Operating Systems

(Memory management: virtual memory, demand paging, performance)
Slides Set #17

By Prof K R Chowdhary

JNV University

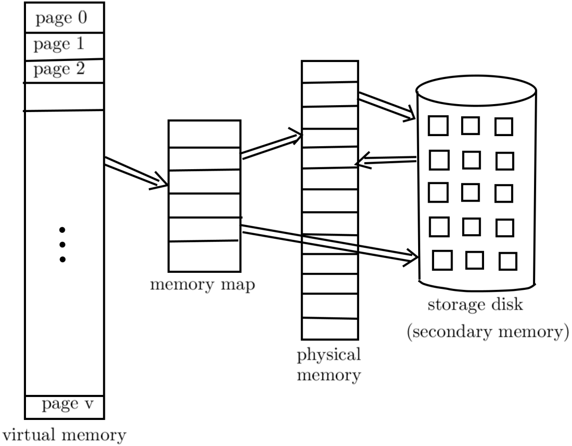
January 10, 2024

Virtual Memory

Virtual memory is a technique that allows the execution of **processes** that are not completely in memory.

- The ability to execute a program that is only partially in memory would give many benefits:
  - A program would no longer be limited by the size of physical memory
  - Because each user program could take less physical memory, more programs could be run at the same time,
  - Less I/O would be needed to load or swap user programs into memory,
- Virtual memory involves the separation of logical memory as perceived by users from physical memory.

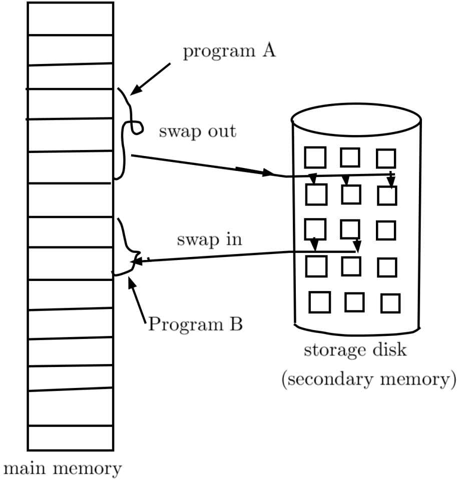
Virtual Memory...



The virtual address space of a process refers to the logical (or virtual) view

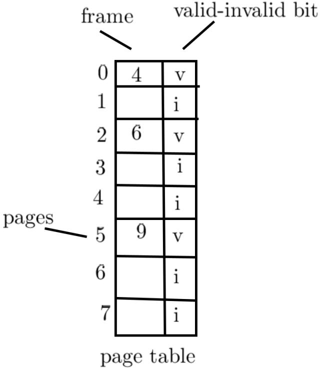
Demand Paging

- Consider how an executable program might be loaded from disk into memory. Loading the entire program into memory results in loading the executable code for all options,
- A demand-paging system is similar to a paging system with swapping



Demand Paging: Concepts

- When a process is to be swapped in, the **pager** (i.e., program that loads the page in RAM) guesses which pages will be used before the process is swapped out again.
- With this scheme, we need some form of hardware support to distinguish between the pages that are in memory and the pages that are on the disk.
- Access to a page marked invalid causes a **page fault**, this initiates loading of page from disk to RAM in a vacant frame.



Demand Paging: ....

- In the extreme case, we can start executing a process with no pages in memory.This scheme is **pure demand paging**: never bring a page into memory until it is required.
- Theoretically, some programs could access several new pages of memory with each instruction execution. Programs tend to have **locality of reference** (What is it?)
- The hardware to support demand paging is the same as the hardware for paging and swapping:
  - **Page table.**
  - **Secondary memory**.
- A crucial requirement for demand paging is the ability to restart any instruction after a page fault.

Performance of Demand Paging

Demand paging can significantly affect the **performance** of a computer system. To see why, let's compute the effective access time (ma)) for a demand-paged memory.

Let p be the probability of a page fault ($0 \leq p \leq 1$). We would expect p to be close to zero – that is, we would expect to have only a few page faults. The effective access time is then $(1 - p) \times ma + p \times$page fault service time.

With an average page-fault service time of 8 milliseconds and a memory-access time of 200 nanoseconds, the effective access time in nanoseconds is:

$$
\begin{aligned}
&=(1 - p)(200) + p \times (8 millisec)\\
&=(1 - p)200 + p \times 8,000,000\\
&=200 + 7,999,800 \times p.
\end{aligned}
$$

Conclusion: p should be as small as possible.