Administrative Overview

- 6 Projects
- Design Review: Monday before 6:30pm
- Lab Friend Center 010 ("Fishbowl")
COS 318 Project 1
Bootloader
Problem

- We will write an Operating System
  - Manages programs, resources, users, etc.
- How are programs loaded?
  - The OS takes care of this
- How is the OS loaded?
Booting a Computer
On Startup…

- The BIOS is loaded
  - Typically doesn’t know anything about the OS
  - Minimal functionality
- The BIOS loads & runs the first sector of a boot device.
  - An OS cannot fit in just one sector
Bootup Details

- Start at 0xFFFFF0
- Self test & initialization
- Search for a boot device
  - Hard disk
  - Floppy
  - Flash
  - …
Bootup Details

- 1st sector loaded to 0x7c00
- Jump to 0x7c00
- 512 bytes to load the kernel
Bootloader

Disk

Memory

- BIOS: 0x100000 = 1MB
- DEVICES: 0xFF0000
- VGA DISPLAY: 0xC00000
- Kernel STACK: 0xA0000 = 640KB
- KERNEL: 0x07C00 (bootloader)
- BIOS DATA: 0x1000
- 0x0
Entering the Bootloader

- %dl = Boot device number
  - Load the kernel from this device
- %cs = Code segment
- NO STACK! (%ss, %sp unset)
- %ds unset (set it to 0x07c0 before fetching from memory!)
- Other registers unset
The kernel might be big (extra credit)
Solution (extra credit)

- Move the bootloader
Loading the kernel

- Load to address 0x0000:1000
- Set up the stack
- Set %ds for kernel
- Switch control to the kernel (long jump to kernel)
Addressing

- **Real Mode**
  - 1 MB
  - Format: 0x0000:0000
  - Physical address = (segment << 4)+offset
    - Ex: 0x07c0:0000 = 0x0000:7c00

- **Protected Mode**
  - 4 GB (32-bit)
  - Format: 0x0000:00000000 (32-bit)
  - Virtual Addressing (user mode)
  - Physical address = a bit more complicated…
Registers

General Registers
31 16  8  0

E_X

  _X

  _H   _L

EAX, EBX, ECX, EDX
AX, BX, CX, DX
AH, AL, BH, BL, ...

Segment Registers
16  CS, DS, SS, ES, FS, GS

Index Registers
32  BP, SI, DI, SP

Status & Control
32  EFLAGS, EIP
AT&T Syntax

- Registers: %ax, %ah, %eax, ...
- Definitions
  - .equ BOOT_SEGMENT, 0x07c0
- Constants: $0x0100, $4
- Labels
  - _start:
  - print_string:
- Memory access
  - movw %ax, (0x40)
  - movb %dl, (a_label)
  - movw %es:(%ax), %dx
- Comments
  - /* multiline */
  - # to the end of the line
- Directives
  - .equ, .byte, .word, .ascii, .asciz