Assembly language programming
By xorpd

BASIC ASSEMBLY

Introduction to Memory

xorpd.net

Objectives

- We will learn about:
  - Why we need the ability to remember more data in our programs.
  - The basic model of computer memory.
  - Memory devices outside the processor.
  - Memory abstraction mechanisms.

Challenge

- Challenge:
  - Write a program that receives a number n, followed by n different integers.
  - The program returns a list of the n given numbers, sorted from the smallest to the largest.
- Limitations:
  - We have to remember all the numbers, so we can sort them.
  - We don't have enough room for all the numbers in our registers.

What's missing?

- So far we have created some pretty interesting programs.

- But they were still pretty simple:
  - Our programs had to process the input immediately and produce output.
  - We have to struggle for space: We don't have so many registers to work with.
  - It seems like our programs could keep only a very small state. They couldn't remember much.

- We would like to be able somehow keep and use more data in our programs.

Computer's Memory

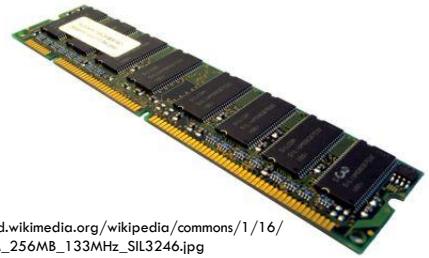
- Computer's memory is usually modeled as a **flat list of cells.**
  - Every cell has a "name", or an address.
  - The contents of every cell could be written to or read from.

0x6ab2	0x6ab3	0x6ab4	0x6ab5	0x6ab6	0x6ab7	0x6ab8	0x6ab9	0x6aba
10010010	11110000	10101111	01110101	00000000	00000000	11100010	00000000	10010011

- Different hardware implementations.
  - RAM, Hard Disks, USB, CD/DVD and more.
  - The x86 processor has many instructions to communicate with the RAM.

RAM

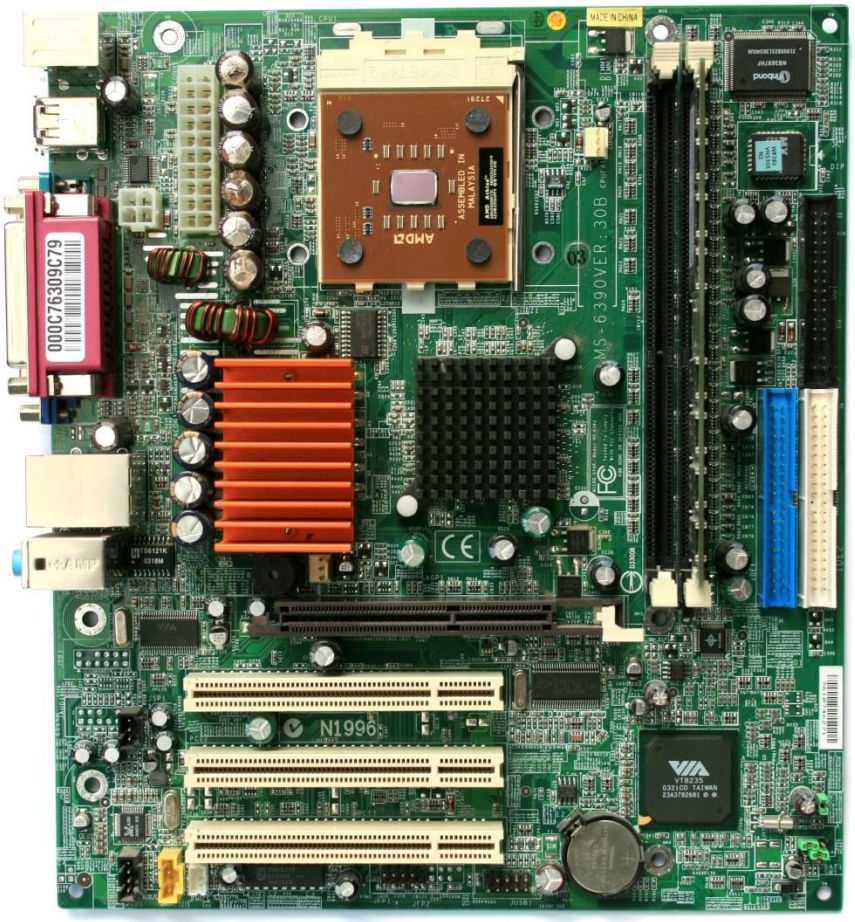
- RAM – Random Access Memory.
  - "Random" means that every memory cell could be accessed directly (in any random order).



http://upload.wikimedia.org/wikipedia/commons/1/16/RAM_SDRAM_256MB_133MHz_SIL3246.jpg

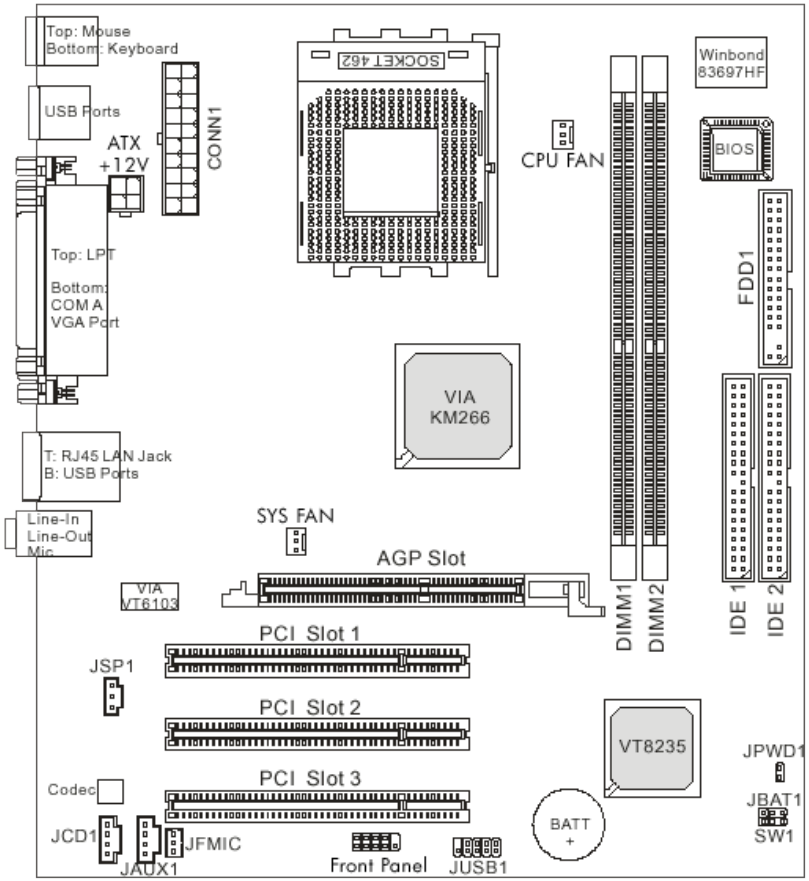
- The processor communicates with the RAM.
  - The processor and the RAM are connected together through electricity in the motherboard.
  - The processor may send "read" or "write" requests to the RAM.
  - The lines which transfer the data are called "buses".

Motherboard layout



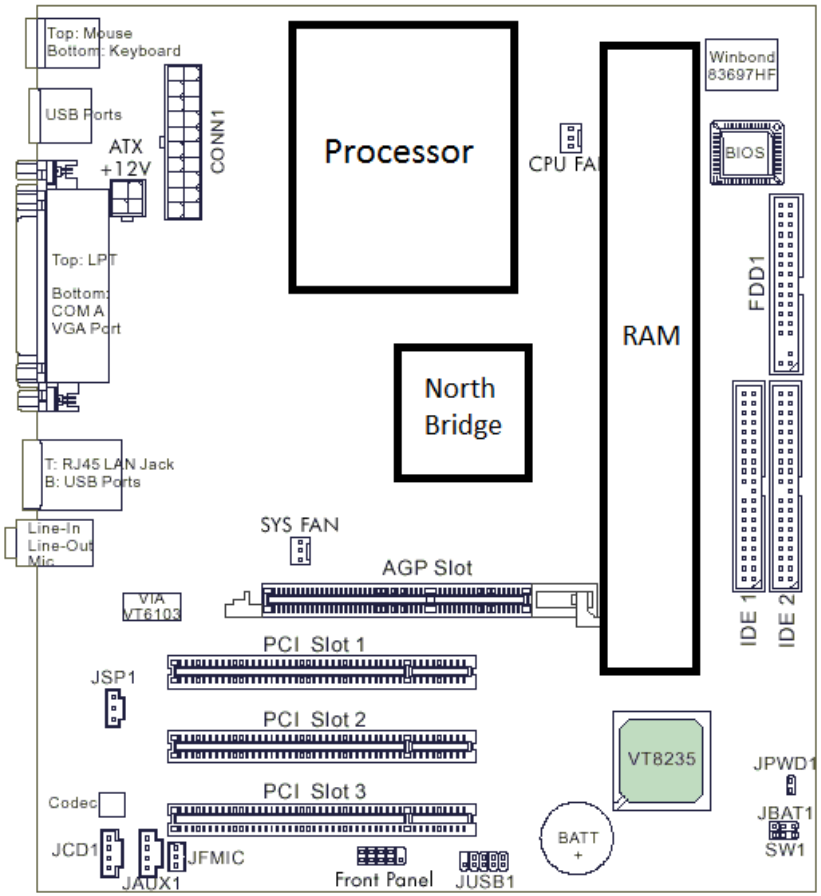
http://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/MicroATX_Motherboard_with_AMD_Athlon_Processor_2_Digon3.jpg/650px-MicroATX_Motherboard_with_AMD_Athlon_Processor_2_Digon3.jpg

Motherboard layout



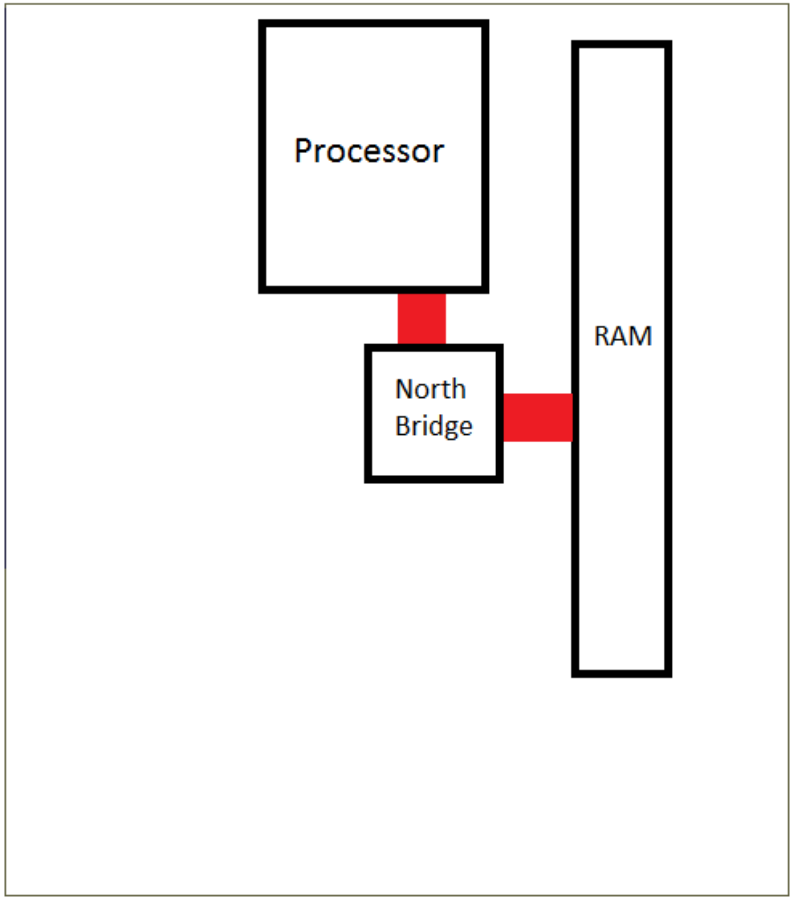
http://h10025.www1.hp.com/ewfrf-JAVA/Doc/images/2/c00411559.gif

Motherboard layout



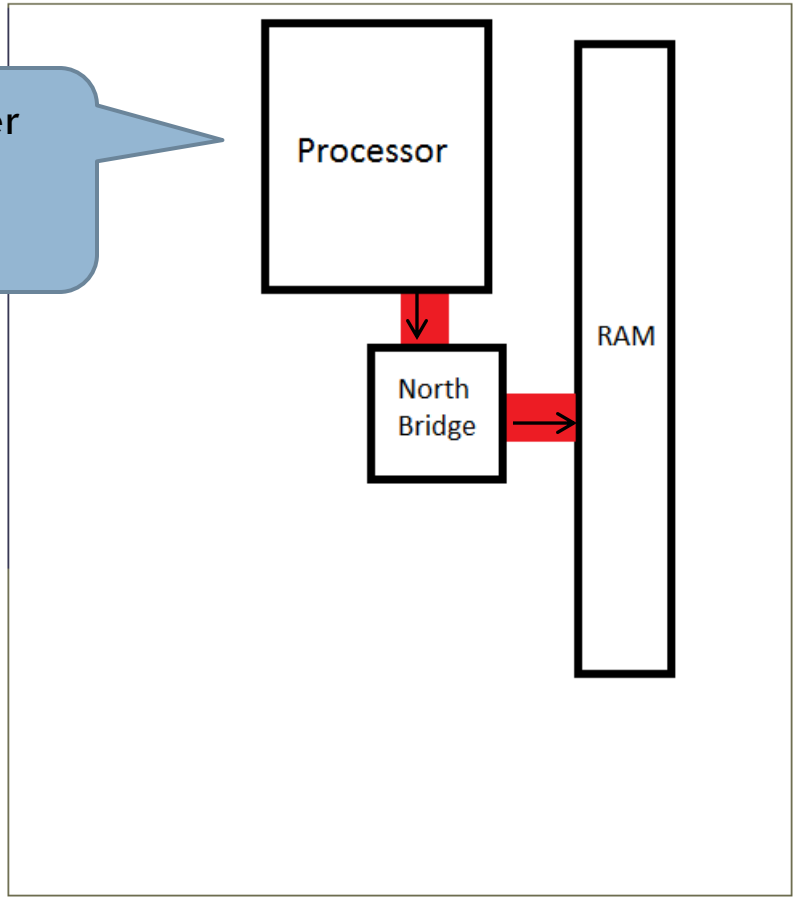
http://h10025.www1.hp.com/ewfrf-JAVA/Doc/images/2/c00411559.gif

Motherboard layout

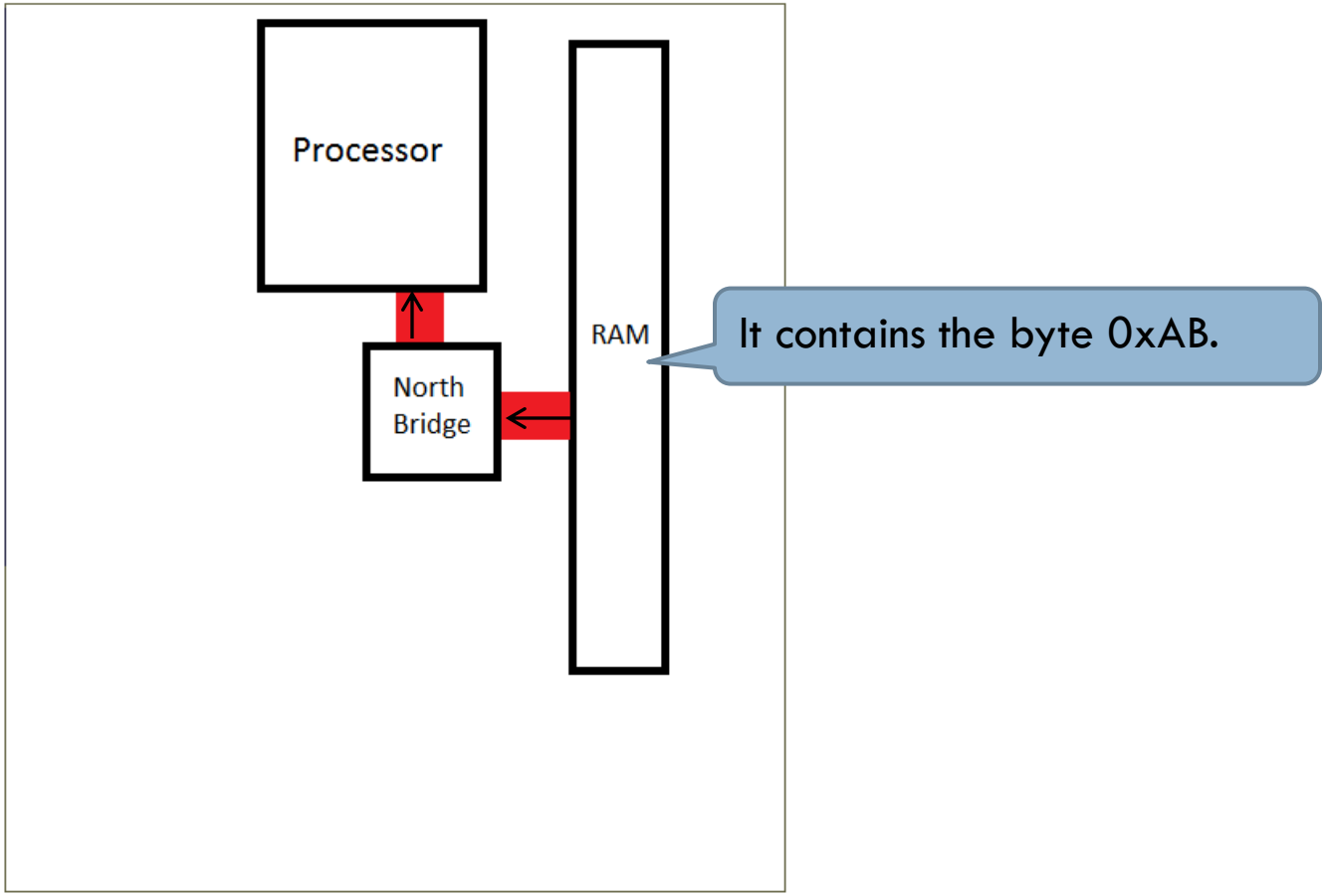


Memory queries

What is inside cell number 0xDE7134 10 ?



Memory queries



Memory Abstraction

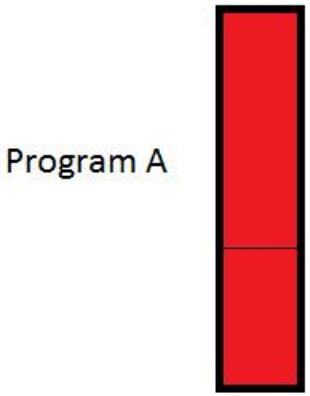
- Memory issues in the early x86 processors:
  - Hard to access all large addresses.
    - Registers were small, so it was hard to represent large addresses.
    - 16 bit registers allow access to 2^{16} bytes, or 64KB.
  - Programs could corrupt each other's memory.

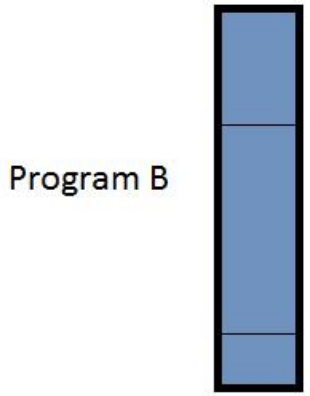
- Later x86 processors introduced new solutions to memory management:
  - **Segmentation** (Intel 8086)
    - Using two registers to represent one longer address.
  - **Paging** (Intel 80386)
    - Simulating a lot of memory, even when actually there is a little.
    - Every program thinks that it owns all the memory in the system.
    - Programs can not override each other's memory.
    - Handles privilege levels system for security.
    - Created by cooperation between the operation system and the processor.

Memory Abstraction (Cont.)

- Today the "end programmer" doesn't have to worry about memory management.
  - Your program will run under the illusion of owning lots of flat memory.
  - In reality, your program shares the total memory of the system with other programs.
  - The operation system and the processor work together to create this illusion.
    - The memory addresses your program sees are not real. They are "virtual".

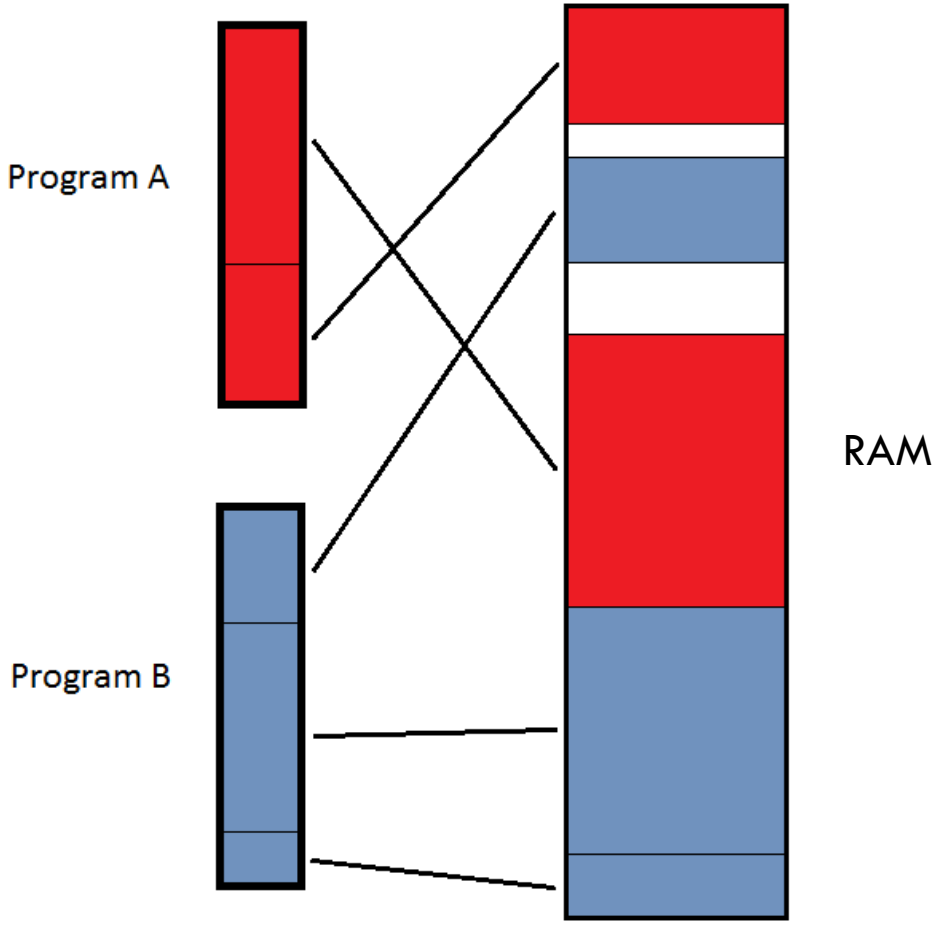
Paging illustration



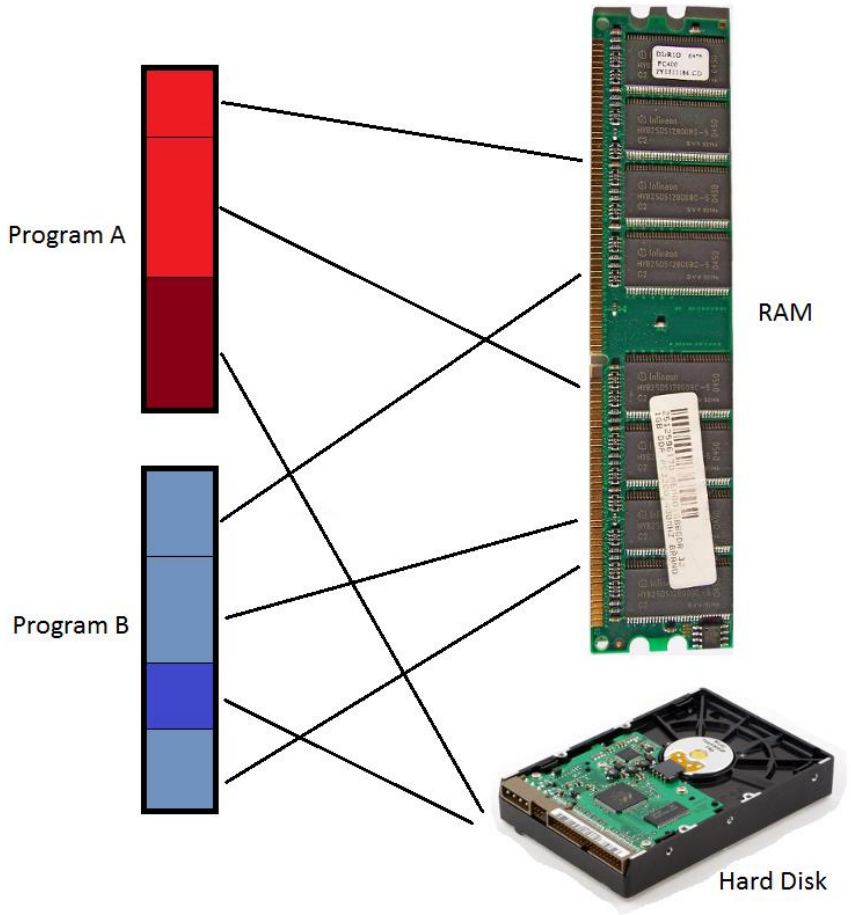


http://commons.wikimedia.org/wiki/File:DDR_RAM-3.jpg

Paging illustration (Cont.)



Paging illustration (Cont.)



http://commons.wikimedia.org/wiki/File:DDR_RAM-3.jpg

http://commons.wikimedia.org/wiki/File:Samsung_HD400LD_Hard_Disk_B.jpg

Summary

- Some tasks require that our programs will be able to remember more data.
- There are hardware devices that store big amounts of data.
  - They are separate from the processor.
  - The processor can communicate with the memory devices using memory related instructions.
- Your programs are given the illusion of flat memory:
  - Contiguous address space.
  - Unique ownership of the memory.
  - Lots of memory.
- You are free to think about your code instead of thinking about memory management.