**Outline**

- FP Q&A - Whis
- Conventions, contracts & protocols
- Starting a program
  - Turning on a machine

**Q&A**

- What does a computer do?
- What do they mean by symbols?
- What do they mean by machine language?
- What do they mean by data representation?
- What do they mean by two's complement?
**A**: Why does it matter?

- **B**: Is 0 any different?

**Proof**

```
\begin{align*}
A & \not\Rightarrow A-B = 0 \\
B = 255 & \quad \text{why}, \quad 0, \\
0 & \quad \text{Denum} \\
-150, M = 2 & \quad \text{even} \\
\end{align*}
```
A special program is always running called the operating system.

- Reads a file into memory
- Jumps to start instructions
- Header of executable
- Language RT calls MAIN

code
state
data
symbols
file

1.6-sec.o

- Registers
- Address space
- Stack, SP, GP
- Run-time data structures: malloc
- argv, c
- argc, V
- stdin, stdout, stderr

CRT0
- main()

The "Other half"
Q: What function do you call?
- prog knows the name.
- machine needs an address
- not known until all the objects are put together
- linking resolves the address

* like what you did with forward references in assembler

Language run time

- sets up machine state to meet language conventions
- sets up data structures and application program state

⇒ System calls.

OS:
- provides a common abstraction of physical resources
  - disk = file
- protected access to shared resources

How does apply "call" a system function
- don't have address of routine
- separate address space
- system protects itself
  - not on your box
3 Kinds of entries

Traps (synchronous to inst) => syscall

Exceptions

Syscall

Place arguments in registers
create syscall inst

OS init vector

- System call has access to
  HW resources that are
  not accessible to application

General form: PC + 0x0bfc0

- Call to determine 
  cause of the
  trap or interrupt
  event

- RFE inst
What happens when you turn it on?

PC = 0x10FC0 0000

registers undefined

⇒ OS boot strap
  - verify HW is operational
  - bring up all the subroutines
  - start process schedule

How did OS get there

- from eg. BIOS
- boot loader - reads blocks 0 from disk