ECE 424 day38

Table of Contents

1. Virtual machines

What are "**privileged instructions**"?

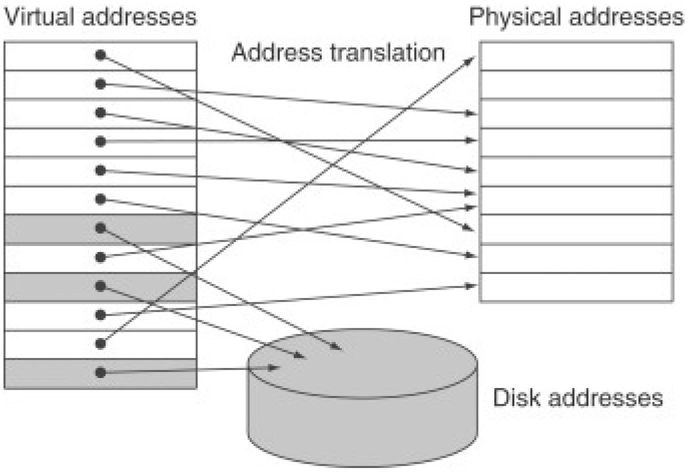
2. Virtual memory

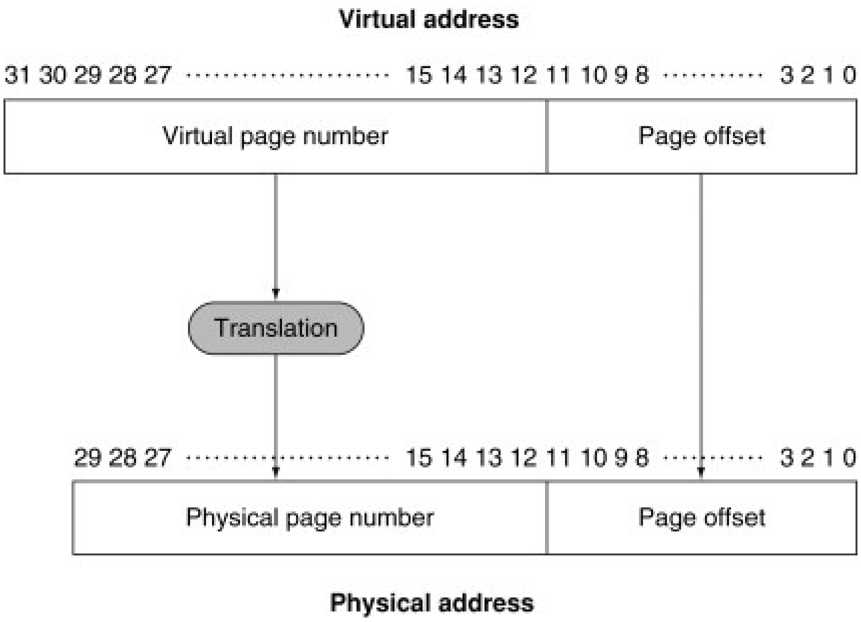
Treats main memory as a(nother) cache that translates to physical storage (SSD, HDD, etc.).

- Virtual addresses between **processes** can overlap
- Physical addresses correspond to installed main memory

Virtual memory is split into "cache blocks" called **pages** which are the smallest unit of translation between disk and main memory. A page can correspond to a (portion of a) file on disk, or general purpose scratch space for variables (RAM).

These RAM pages can be stored on disk if the portion of main memory is needed for something else. This storage is called **swap space** and is usually the same size or larger than main memory.



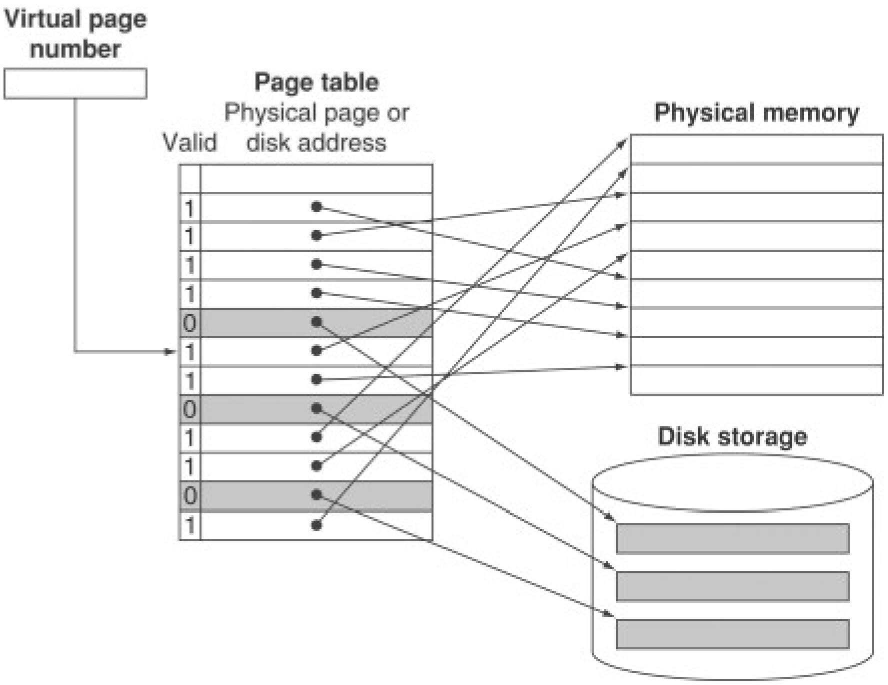


A **Page fault** is when the program asks for a **virtual memory** address, found in a certain page, that is not currently stored in main memory. The system then needs to seek to and transfer the page contents to a free page.

Decisions:

- Which physical address block to use
- If full, which page to **swap** to disk to make space for the new page.

Writing back pages uses a *dirty bit* to ID whether this is necessary.



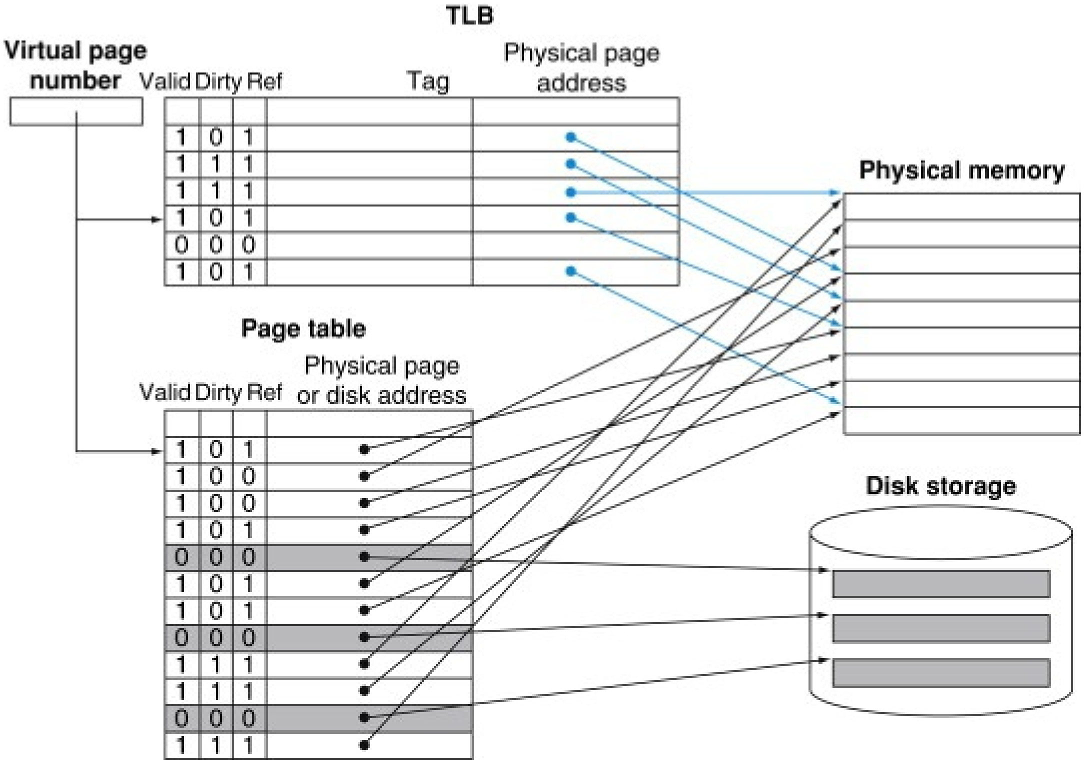
2.1. Virtual memory, virtual machine, and state

The operating system needs to keep the following information for each process:

- Page table
- **Page table register** -> pointer to the current page table.
- Program counter
- CPU registers

Switching between processes then requires saving and restoring this state information.

2.2. Translation-lookaside buffer (TLB)



Since the page table has an entry for every virtual page, no tag field is needed; in other words, unlike a TLB, a page table is *not* a cache.

Last updated 2021-11-29 14:50:40 UTC