Quiz: Address Translation

Total points 40/52

Take the quiz solo, but feel free to consult a partner student, the book, the videos or other resources if needed. Re-take quiz if your score is less than 80% or if you just want some more practice.

The respondent's email (**faiyaz@pdx.edu**) was recorded on submission of this form.

Will the OS access a process's page table at each of these moments? *

	Yes	No	Maybe	Score	
at process creation time	◉	○	○	1/1	✓
when the process calls a system call	○	◉	○	0/1	✗
at process termination time	○	◉	○	0/1	✗
on TLB miss (software managed TLB)	◉	○	○	1/1	✓
on TLB hit (i.e., during normal address translation)	○	◉	○	1/1	✓
when the process allocates/deallocates memory	◉	○	○	0/1	✗
on TLB miss (hardware managed TLB)	○	◉	○	0/1	✗

✓ Caches (like the TLB) are said to have three types of misses (the three Cs): Compulsory misses when addresses are first encountered, Capacity misses when more addresses are accessed than the cache can contain at once, and Conflict misses when multiple addresses map to the same cache location. Which of these three miss types do NOT occur in a TLB? *1/1

- ○ Capacity
- ◉ Conflict ✓
- ○ Compulsory

✗ A given memory page for a process always maps to the same physical page frame. *0/5

- ◉ True ✗
- ○ False

✓ if we have a 32 bit address space and our page size is 4096 bytes, then how many bits of each 32 bit address are used to determine the byte offset within the page? *5/5

- ◉ 12 ✓
- ○ 32
- ○ 24
- ○ 4096
- ○ 20

✓ if we have a 32 bit address space and our page size is 4096 bytes, then what is the maximum number of page frames in physical memory? *5/5

- (●) 2^20 ✓
- ( ) 2^24
- ( ) 2^32
- ( ) 2^12

✓ if we have a 32 bit address space and our page size is 4096 bytes, then how many page addresses will be kept in the TLB? *5/5

- ( ) 4096
- (●) unknown. the TLB is just a cache and therefore can be any size. ✓
- ( ) 2^20
- ( ) 64

Feedback

The size of the TLB will not depend on the page table size, the size of memory, or really much of anything other than the amount of circuits that the hardware designers decided to dedicate to the TLB. It is a cache, and caches can really be of any size. Of course, larger TLBs generally give better system performance, but they also cost more money per chip. The hardware/system designers generally size them by simulating a set of well-established benchmarks and measuring system performance (TLB hit ration among other metrics) for each simulated configuration of the system.

✓ if we have a 32 bit address space and each page table entry is 24 bits wide (20 for the page frame number and 4 for the management bits) then how much memory is required to store each page table (assuming a single-level linear page table)? *5/5

- ○ 2^20. bytes
- ○ 2^12 bytes
- ◉ 2^20 * (24 / 8) bytes. ✓
- ○ 2^24 bytes

Feedback

 # pages in the system = 2^20
 # entries in page table = one for every page or 2^20 entries
size of each page table entry = 24 bits = 24/8 bytes = 3 bytes
*memory needed to hold page table = 2^20 * (24/8) bytes ~= 3.1M bytes*

✓ Why/how do multi-level page tables save space in memory? * 5/5

- ○ second level page table entries do not need to be mapped into the TLB
- ○ some intermediate page table entries are empty and not needed
- ○ not all pages within a virtual address space are allocated
- ◉ avoids maintaining page table information for unused pages ✓
- ○ second level page table entries can be omitted for pages that are not in use.

Which of the following are reasons why a multi-level page table saves memory space. *

	yes, valid reason why multi-level page tables save memory space	no, not a valid reason why multi-level page tables save memory space	Score	
second level page table entries can be omitted for pages that are not in use.	○	◉	0/1	✗
avoids maintaining page table information for unused pages	◉	○	1/1	✓
not all pages within a virtual address space are allocated	○	◉	0/1	✗
some intermediate page table entries are empty and not needed	○	◉	0/1	✗
second level page table entries do not need to be mapped into the TLB	○	◉	1/1	✓

✓ To save memory, could the OS page out unused page table space? * 5/5

- (●) yes, we could, but it slows access to memory and thereby reduces speed of applications. ✓
- ( ) depends on how much external fragmentation the system chooses to tolerate
- ( ) no can't page a page table
- ( ) yes, and this is a superior technique

What is the meaning of each of the administrative bits in a page table entry? *

	D bit	R bit	W bit	V bit	Score	
indicates that the page is valid, loaded into memory	○	○	○	◉	1/1	✓
indicates that the page is writeable	○	○	◉	○	1/1	✓
indicates that the page has been referenced (read or written) recently	○	◉	○	○	1/1	✓
indicates that the page has been been modified recently and the modifications have not yet been written to swap space.	◉	○	○	○	1/1	✓