ECE 471 – Embedded Systems
Lecture 4

Vince Weaver

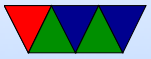
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

8 September 2021

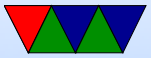
Announcements

- Any questions on HW#1?
- HW#2 will be posted Friday
- Let me know if you don't have a Raspberry Pi yet
- Office hours will be on Wednesday and Thursday

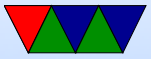


Current Events

- ARM released the Cortex-R82 processor
  64-bits, with MMU so can run Linux
- 5 times faster than previous Cortex-R8
- What would it be used for? Cortex-R widely used in storage devices. Can offload compute to storage?

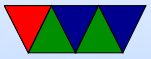


Raspberry Pi



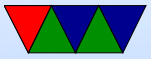
What is a Raspberry Pi?

- Raspberry Pi Foundation wanted small board to encourage CS in schools
- Easy to use and cheap enough that students can experiment without worrying too much about bricking it
- Back in the day small micro-computers encouraged hacking, modern Windows systems not so much
- There are other small embedded boards (BeagbleBone, etc.) but Pi is a nice combination of performance, cost, and available software



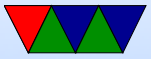
Raspberry Pi Software

- Can run many operating systems. Even write your own (see ECE598) or bare metal.
- We'll be running Linux.



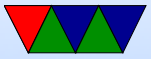
Why use a Raspberry Pi?

- There are many other single-board computers
- Pi has really good software support
  This really matters, many companies give you a blob of code with no support.
- It's also relatively well documented.



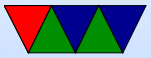
Raspberry Pi Models

- Model Names originally from BBC Micro
- All have more or less same SoC. VideoCore IV GPU runs show (VideoCore VI on pi4)
- First released in 2012

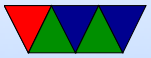


BCM2835/BCM2708 – ARM1176

- **Model 1B** – 700MHz ARM1176, 512MB RAM, SD, USB hub+USB Ethernet
- **Model 1B+** – like B but micro-SD, composite video-out inside of audio jack, 4 USB ports, longer GPIO header, re-arranged outputs, more mounting holes, fewer LEDs, lower power
- **Model 1A / Model 1A+** – less RAM (256MB/512MB), no Ethernet, no USB hub, cheaper, less power

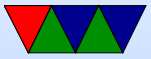


- **Zero** – 1GHz, 512MB, smaller, cheaper, $5
- **Zero W** – 1GHz, has wireless, $10
- **Compute Node** – like B but on SO-DIMM backplane, eMMC



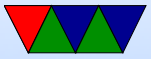
BCM2836/BCM2709 – ARM Cortex A7

- **Model 2B** – like 1B+ but with 1GB RAM, 900MHz Quad-core Cortex A7



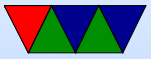
BCM2837/BCM2710 – ARM Cortex A53

- **Model 3B** – 64-bit, 1.2GHz Cortex A53, wireless Ethernet, bluetooth
- **Model 2B (v1.2)** – like Model 2 but with the Cortex A53
- **Compute 3**
- **Model 3B+** – better thermal, faster Ethernet (1GB but maxes at 300MB), power over Ethernet header. Still only 1GB (cost?)
- **Model3 A+**



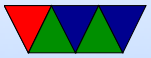
BCM2711 – ARM Cortex A72

- **Model 4B**
- 1.5GHz
- Videocore VI at 500MHz
- 1, 2, 4 or 8GB RAM
- USB3, microHDMI*2
- PCIe if you de-solder USB chip
- Real gigabit Ethernet
- GPIO header has more i2c/spi etc options
- **pi400** built into keyboard (4GB 1.8GHz)



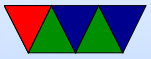
Pi Pico - RP2040

- Completely new design, custom SoC
- Dual core ARM-cortex M0+
- 133MHz
- 264k SRAM
- 2MB Flash
- $4



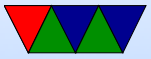
Programming the Pi

- We'll use C on Linux



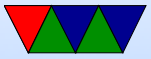
Why Linux?

- Open source
- Free
- Widely used for ARM-based embedded systems
- I like Linux.



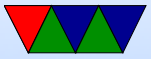
Brief Linux History

- UNIX history, UNIX lawsuit, rise of the BSDs
- Linus Torvalds (from Finland) gets a 386, announces his custom OS in 1991
- No free UNIX? FreeBSD caught up in AT&T lawsuit
- Don't be afraid of Linus (or open-source projects in general)
  The media over-hypes how angry some developers get.
- Just turned 30 years old



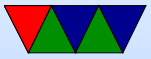
Why C?

- Portability (sort of) (i.e. how big is an `int`)
- Higher than assembly (barely)
  Pearce: "all the power of assembly with all the ease-of-use of assembly"
- Why over Java or C++?
  They can hide what is actually going on. C statements map more or less directly to assembly. With things like operator overload, and exceptions, garbage collection, extra layers are added. This can matter for both size,



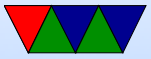
speed, determinism, and real time.
Might be restricted to a subset of C++

- Why over python?
  Mostly speed. (although you can JIT) Also if accessing low level hardware, in general you are calling libraries from python that are written in C anyway.
- What about Rust and Go? Don't overlook momentum of an old platform, sample code, libraries, etc.



Downsides of C?

- Undefined behavior. Compiler is allowed to do anything it wants (including dropping code) if it encounters something undefined by the standard. This can be something as simple as just overflowing an integer or shifting by more than 32.
- "Enough rope to shoot yourself in the foot". C gives a lot of power, especially with pointers. It assumes you know what you are doing though. With great power comes great responsibility.

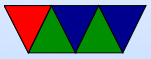


Downsides of C – Security

- Biggest issue is memory handling (lack thereof)
- Buffer overflows

```
int a[5];
a[0]=1;                 // fine
a[10000000]=1;          // obviously bad
a[5]=1;                 // subtly bad
```

- How can that go wrong? Crash? Corrupt Memory? Wrong results? Total system compromise?



- Overwriting stack can be bad, as return address there
- Especially if user input going into the variable