0/9 Questions Answered

Check-in Quiz 03 Virtual Memory

Q1
3 Points

Consider a system with the following configuration:

- 64 bit address space (one address is 64 bits)
- 16 GiB physical memory
- 16 bit (2 bytes) addressability
- 4 KiB page size

Note that we have an addressability of 16 bits (2 bytes), in the VM lectures, we usually worked with 8-bit addresability. This means that each address corresponded to 16 bits of memory, meaning that instead of each address corresponding to a single byte, it instead corresponds to two bytes.

As a reminder:
1KiB = 2^10 bytes
1GiB = 2^30 bytes
(2^10 is 2 to the power of 10 or 1024)

Q1.1
1 Point

How many bits are needed to represent the page offset of an address? (e.g. how many bits does it to take to represent the number of possible addresses that correspond to a single page)

Please answer with a number only:

Save Answer

Q1.2
1 Point

How many bits are needed to represent the virtual page number of an address?

Please answer with a number only:

Save Answer

Q1.3
1 Point

How many bits does it take to represent a physical page?

Please answer with a number only:

Save Answer

Q2 LRU
2 Points

Imagine we are working with a system that has 4 physical pages and that we are u sing LRU as the replacement policy.

For the following reference string, what is the number of page faults?

0 1 2 4 5 0 2 2 3 5 0 3 4 1 5 2

Save Answer

Q3 Page Tables
5 Points

For this problem, let us work with an architecture that supports

- 16-bit addresses
- 1-byte addressability
- pages that are 1 KiB
- 4 physical page frames

Assume that on this architecture, we access the following virtual pages:

- 0x36 (11 01 10)
- 0x1B (01 10 11)
- 0x0A (00 10 10)
- 0x0C (00 11 00)
- 0x1A (01 10 10)
- 0x0B (00 10 11)

Assume that physical memory starts empty.

As a reminder:
1KiB = 2^10 bytes
(2^10 is 2 to the power of 10 or 1024)

Q3.1 Array
1 Point

How many entries would exist in the page table if it was a "big array" page table

Save Answer

Q3.2 Inverted
1 Point

How many entries would exist if it used an inverted page table?

Save Answer

Q3.3 Multi-level
1 Point

What if the architecture used a 3-level page table that had 2 bits to index into each level of the page table. (Note in lecture we had a 4-level page table with 9 bits to index into each level)

(Repeating the for convenience)
we access the following virtual pages:

- 0x36 (11 01 10)
- 0x1B (01 10 11)
- 0x0A (00 10 10)
- 0x0C (00 11 00)
- 0x1A (01 10 10)
- 0x0B (00 10 11)

How many Page Table Entries would exist in the page table (after accessing the addresses listed above)
(You can also think of this as how many valid entries are there in the bottom level of the page table).

Save Answer

Q3.4 Multi Level Page Table: Top Level
1 Point

After accessing the specified addresses, how many entries would be

non-`NULL` in the top level of the page table? (Same page table as the one described in 3.3)

Save Answer

Q3.5 Multi Level Page Table: Mid Level
1 Point

After accessing the specified addresses, how many entries would be non-`NULL` in the second-highest level of the page table? (Same page table as the one described in 3.3)

Save Answer

Save All Answers

Submit & View Submission ›