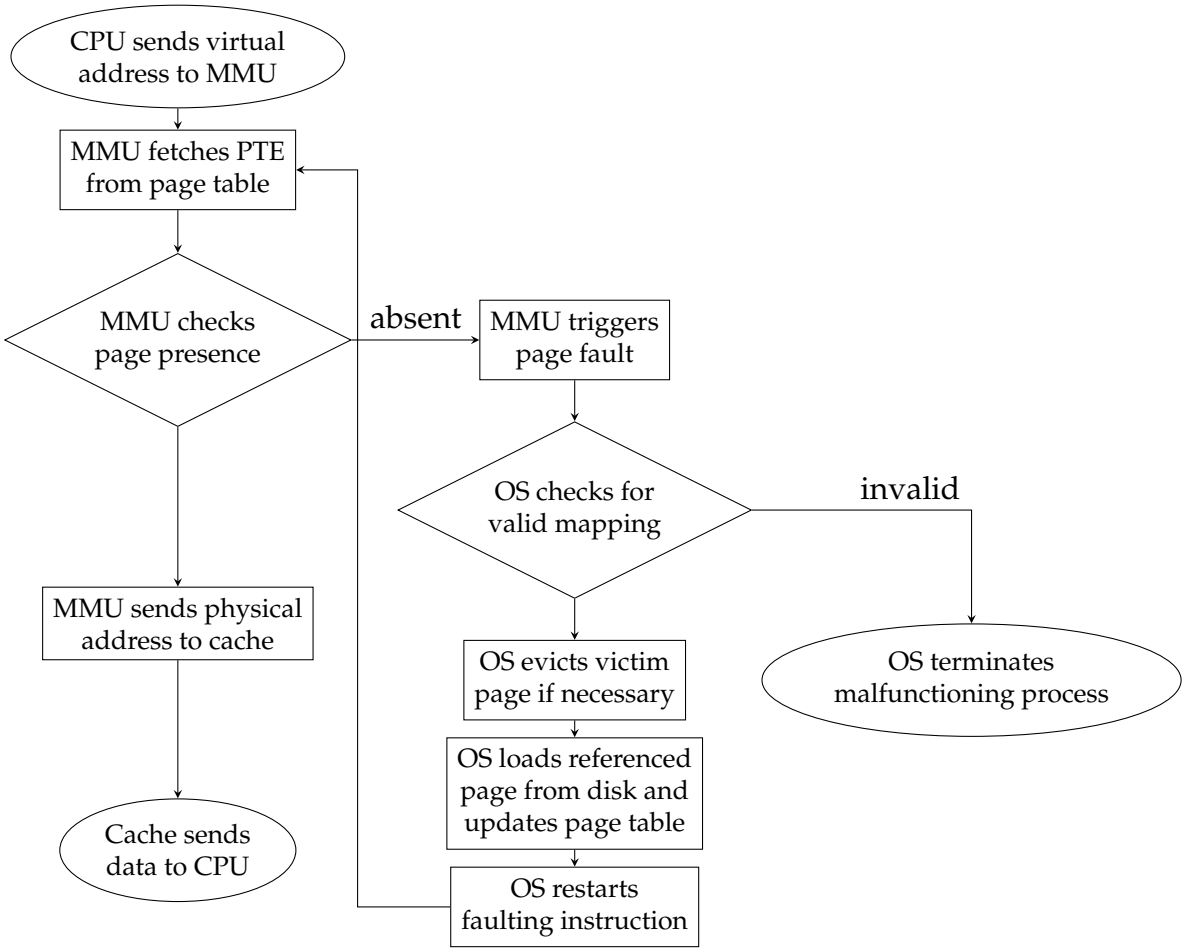
Introduction

This activity will help you understand the details of address translation and memory mapping in a virtual memory system.

1. Address Translation Concepts

Problems: 7 / Allocated Time: 10 minutes

This flowchart illustrates the steps required to read a value from memory on a system with virtual memory.



The problems in this section discuss the details of these steps.

Problem 1. For a simple system with a one-level page table, what sub-steps does the MMU take when it fetches a PTE from a page table? (Hint: think about how to split up the address in order to look for an entry in the page table.)

Problem 2. The MMU must know the *physical* address of the page table in order to read page table entries from memory. Why does it need a physical address?

Problem 3. Why are one-level page tables impractical? How do multi-level page tables fix this problem?

Problem 4. Why might an MMU take longer to look up addresses that *are* mapped in a multi-level page table than the equivalent one-level page table?

Problem 5. What is the Translation Lookaside Buffer (TLB) and what problem is it intended to solve? At what point in the process of fetching PTEs is the TLB used?

Problem 6. When does a page fault happen? What does the page fault handler do

when it is called?

Problem 7. How does virtual memory interact with the memory cache(s)?

2. TLB Practice

Questions: 1 / Allocated Time: 7 minutes

Problem 8. Assume a system with a two-way set-associative TLB, with a total of eight entries. Virtual and physical addresses are both 16 bits, and pages are 2^8 bytes each.

At some point in a program's execution, the contents of the TLB are as follows:

Index	Tag	PPN	Valid
0	`0x13`	`0x30`	1
0	`0x34`	`0x58`	0
1	`0x1F`	`0x80`	0
1	`0x2A`	`0x72`	1
2	`0x1F`	`0x95`	1
2	`0x20`	`0xAA`	0
3	`0x3F`	`0x20`	1
3	`0x3E`	`0xFF`	0

Based on the contents of the TLB, fill in the following table of virtual to physical address mappings. If you don't have enough information to fill a cell, write "?" in that cell.

Virtual address	Physical address
0x7E85	
0xD301	
	0x3020
0xD040	
	0x5830

3. Memory Mapping

Questions: 2 / Allocated Time: 7 minutes

Problem 9. Describe two different things that can provide the "backing store" for a region of virtual memory, and explain what the page fault handler does the *first time* a page of memory with those types of backing store is accessed.

Problem 10. Each process has its own, private virtual address space. However, some parts of the address space might be *shared* with other processes—that is, mapped to the same physical addresses in all the processes that share that region. Give an example of when this happens and explain why it might improve overall system performance.

A. Appendix: Address Translation Symbol Reference

- Basic parameters
  - $N = 2^n$: Number of addresses in virtual address space
  - $M = 2^m$: Number of addresses in physical address space
  - $P = 2^p$: Page size (bytes)
- Components of the virtual address (VA)
  - VPO: Virtual page offset
  - VPN: Virtual page number
  - TLBI: TLB index
  - TLBT: TLB tag
- Components of the physical address (PA)
  - PPO: Physical page offset (same as VPO)
  - PPN: Physical page number
  - CO: Byte offset within cache line
  - CI: Cache index
  - CT: Cache tag