A typical computer

Englander: The Architecture of Computer
Hardware and Systems Software, 2nd edition
Chapter 1, Figure 01-06
Von Neumann Architecture: CPU
What is assembly?

An assembly language is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions.
What is assembly?

very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions
Why high-level languages?

- Stronger abstractions
  - E.g. Object oriented programming
  - E.g. C, C++, Java, Python
- Increased portability
  - E.g. interpreted languages
  - E.g. Java, Python
- Faster development cycles
- ...
Why assembly?

- Debugging often requires reading assembly
- Understand how things work at the machine level
- Helps you write faster code (even in high-level languages)
Assembly languages (architectures)

- x86, x64 (desktops, laptop, servers)
- ARM (phones)
- SPARC (Sun)
- MIPS
- ...
Registers

- Storage close to the CPU
- Most instructions manipulate registers
  - Manipulation of register content
  - Load and store from registers
- Registers are fast
x86 syntax (AT&T / GAS)

- Registers
  - 16-bit: ax, bx, …
  - 32-bit: eax, ebx, …
  - 64-bit: rax, rbx, …, r8, r9, …
  - Referenced by %REGISTER

- Constants: $0, $1, $0x20 (32), …
x86 syntax

- Arithmetic instructions
  - OP a, b -> b = b OP a
  - add %edx, %eax -> %eax = %eax + %edx

- Assignment instructions
  - OP a, b -> b = a
  - mov %edx, %eax -> %eax = %edx

- Condition testing
  - OP a, b

- Control flow
  - OP address
  - jmp address -> unconditionally jump to address
Live demos!
Further questions

- How are parameters passed to functions?
- How are values returned from functions?
- Do all instructions take the same amount of time?
- How does caching work?
- What are the differences between ARM and x86?
Further material

- Intel syntax
- External assembly: yasm, masm, etc
- ddkang@mit.edu
- 6.004, 6.033, 6.172