

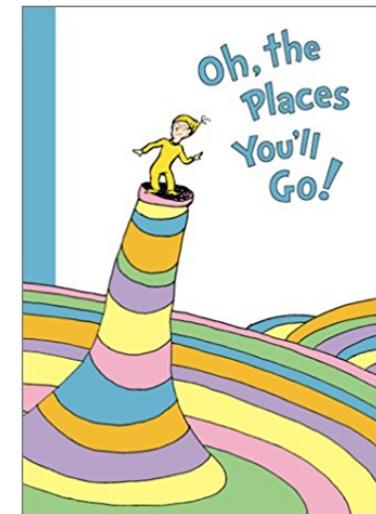


CS107e Wrap Party

Where we started



How far we've come



Where to go next

Bonus: Ask us anything!

Last lab this week

- Check-in: progress, milestones, path to completion

Project demo day

- Fri Mar 23 9-11:30am in B21
- Each team gives 2 minute project pitch at start
- Setup up and demo to students and instructors

Project submissions

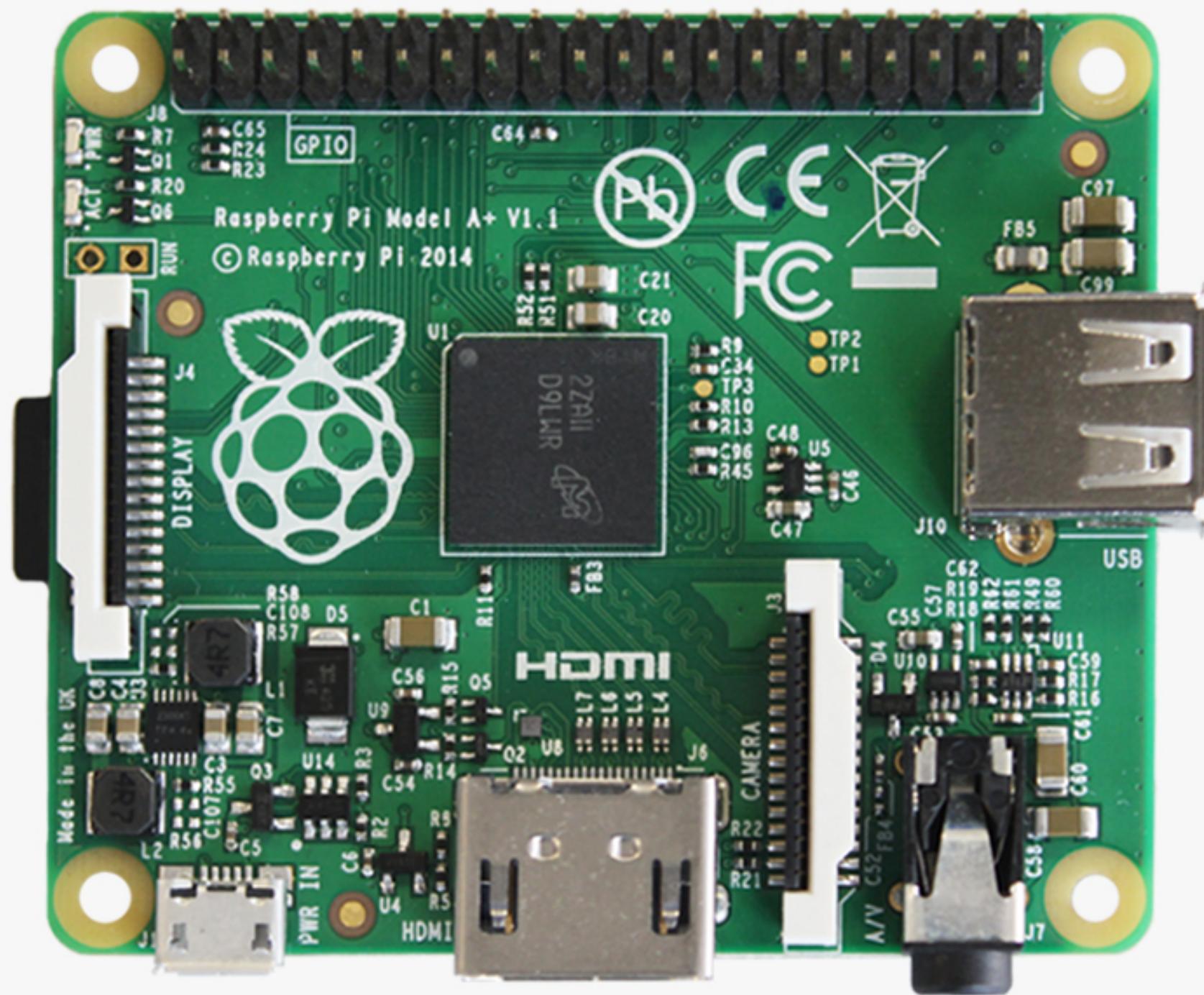
- Due midnight Fri Mar 23rd, final GitHub commit of code
- Include README.md describing your project
(pictures!!, attributions)

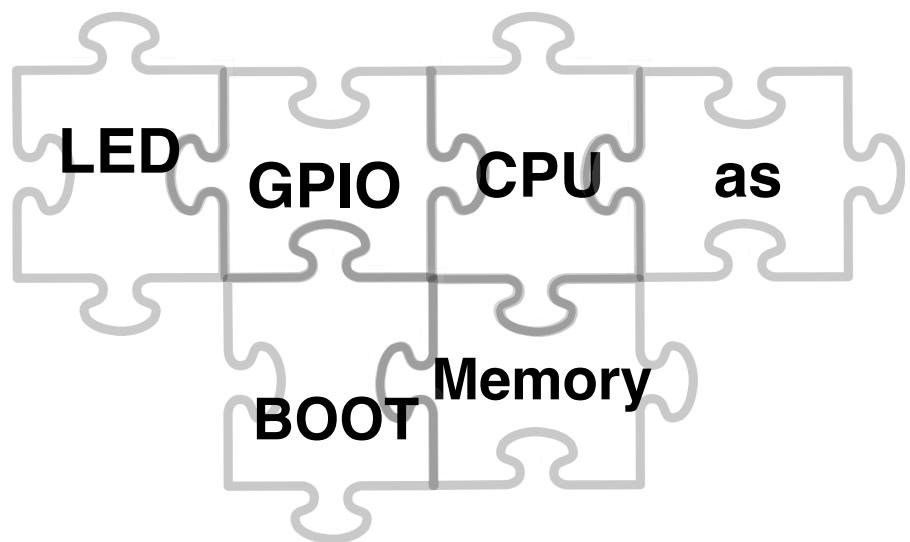
End-quarter logistics

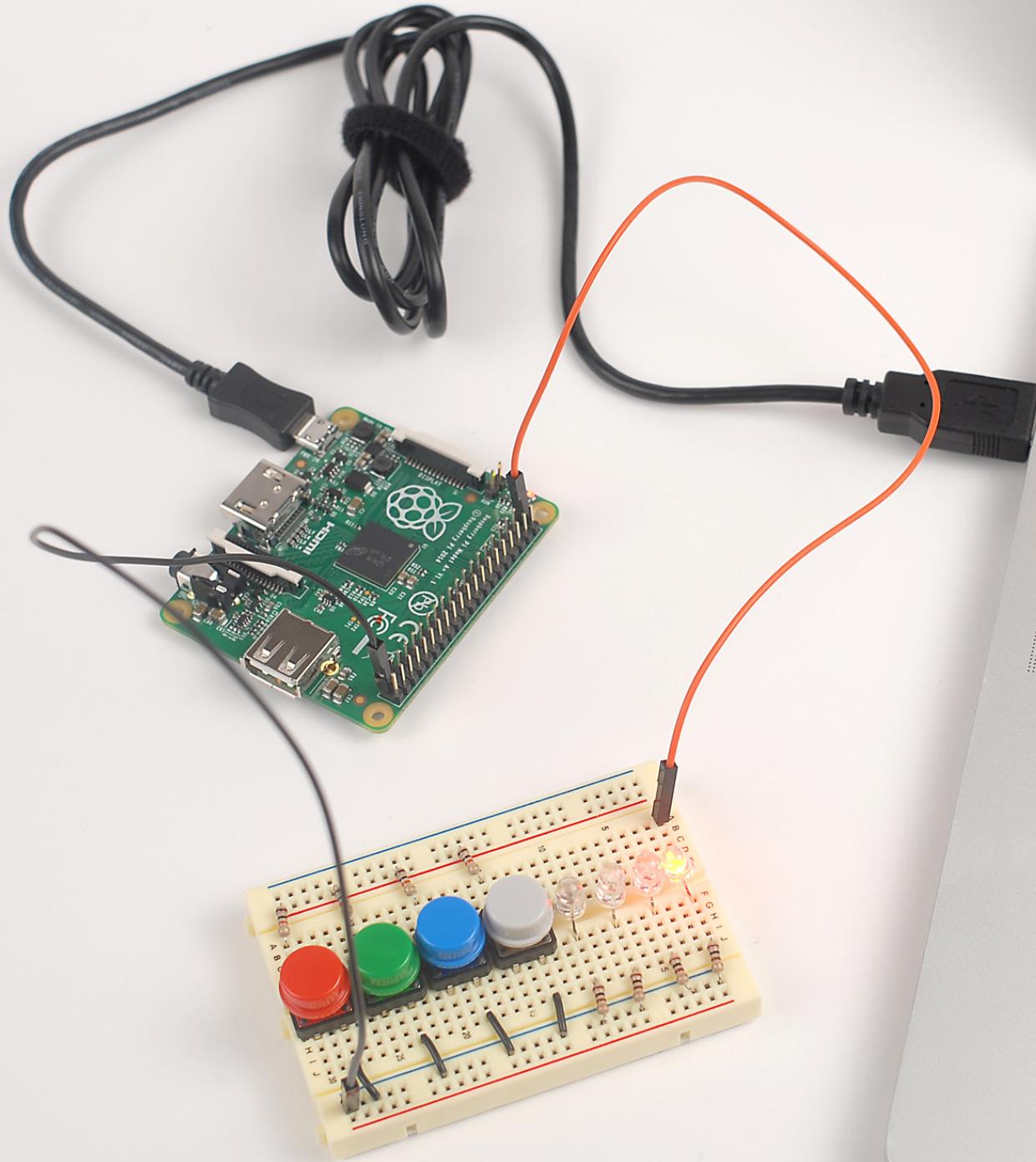
- Return keyboards+mice (Pi kit is yours to keep)
- Submit receipts for reimbursements

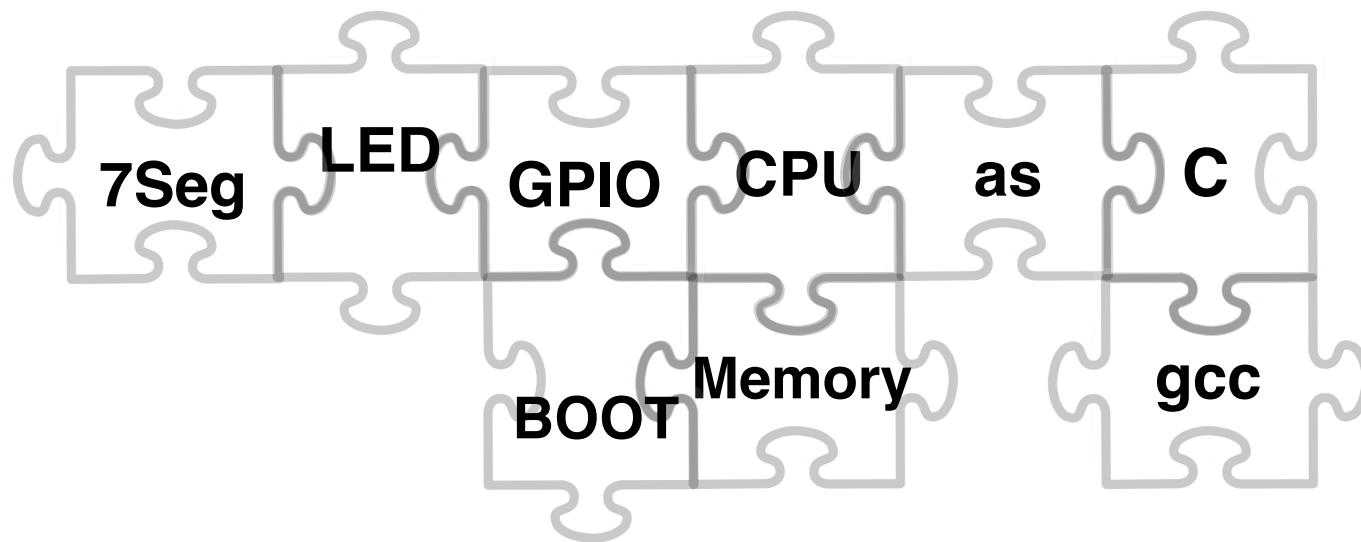
Course learning goal #1:

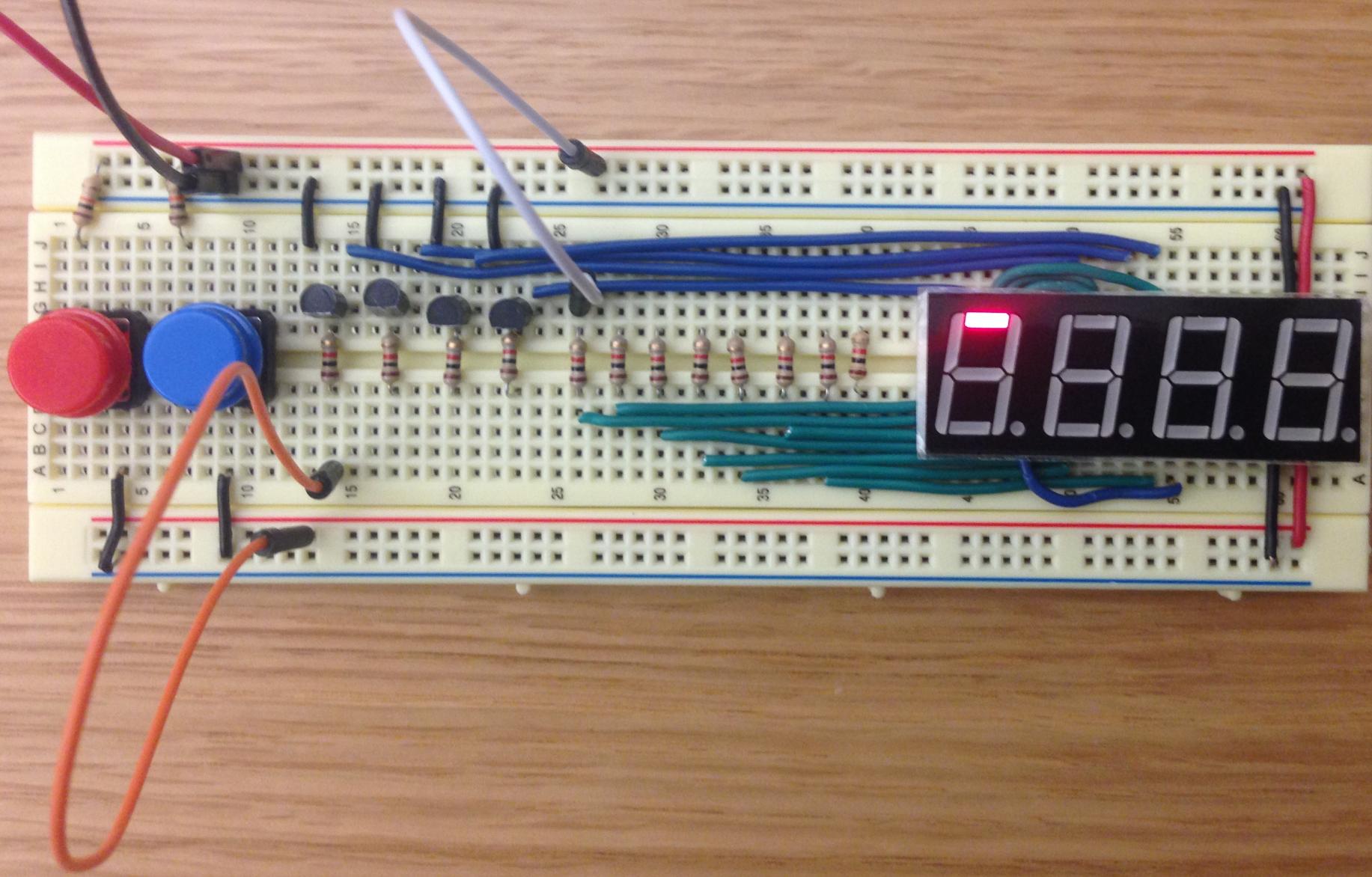
**Understand how
computers work**

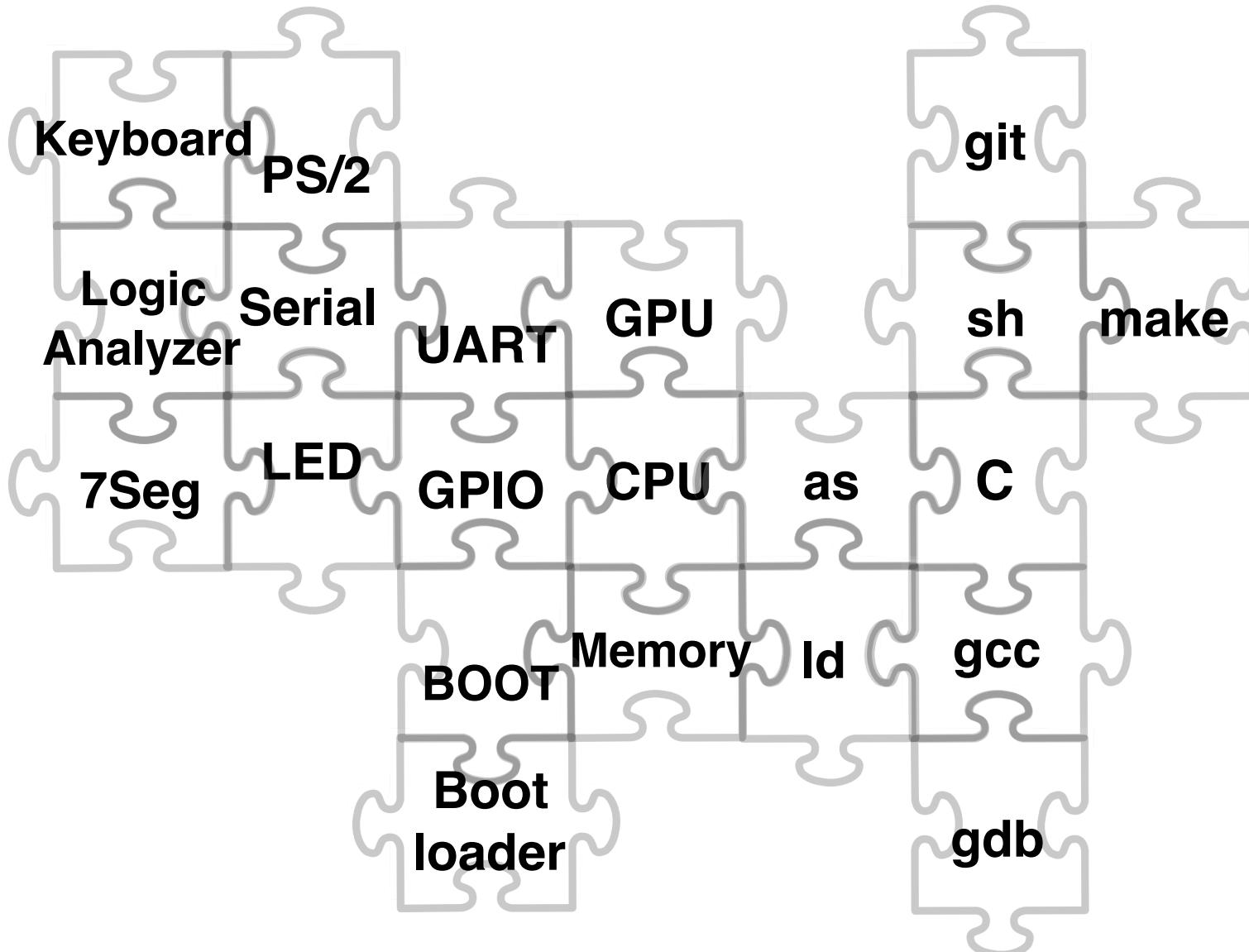




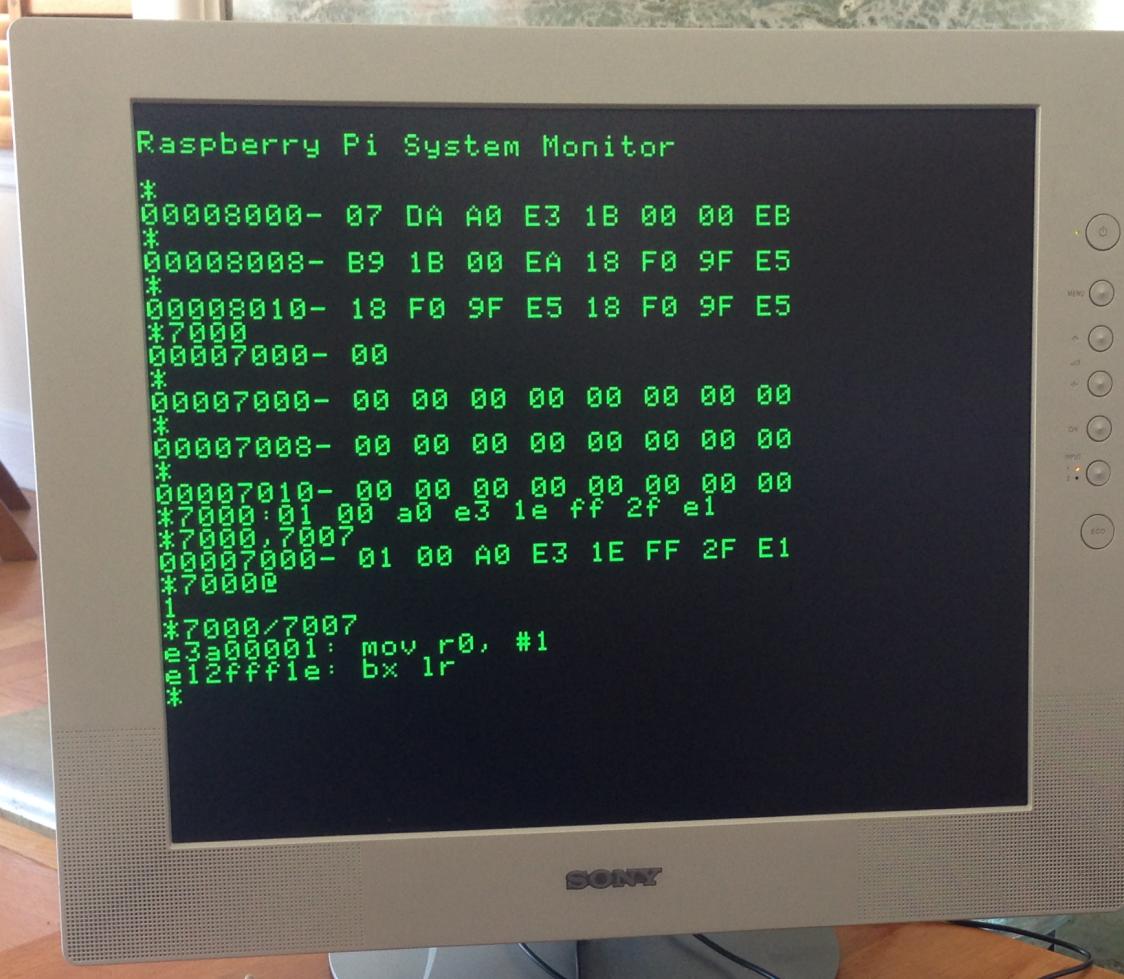
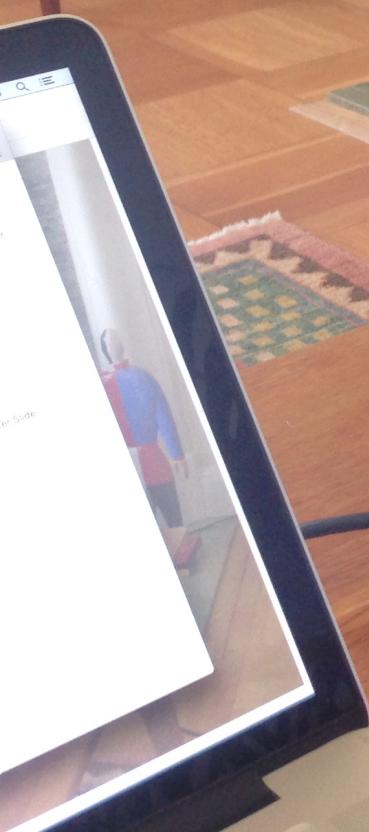




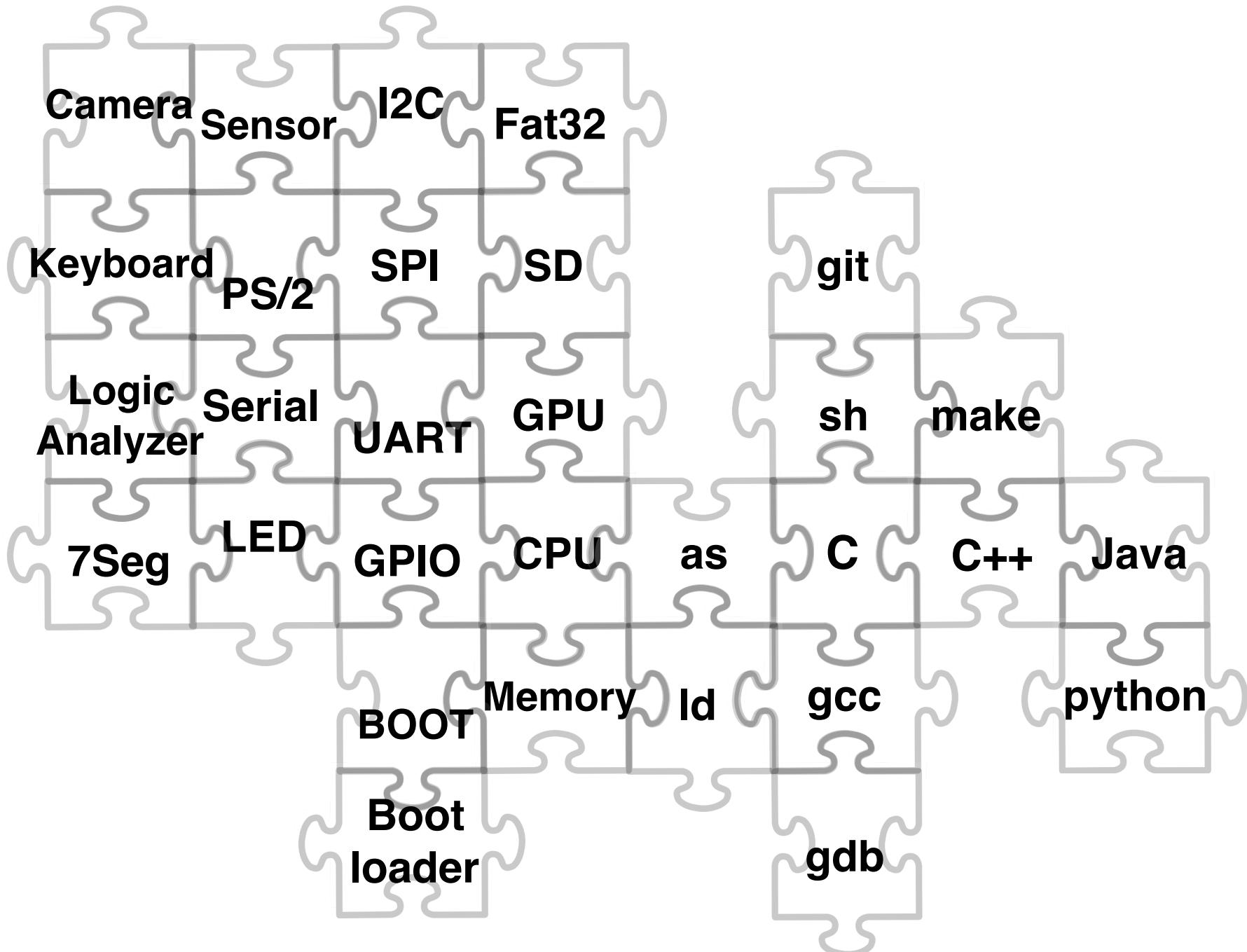


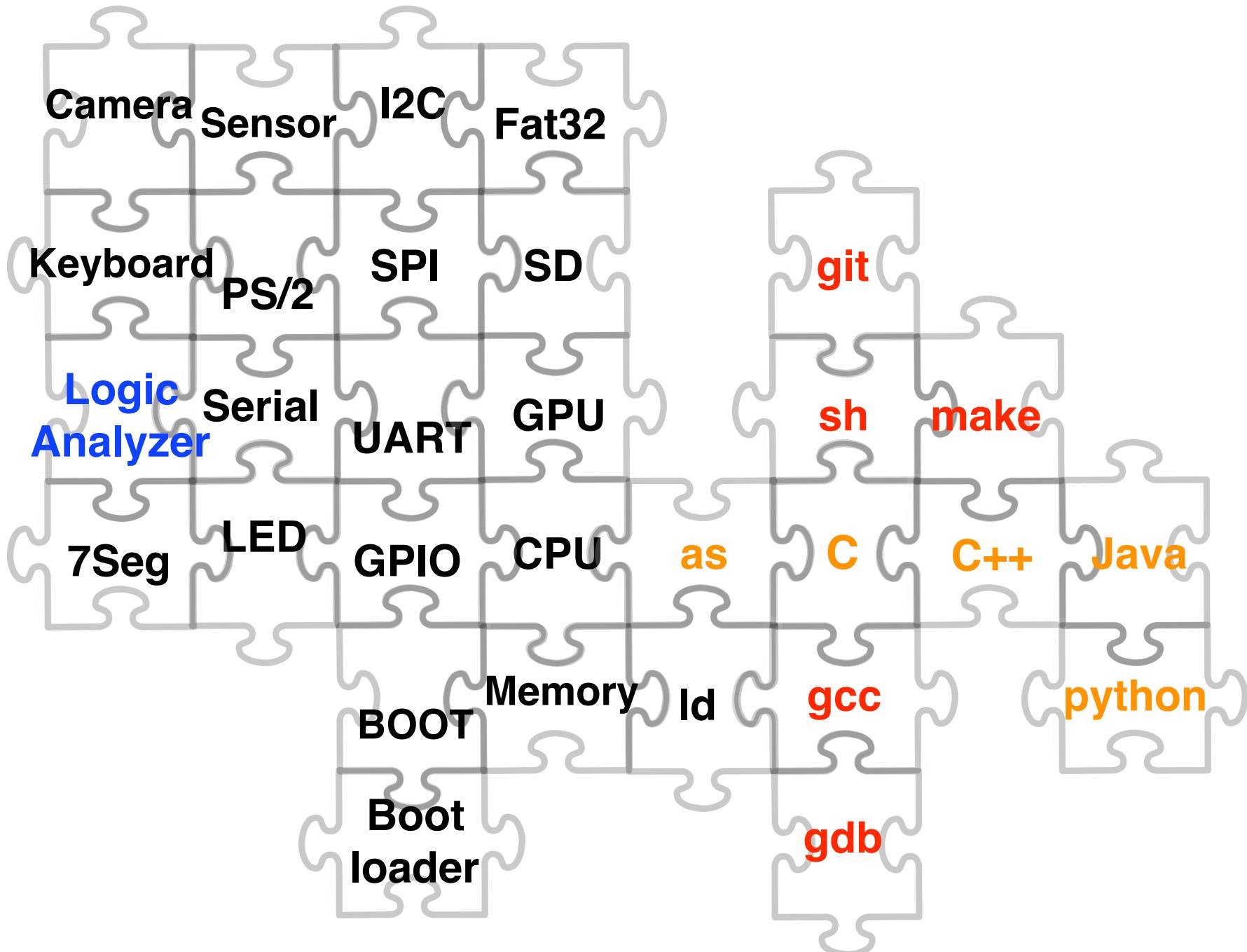


```
Raspberry Pi System Monitor  
*  
00008000- 07 DA A0 E3 1B 00 00 EB  
*  
00008008- B9 1B 00 EA 18 F0 9F E5  
*  
00008010- 18 F0 9F E5 18 F0 9F E5  
*7000  
00007000- 00  
*  
00007000- 00 00 00 00 00 00 00 00  
*  
00007008- 00 00 00 00 00 00 00 00  
*  
00007010- 00 00 00 00 00 00 00 00  
*7000:01 00 a0 e3 1e ff 2f e1  
*7000,7007  
00007000- 01 00 A0 E3 1E FF 2F E1  
*70000  
1  
*7000/7007  
e3a00001: mov r0, #1  
e12ffff1e: bx lr  
**
```



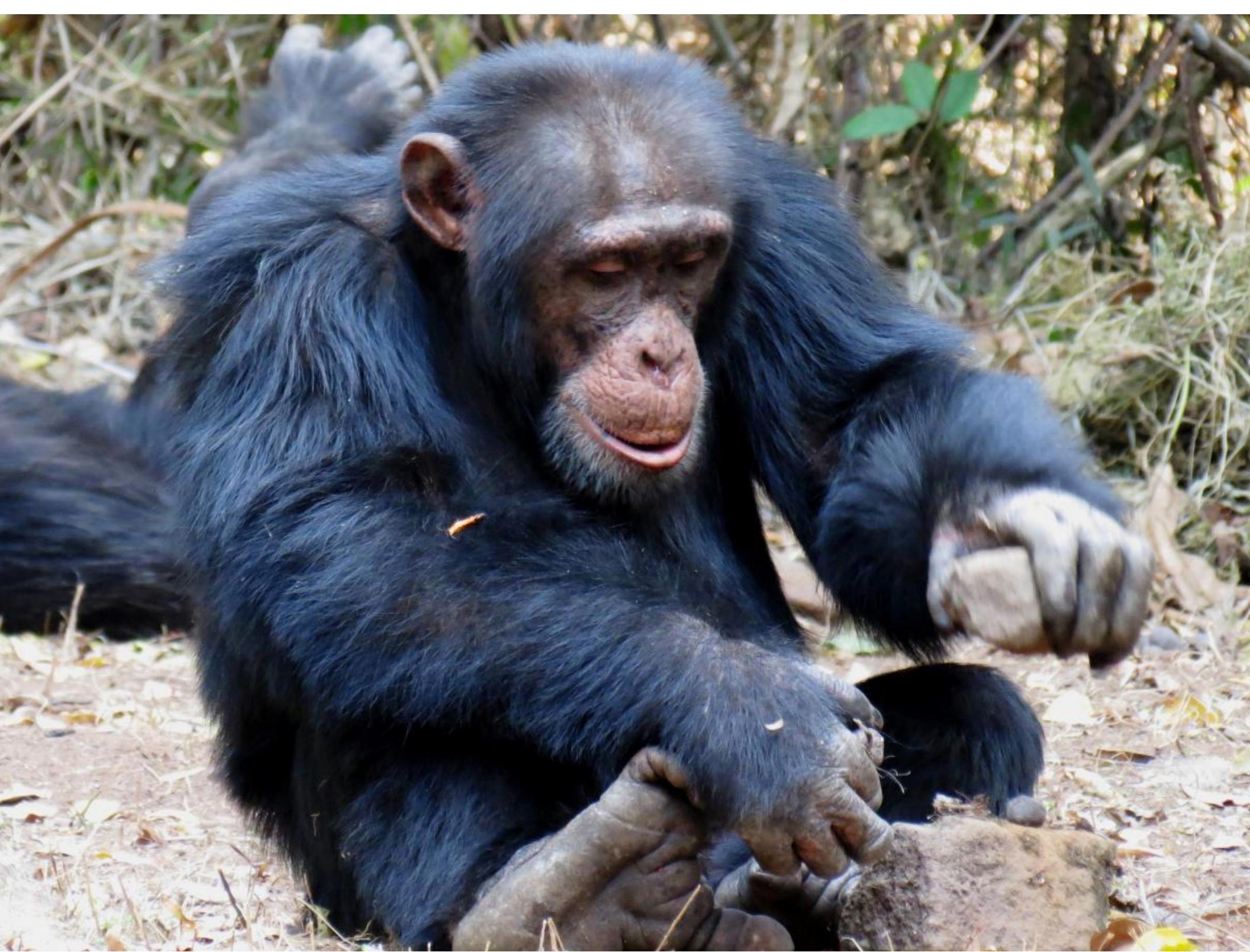
But It Keeps Growing!





Course learning goal #2:

Master your tools







The Essential Set of Skills

Debugging, Testing, and Troubleshooting

Engineering Habits

Always start from a known working state, stop in a working state. If it breaks, what changed?

Take a simple small step, check it carefully, then take another small step. Programming is much simpler if you figure out a good order to write the code!

Test, test, test, and test some more; Test as you go

Make things visible (printf, logic analyzer, gdb)

Systematic (D&C), not random, search for bug. Form hypotheses and perform experiments. Ask: why did my code do what I observed?

Don't be frustrated by bugs, relish the challenge, take frequent breaks



To invent, you need a good imagination and a pile of junk
Thomas Edison

Beyond bare metal...

Intersection of CS107 & CS107E

- **Hardware/software interface**
- **How programs execute (asm, stack, heap)**
- **C compiler and linker**
- **Bitwise manipulation**
- **Computer arithmetic**
- **Pointers, pointers, and more pointers**
- **Tools (unix, git, gcc, gdb)**
- **Lots of coding/debugging mileage**

Things unique to CS107E

- **Boot sequence, bootloader**
- **Interacting with peripherals**
- **C runtime startup**
- **Library internals**
- **Interrupts, supervisor mode**
- **(how to do everything yourself...)**

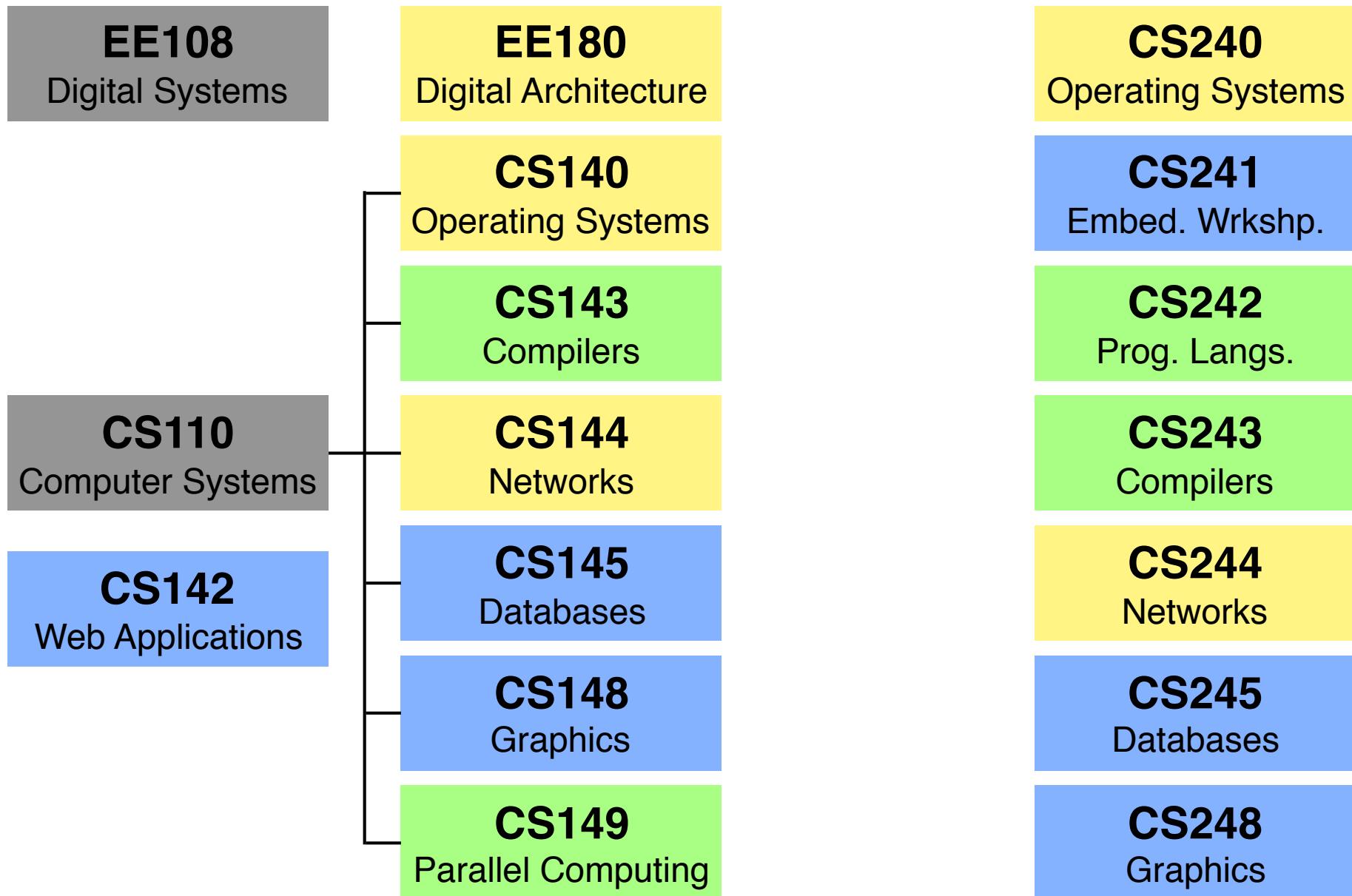
Things for you to learn

- **When not to do everything yourself**
- **What is available in system/C libraries**
- **Leverage tools (gdb, Valgrind, gprof, ...)**
- **Acclimate to user mode**

Opportunities!

- **Section leading**
- **CURIS**
- **Internships**
- **CS for Social Good**
- **Co-term**

Follow-on courses



Ask us anything...