CS 4410 Operating Systems
http://courses.cs.cornell.edu/cs4410/2018su

- What is OS?
- Why study OS?
- Logistics
- Soviet-era keyboard exercise
What is OS?

- Abstraction layer
- Sits on top of hardware
- Resource management
  - Memory
  - Cores
  - CPU
  - Devices
- "Applications" (isolate programs from each other)

Uniformity
- No need to know details

Virtualization
- Illusion of total control
Components/Outline

- Devices (I/O, keyboard, screen...)
- Interaction between applications/threads
  - Writing correct multithreaded programs
- Memory
- Filesystems/Disk
- Networking
- Security
Why take OS?

- Studying software designs
  - design by example.

- Abstractions are leaky
  - e.g. performance
  - need to know how abstractions are implemented.

- Need to build OS?
Logistics

- Weekly HW (out on Mondays)
- Weekly quiz (end of class on Mon)
- Final exam (in-class, last day)
Academic integrity

- Do your own work
  - don't share solutions
  - don't copy solutions
  - don't look up solutions to similar problems
  - don't copy my course materials.
Build a keyboard

- gates (and/or/etc...)
- switches for keys

- encoders, muxes, latches, ...
- tri-state buffer
Strawman 1

Easy, just add lots of PMS to CPU
- Cons: lots of soldering
Strawman 2

Pros: less soldering.
Cons: still soldering.
- "logic": 104 to 7 encoder, kinda expensive.

Unresolved:
- how to interact with program?
Strawman 3

- Pros: 2-to-3 + 16-to-4 much cheaper than 128-to-7.
- Unresolved: now what?

7-bit

Encoder (which row?)

Encoder (which col?)
Strawman 4

- Buttons (as in SM3)
- Mouse
- 8 bit keycode
- Mic

Con: Soldering galore!

Strawman 5

- CPU
- Buttons
- Mouse
- 8 bit bus
- 8 bit key code
- 8 bit pos.

Unresolved:

1. Where does control come from?
2. What happens to keystrokes while disconnected?
3. What does "hello world" program look like?