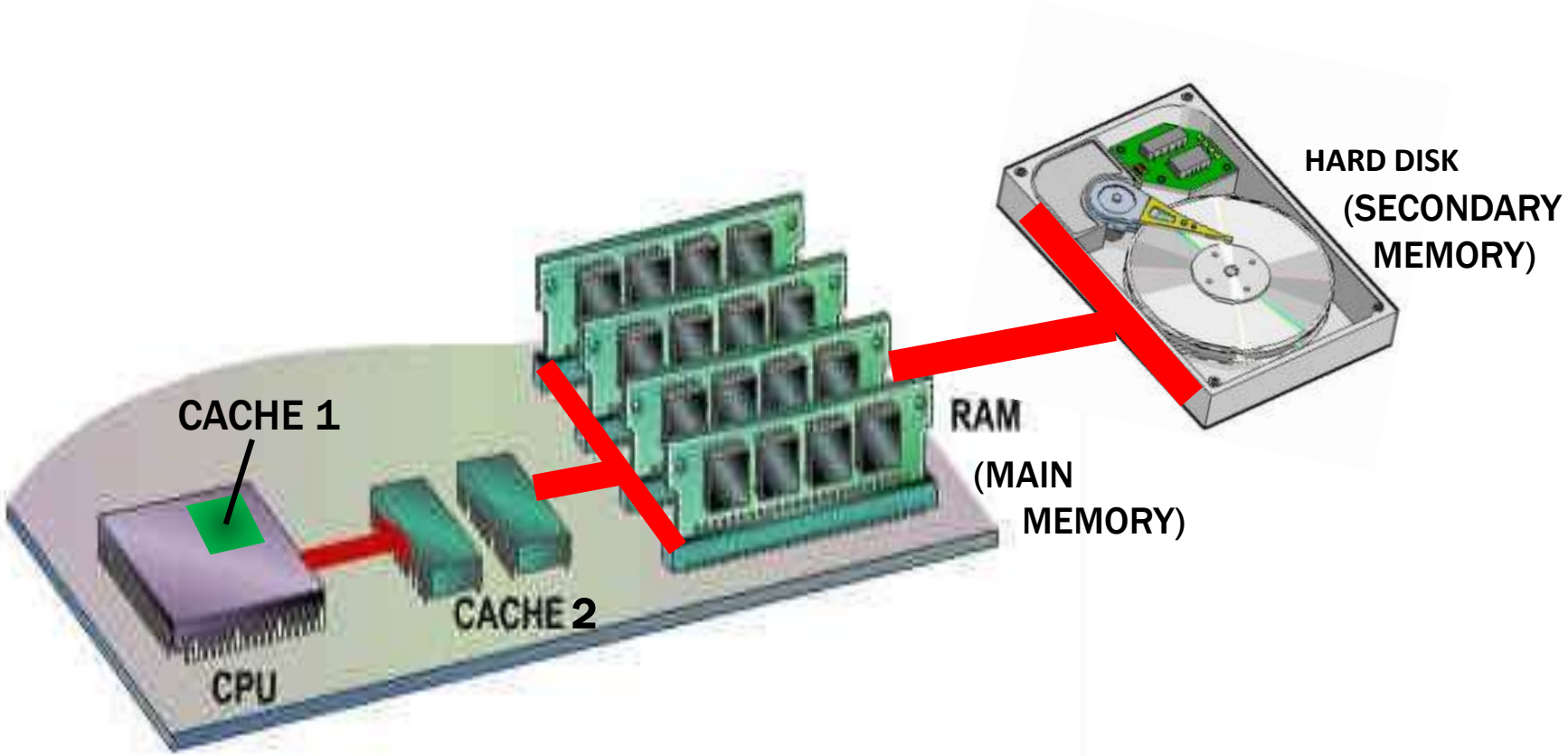
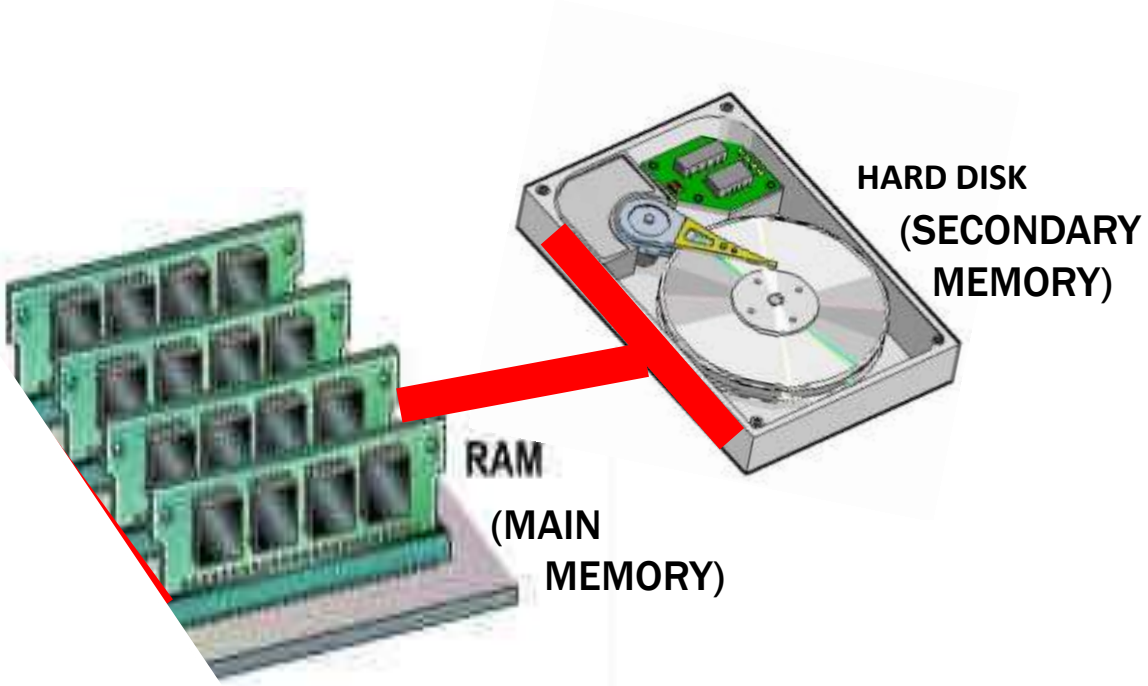
Memory Management: Virtual Memory

Damian Gordon

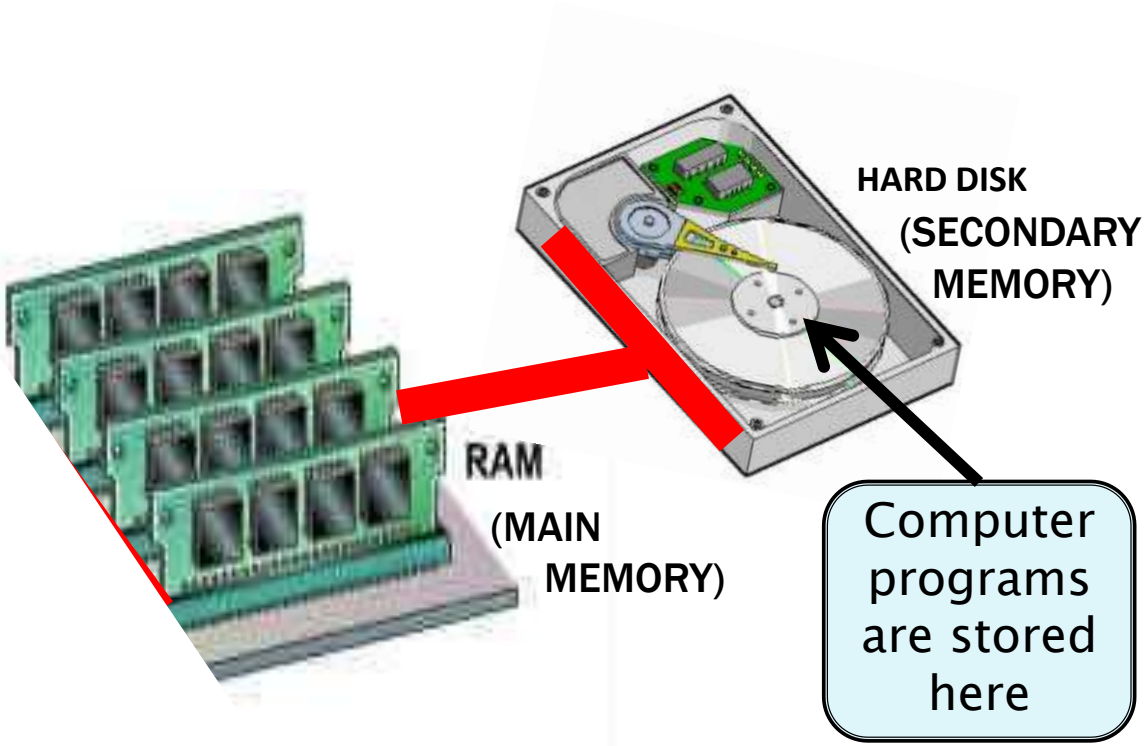
Virtual Memory



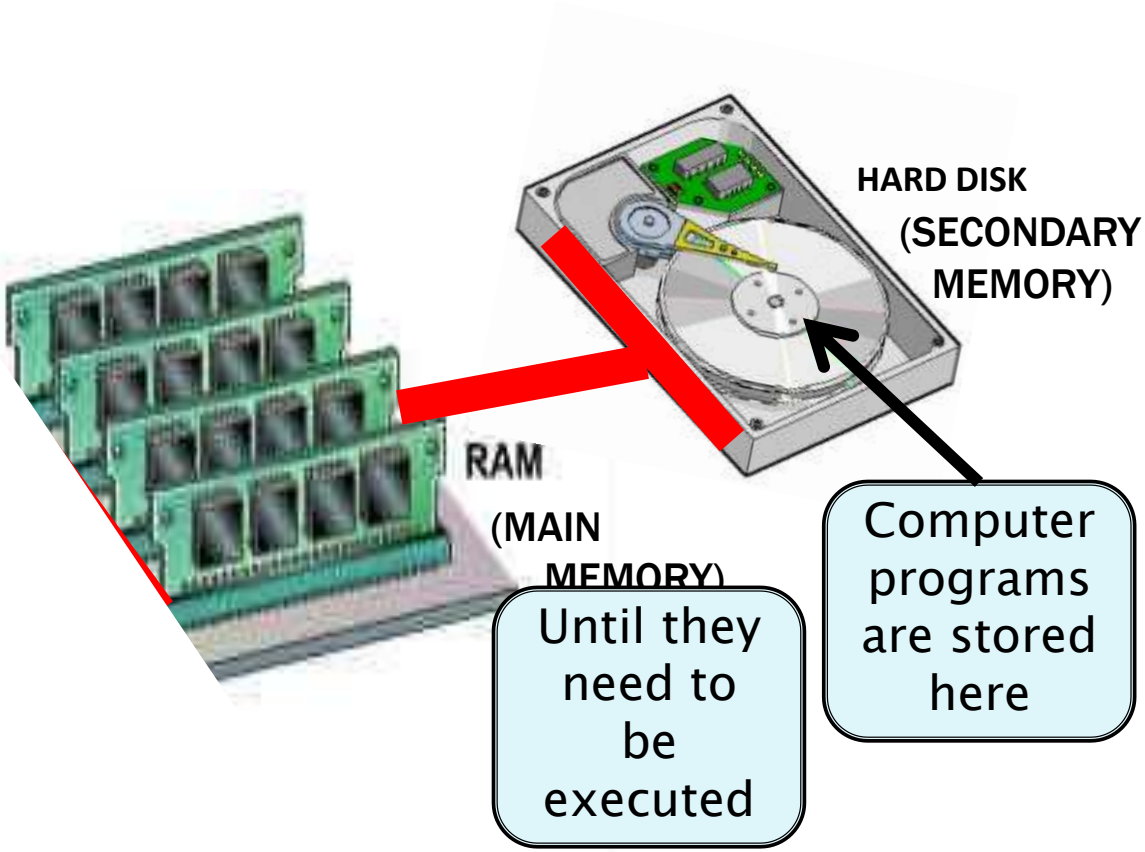
Virtual Memory



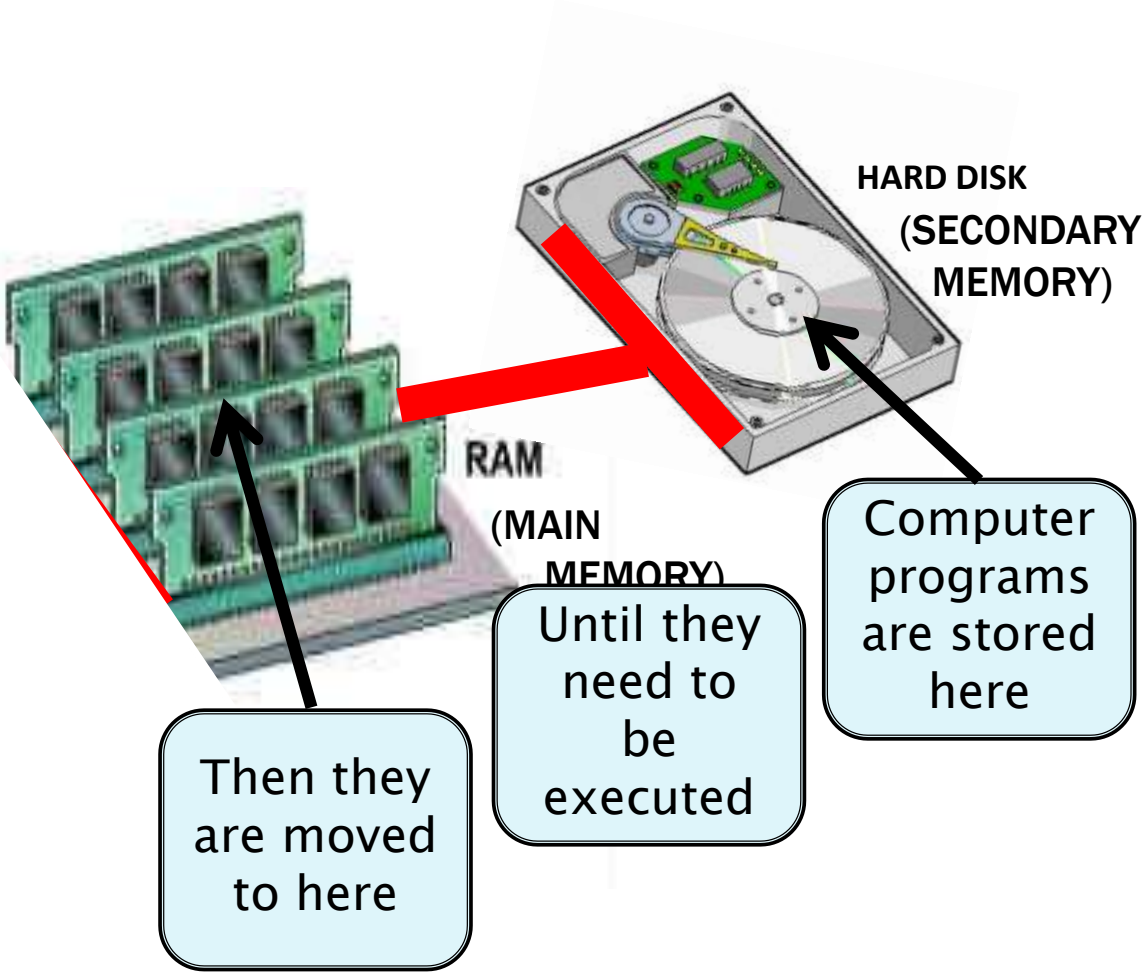
Virtual Memory



Virtual Memory



Virtual Memory



Virtual Memory

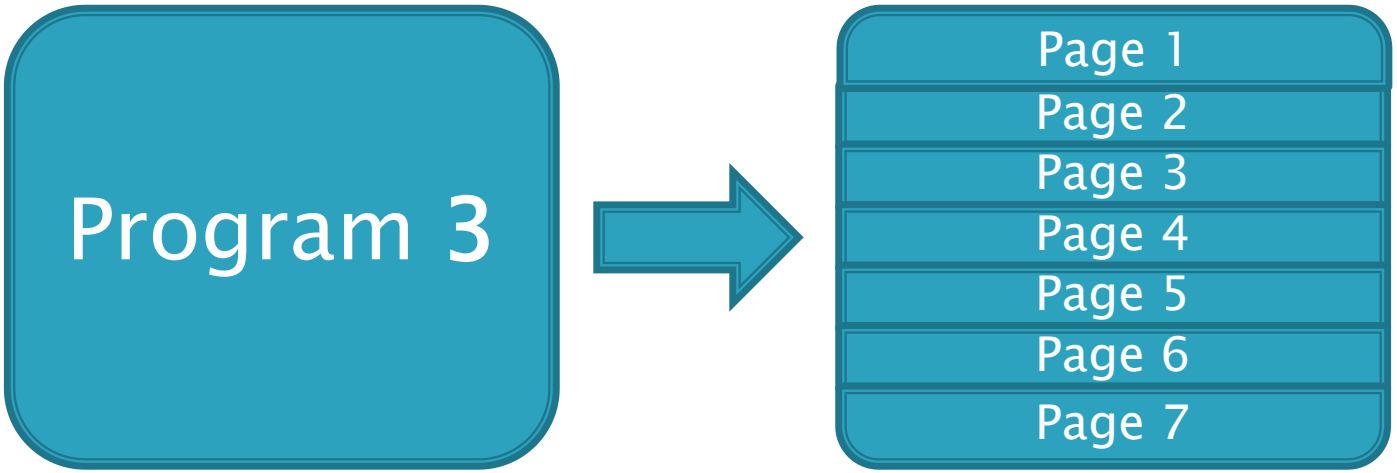
- In modern operating systems, before a program is loaded into main memory, it is divided into chunks, called **PAGES**.

Virtual Memory

- In modern operating systems, before a program is loaded into main memory, it is divided into chunks, called **PAGES**.

Virtual Memory

- In modern operating systems, before a program is loaded into main memory, it is divided into chunks, called **PAGES**.

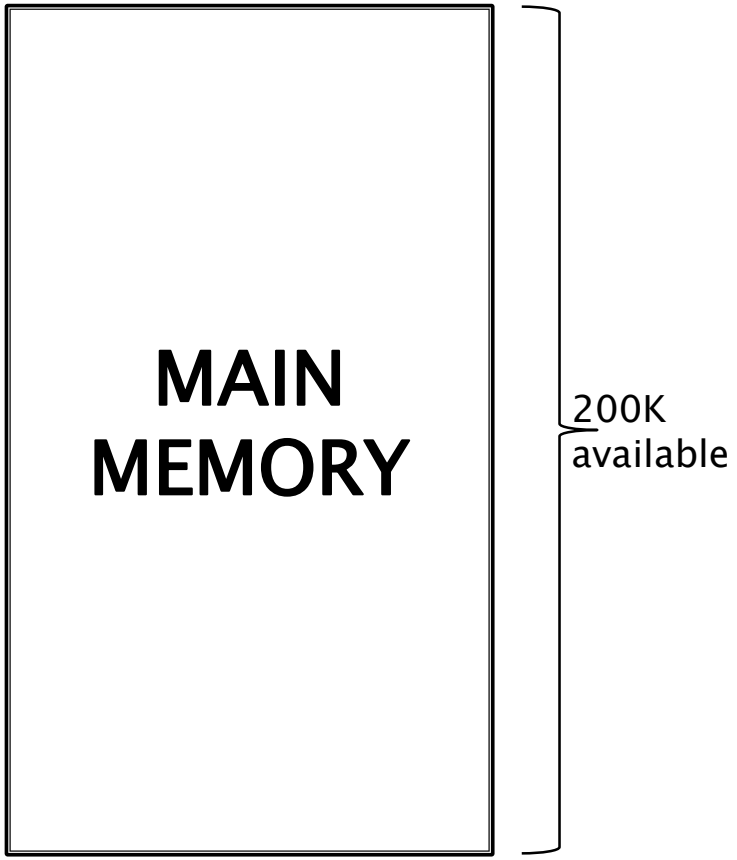


Virtual Memory

- Each **PAGE** is loaded into memory locations called **PAGE FRAMES**.

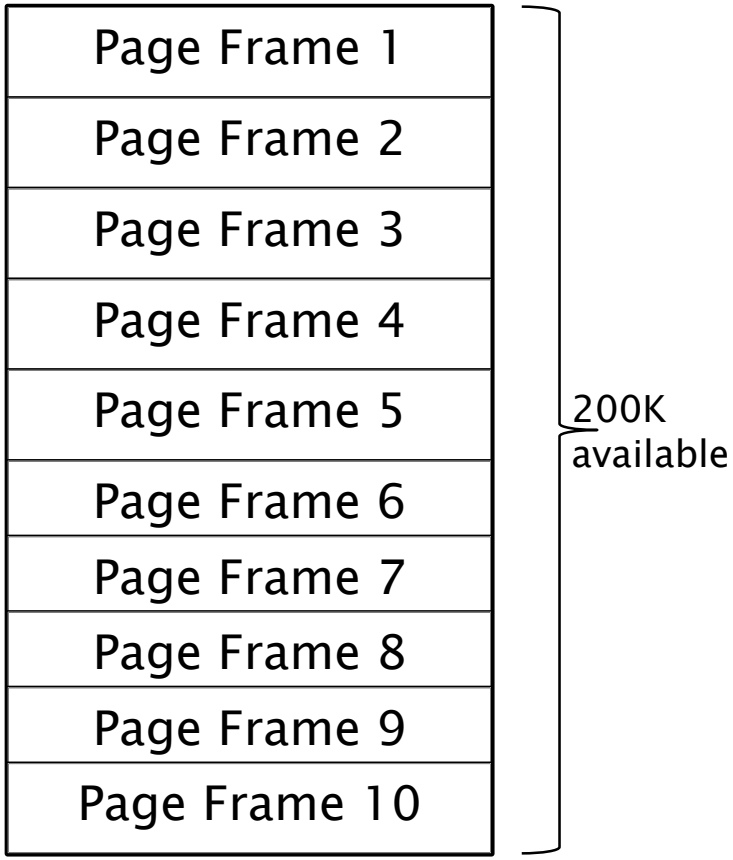
Virtual Memory

- Each **PAGE** is loaded into memory locations called **PAGE FRAMES**.



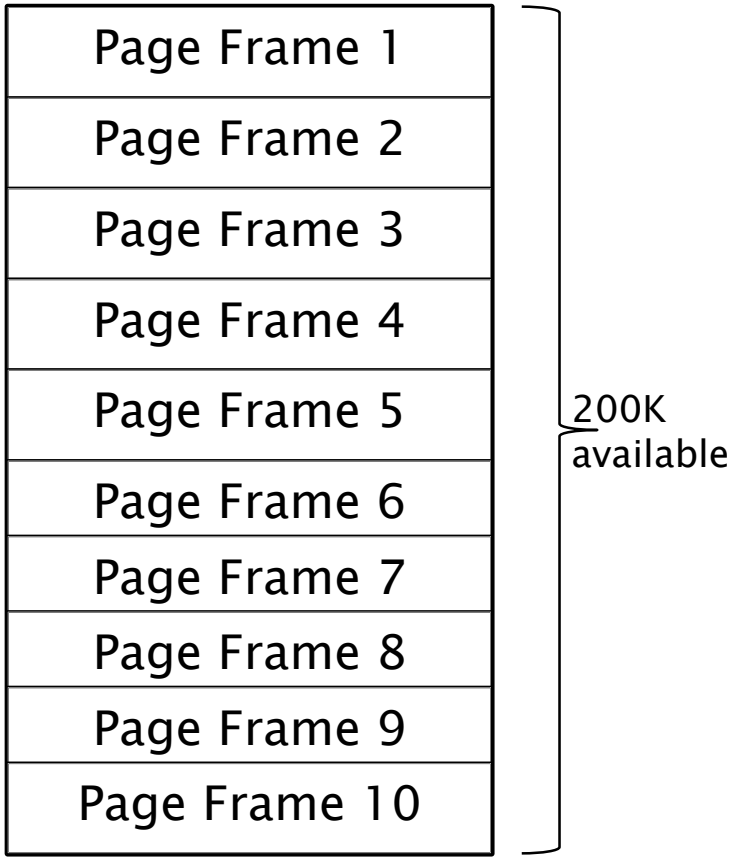
Virtual Memory

- Each **PAGE** is loaded into memory locations called **PAGE FRAMES**.



Virtual Memory

- If the **PAGES** are the exact same size as the **PAGE FRAMES** (and the same size as the disk sectors), this scheme works very well.



Virtual Memory

- The Memory Manager prepares a program for execution by doing the following:

1. Determine the number of pages in the program
2. Locate enough empty page frames in main memory
3. Load all the program's pages into them

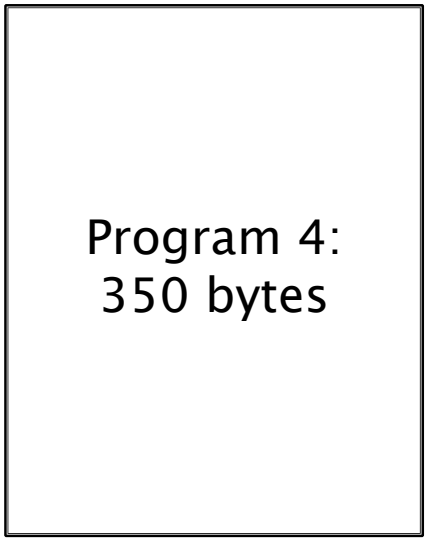
The empty page frame does not have to be contagious.

Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.

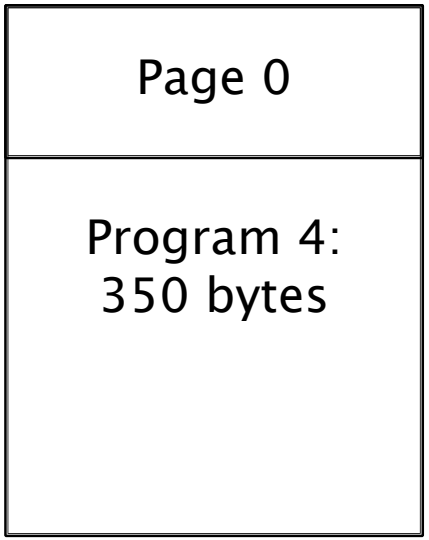
Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.



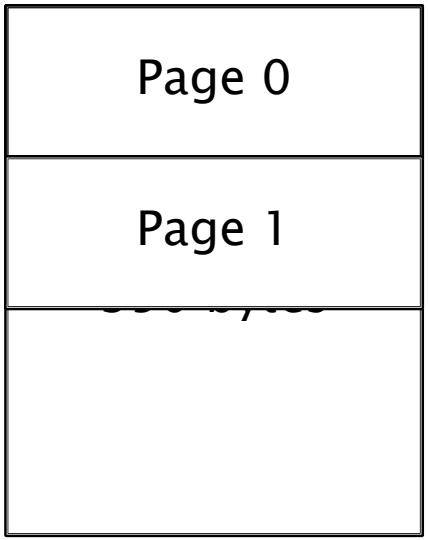
Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.



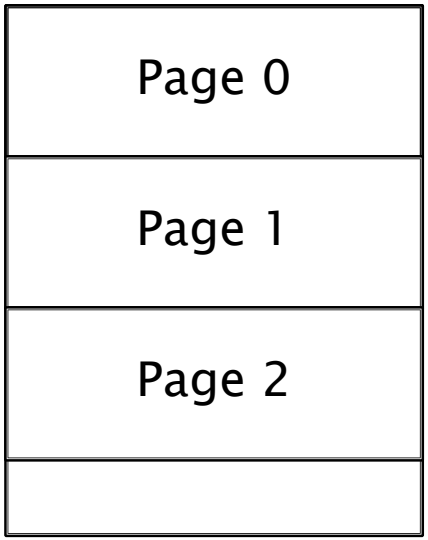
Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.



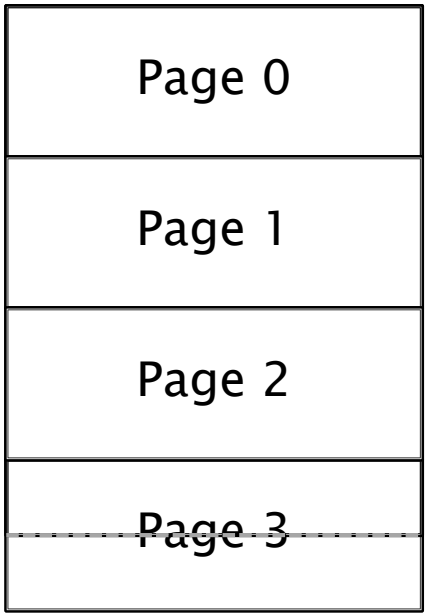
Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.



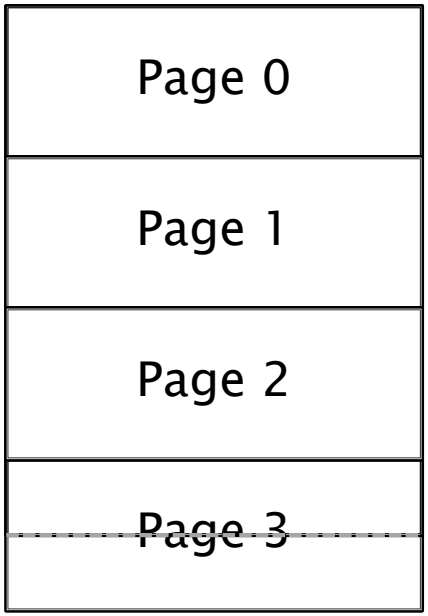
Virtual Memory

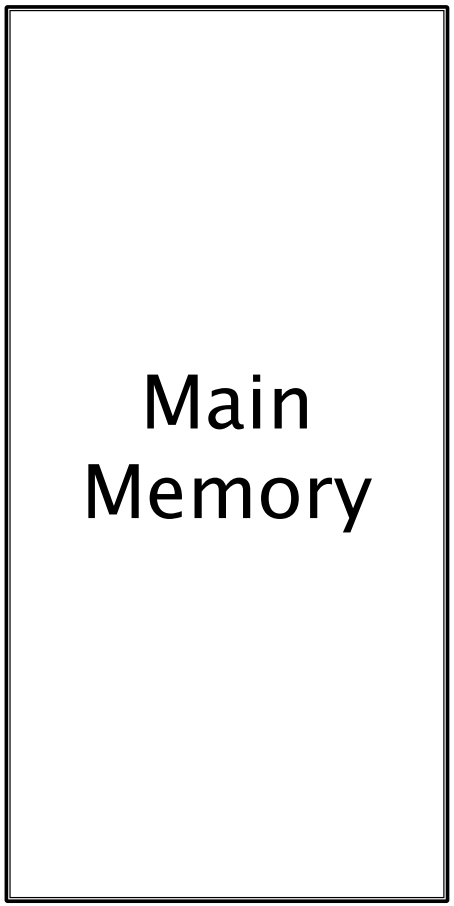
- Consider a program that 350 bytes, and the page size is 100 bytes.



Virtual Memory

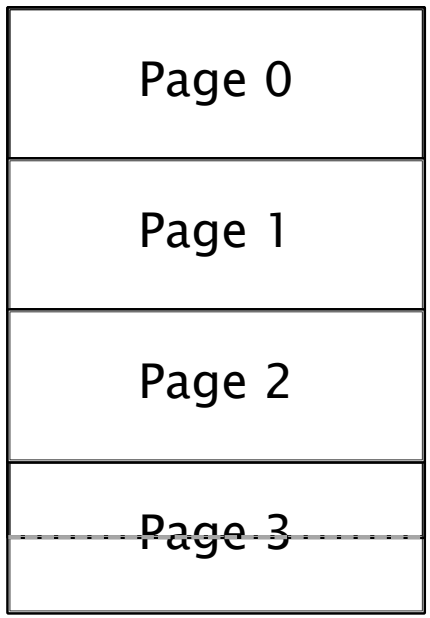
- Consider a program that 350 bytes, and the page size is 100 bytes.

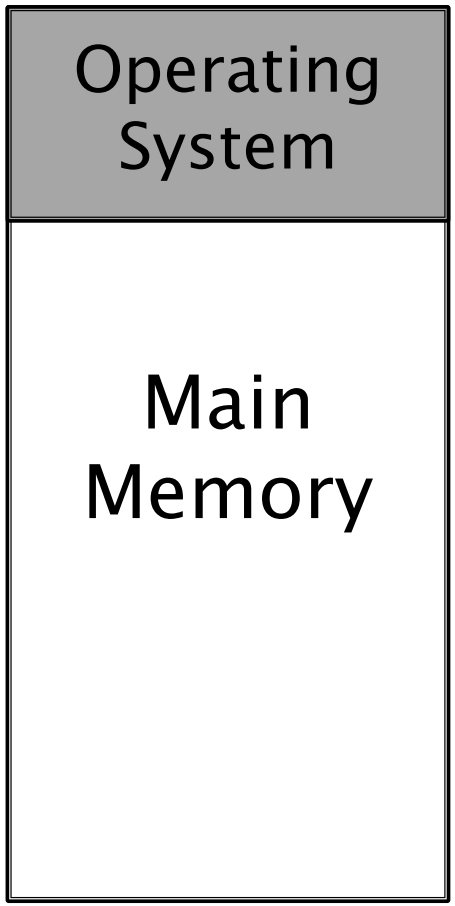




Virtual Memory

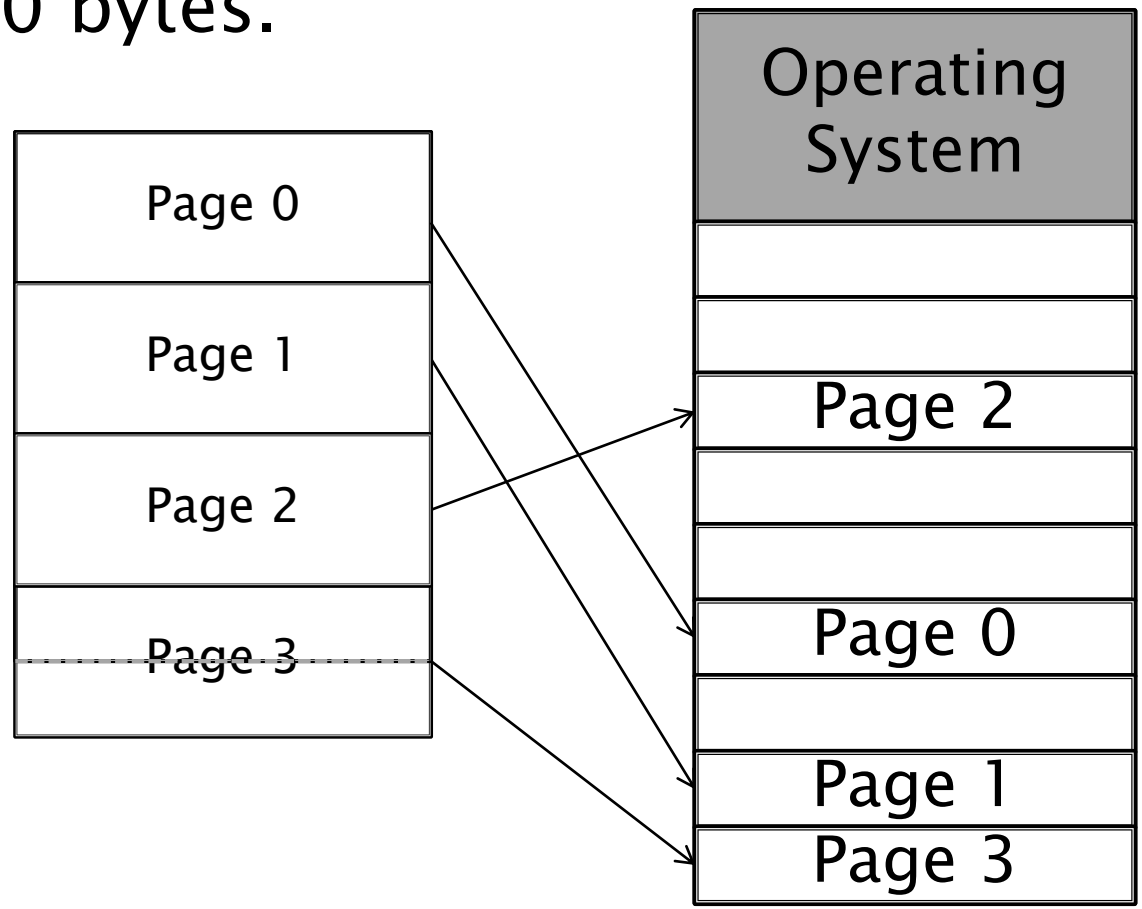
- Consider a program that 350 bytes, and the page size is 100 bytes.





Virtual Memory

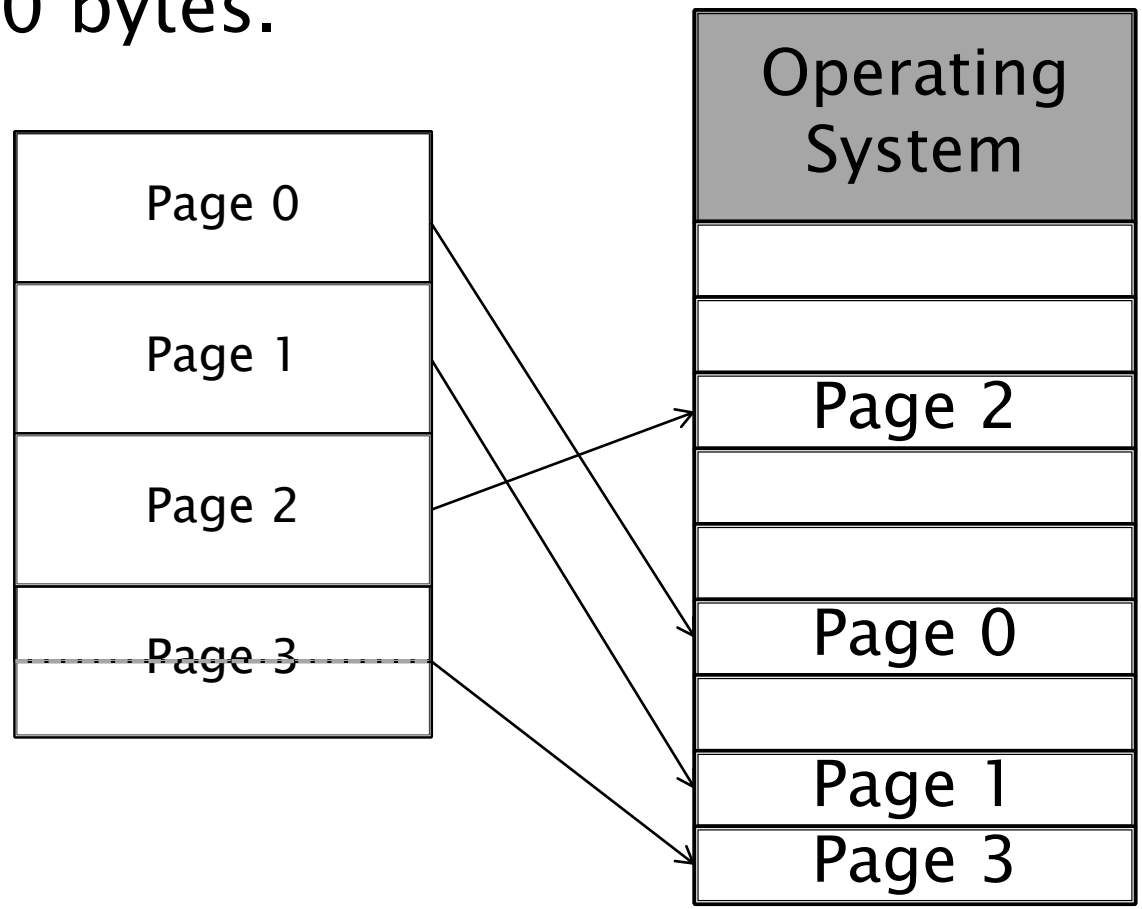
- Consider a program that 350 bytes, and the page size is 100 bytes.



Virtual Memory

- Consider a program that 350 bytes, and the page size is 100 bytes.

A little bit of internal fragmentation



Demand Paging

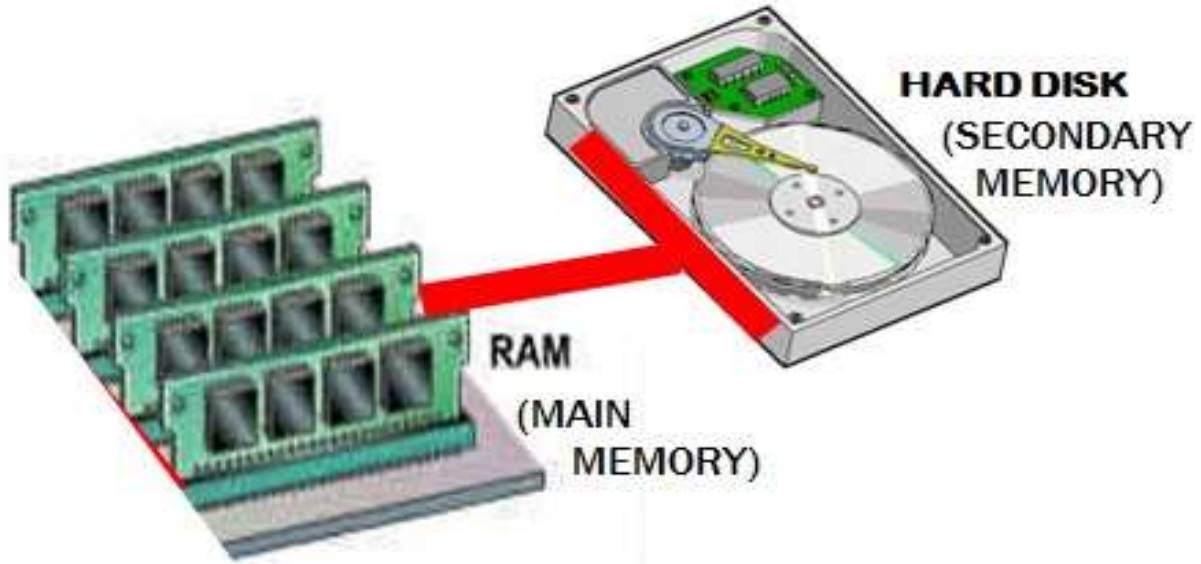
- A useful extension to the notion of **PAGING** is **DEMAND PAGING**.
- Demand Paging introduces the notion that you don't have to load the whole program into memory, just part of it.
- Because not all of the program needs to be in memory at the same time.

Demand Paging

- This means that lots of programs can be run at the same time, and there is an illusion of a significantly larger amount of memory than with regular paging.

Demand Paging

- To make this work, pages have to be moved very quickly from Secondary Storage to Main Memory and back again (this is called “swapping”).



Virtual Memory

- This leads to the concept of **VIRTUAL MEMORY**
- The size of main memory appears much larger than the actual size, since many programs can appear to be fully loaded into main memory at the same time, when in actual fact, only part of many programs are loaded into main memory.