

# **ECE 598 – Advanced Operating Systems Lecture 3**

Vince Weaver

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

`vincent.weaver@maine.edu`

20 January 2015

# Announcements

- Let me know if you need to borrow a Pi



# Bootloaders on ARM

- The most common is uBoot
- uBoot – Universal Bootloader, for ARM and other embedded systems
- More of a challenge to write a bootloader for a widely nonstandardized architecture like ARM. (Why is ARM so nonstandardized?)



# Uboot 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



# Entering kernel ARM/u-boot

- vmlinux. strip. compress. piggy / piggyback.  
decompression code tacked on convert to zImage.  
mkuimage converts to uimage suitable for booting with  
u-boot
- No bios really. Bootloader provides all info.
- Modern day Device Tree provides config info for hardware  
(memory size, interrupts, what hardware is there)



# 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)



# 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





# So how do we start with own OS?

- Make simple binary.
- Compile it with ARM toolchain (cross compile?)
- Replace kernel.img on your memory card.
- Boot into it!
- Easier said than done.
- What kind of setup do you have?



# Setting up a Raspberry Pi

- Easy (if you have the hardware): HDMI monitor and USB keyboard
- Medium: network cable and ssh
- Hard: Serial cable to laptop/desktop



# Logging in

- Raspbian
- Password: pi / raspberry



# Raspberry Pi hardware

- 700MHz (overclock?), 1-issue in-order, VFPv2 (no neon), DSP
- 512MB RAM
- 16k 4-way l1 i/d cache, 128kb L2 controlled by vcore (linux configs for cpu)
- VideoCore IV (24Gflops) GPU
- BCM2835 SoC



- ARM1176 – v6, older than the v7 (Cortex A8, A9, A15)
- floating point?
- ARM32 and THUMB; no THUMB2



# BCM2835 features

- peripherals at 0x20000000 phys address
- uart and 2 spi
- bsc, aka i2c
- dma controller
- emmc
- gpio



- interrupts (mailbox, doorbell)
- pcm/i2s audio
- pwm
- spi
- spi/i2c slave
- system timer
- uart



- arm timer
- usb
- video? hdmi? composite? not fully documented but possible

