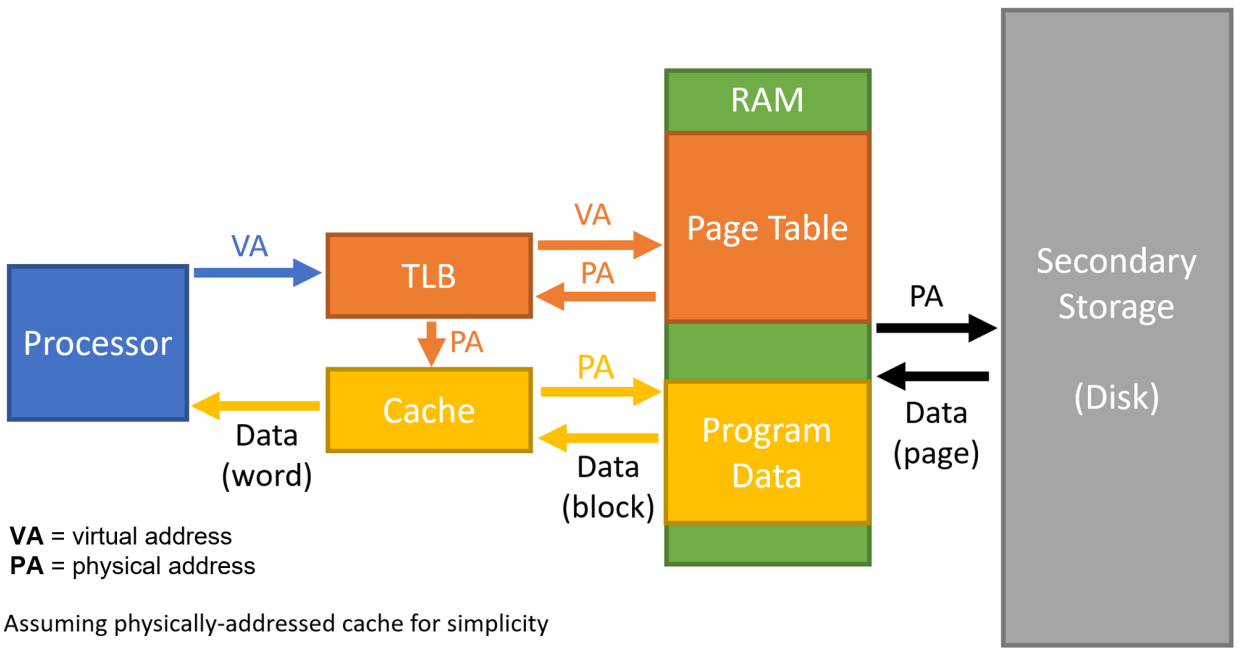
HW9.5. Resolving a Page Fault

A page fault occurs when the data that the program requests is not present in the main memory. For this question, we will try to enumerate the steps (in order) when resolving a page fault.

As a recap, the memory hierarchy would look like the following:



Arrange the following steps in the correct order.

Follow the arrows shown in the figure. We start from the processor side.

Drag from here:

- We load a page of data from the secondary storage (Disk) to the main memory (RAM).
- The virtual address to physical address translation is loaded to the TLB. We now have the physical address of the data we need.
- Since the TLB misses, we access the page table in the main memory (RAM) for the translation. Valid bit is 0. Data is on the secondary storage (Disk). Page fault.
- Page table is updated in the main memory (RAM) to point to the loaded page from the secondary storage (Disk). There's a valid physical address translation now.
- Given the physical address, we access the data cache. Since the page was just loaded earlier, it does not exist in the cache. Cache misses.
- Since the cache misses, we load the block of data from the main memory (RAM).
- Given the virtual address, we access the TLB to look for the physical address translation.

Construct your solution here:

Homework 9

Assessment overview

Total points: 0/100

Score: 0%

Question

Value: 15

History:

Awarded points: 0/15

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

No translation exists for that address. TLB misses.

The cache can now return the corresponding data word to the processor.

Save & Grade *20 attempts left*

Save only

Additional attempts available with new variants