

# CS140e – Embedded Operating Systems

**Instructor:** Dawson Engler

**CAs:** Holly Chiang

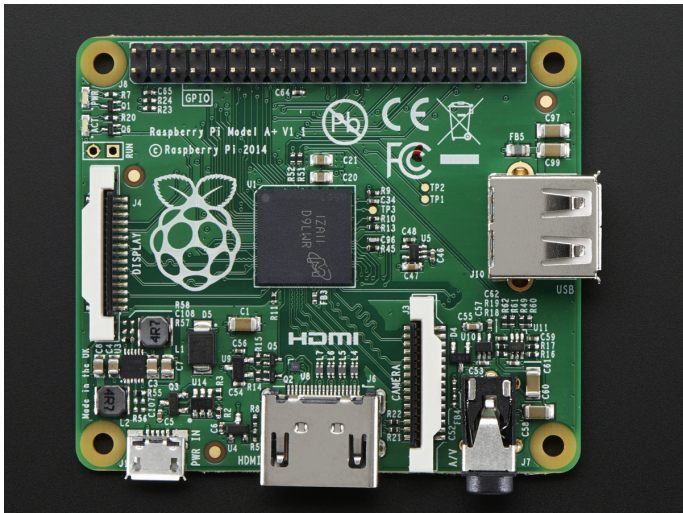
Stanford University

# Outline

- 1 What, Why
- 2 Administrivia

# What

- Write a small, clean operating system on the raspberry pi A+.
  - [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)





# What you will build

- **Bootloader.**
- **Threads.**
- **Virtual Memory.**
- **Simple file system on SD card.**
- **Final project: open ended. Expense account at [sparkfun/adafruit](https://sparkfun.com).**

# Why OS

- **If you can write a real OS, you can write pretty much anything.**
- **No abstractions: will understand (a) reality.**
  - Once you get this, easy to delta to other examples.
- **The real world is not a clean, textbook chapter.**
  - Difficult to understand documents.
  - Wrong.
  - Incomplete.
  - Not written to be used.
  - You will learn how to orientate and operate in such a world, without a lot of drama.
  - We chose r/pi A+ because lots of useful blog posts for how to do things. (Later pi's are less helpful).

# Why R/pi

- **Most OS classes (cs140) use a fake simulator.**
  - A lot of work. Not that cool at the end.
- **r/pi = real computer for about \$20.**
  - HDMI, SD card, memory: can put mouse, display, keyboard, have a lot of control.
- **Unlike most machines: Makes interacting with the real world very easy.**
  - Can build many interesting systems because can use weird hardware.
  - motion sensor, ir sensor, accelerometer, gyroscope, light sensor, etc.
- **Since bare metal: Very easy to build cool tools that are hard otherwise (gprof, eraser).**

# You will develop two super-powers

- **Differential debugging.** You write code, it doesn't work. Error could be:
  - The code you wrote;
  - Hardware fault (smoked something);
  - Wiring mistake;
  - Subtle cache issue;
  - Compiler problem;
  - ...
  - You will get good at breaking down problems to isolate.
- **Epsilon-steps.**
  - Engler's theorem: Given a working system  $W_k$  and a change  $c$ , then as  $c \rightarrow \epsilon$  then the time  $T$  it takes to figure out why  $W_n + C$  doesn't work goes to 0 ( $T \rightarrow 0$ ).
  - For a fixed amount of IQ, the smaller the step you can take from a working system, the faster you can debug when it doesn't work.



# Outline

- 1 What, Why
- 2 Administrivia

# Administrivia

- **Class web page:** <http://web.stanford.edu/class/cs140e/>
- **Textbook: *Operating Systems: Three Easy Pieces*,  
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau**
  - Is online: <http://pages.cs.wisc.edu/~remzi/OSTEP/>.
  - Also is cheap on amazon.
- **The main novelty: lectures will be blended with CS140 (Mazieres). Labs will be ours.**
  - Long-term goal is replace their labs with ours.
  - David's lectures give a good basis for the underlying principles.
  - Will give class credit for improvements you do for cs140e. (FAQs, useful documents, etc)

# Blending mechanics

- Most lectures during this time will meet in the CS140 classroom (Skilling 153). I will be at each one.
- We will then have labs after those lectures. We'll provide food often.
- **\*Tentative\* schedule:**
  - Monday: 3-4:20 Skilling for cs140. 5:30-7:30 mandatory lab.
  - Wednesday: 3-4:20 Skilling for cs140. 5:30-7:30 will go over all slides for the week, then lab time.
  - Thursday (optional): additional lab hours to help you.
- **Do not panic about workload: more in-class hours = less outside class hours.**
  - I expect your workload to be comparable to cs140.
  - But I expect the cs140e HWs will be more valuable :)
  - Can look at lab section as a review each week.
  - (Hopefully) guarantees you will not fall behind.

# Administrivia 2

- **Google group, TBA. Is main discussion forum**
- **Staff mailing list:** `cs140e-staff@stanford.edu`
  - Please use google group for questions other people might have
  - Otherwise, please mail staff list, not individual instructors
- **Key dates:**
  - Midterm exam: Wednesday, February 13, 3:00–4:20pm (in class)
  - Final exam: Monday, March 18th, 3:30pm-6:30pm
- **Exams open note, but not open book**
  - Bring notes, slides, any printed materials *except* textbook
  - No electronic devices permitted

# Grading

- **No incompletes**
  - Talk to instructor ASAP if you run into real problems
- **Final grades posted March 26**
- **60% of grade from projects**
  - For each project, 50% of score based on passing test cases
  - Remaining 50% based on design and style
- **25% of grade based on exams using this quantity:**  
 $\max(\text{midterm} > 0 ? \text{final} : 0, \frac{1}{2}(\text{midterm} + \text{final}))$
- **20% participation.**
- **If you do extra, will get extra. Projects will be open-ended. Class is open-ended.**
- **Most people's projects pass most test cases**
  - Please, please, please turn in working code, or **no credit** here
- **Means design and style matter a lot**
  - Software systems not just about producing working code

# Lecture videos

- **CS140 lectures will be televised.**
- **Go to lecture anyway.**
  - I will!
  - I will stay as long as needed to answer any questions you have after each class.

# CS140 Course topics

- **Threads & Processes**
- **Concurrency & Synchronization**
- **Scheduling**
- **Virtual Memory**
- **I/O**
- **Disks, File systems**
- **Protection & Security**
- **Virtual machines**
- **Note: Lectures will often take Unix as an example**
  - Most current and future OSes heavily influenced by Unix
  - Won't talk much about Windows