ECE 471 – Embedded Systems
Lecture 9

Vince Weaver

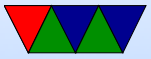
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

30 September 2014

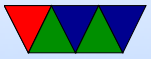
Announcements

- HW4 was posted a bit late, HW deadline is Friday
- Update your pi! Bash bug! Also assembler issues people were having. `apt-get update` ; `apt-get upgrade`



Update from Last Class

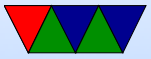
- Linux does not support things like pullups, but people have written code that will poke the relevant bits directly.



Homework 3

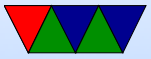
- Comment code! Fix wrong comments! My sample code had some out-of date comments (referencing "eax")
- print_number code. The divide by 10 code is almost more interesting. Good to be able to look at code and see what doing. Reverse engineering, but also debugging code you don't have the souce to.

```
print_number:
        push    {r10,LR}        // Comments removed for HW
        ldr     r10,=buffer     //
        add     r10,r10,#10     // why 10 bytes?

divide:
```



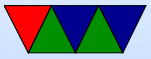
```
        mov     r1,#10          //

        bl      divide_by_10    // why no div instruction?
        add     r8,r8,#0x30     // why add 0x30?
        strb    r8,[r10],#-1    //
        adds    r0,r7,#0        //
        bne     divide          //

write_out:
        add     r1,r10,#1       //

        bl      print_string    //

        pop     {r10,LR}        //

        mov     pc,lr           //
```

- THUMB code should have been about 10 bytes less. Everyone saw longer results. My guess is debug info. If you run `strip` on your code to get rid of debug info you

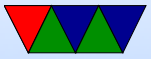


get the expected values.

- cal. Missing days. Julian to Gegorian calendar. People sad who paid weekly but paid rent monthly.



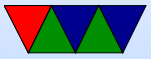
Booting a System



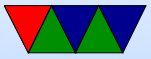
Firmware

Provides booting, configuration/setup, sometimes provides rudimentary hardware access routines.
Kernel developers like to complain about firmware authors.
Often mysterious bugs, only tested under Windows, etc.

- BIOS – legacy 16-bit interface on x86 machines
- UEFI – Unified Extensible Firmware Interface
  ia64, x86, ARM. From Intel. Replaces BIOS
- OpenFirmware – old macs, SPARC



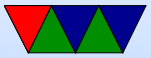
- LinuxBIOS



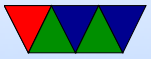
Boot Methods

Firmware can be quite complex.

- Floppy
- Hard-drive (PATA/SATA/SCSI/RAID)
- CD/DVD
- USB
- Network (PXE/tftp)

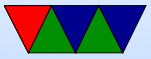


- Flash, SD card
- Tape
- Networked tape
- Paper tape? Front-panel switches?



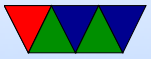
Disk Partitions

- Way to virtually split up disk.
- DOS GPT – old partition type, in MBR. Start/stop sectors, type
- Types: Linux, swap, DOS, etc
- GPT had 4 primary and then more secondary
- Lots of different schemes (each OS has own, Linux supports many). UEFI more flexible, greater than 2TB



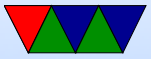
Bootloaders on ARM

- uBoot – Universal Bootloader, for ARM and under embedded systems
- So both BIOS and bootloader like minimal OSes

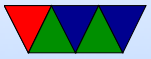


Raspberry Pi Booting

- Unusual
- Small amount of firmware on SoC
- ARM 1176 brought up inactive (in reset)
- Videocore loads first stage from ROM
- This reads `bootcode.bin` from fat partition on SD card into L2 cache.



- This runs on videocard, enables SDRAM, then loads `start.elf`
- This initializes things, the loads and boots Linux `kernel.img`. (also reads some config files there first)



More booting

- Most other ARM devices, ARM chip runs first-stage boot loader (often MLO) and second-stage (uboot)

- FAT partition
  Why FAT? (Simple, Low-memory, Works on most machines, In theory no patents despite MS's best attempts (see exfat))
  The boot firmware (burned into the CPU) is smart enough to mount a FAT partition