ECE 471 – Embedded Systems
Lecture 13

Vince Weaver

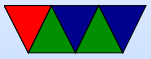
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

16 October 2014

Announcements

- Homework 6 will be posted this weekend.
- Midterm will be Thursday 23 October

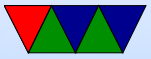


Homework 5 Review

- How do we know what bytes to send to an i2c device? Not standard, need to read manual for that device.

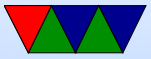
- 7a. GPU is used on boot on Pi. This is unusual, for the GPU to handle the booting. It's *not* unusual for an embedded board to have a GPU. A GPU lets you get by with a smaller CPU, as the CPU doesn't have to spend all its time updating the display.

- 7b. This question was poorly worded. What I was



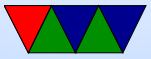
looking for was the bootloader, which is the chunk of code responsible for loading the kernel into memory. `init` is the program later run by the kernel responsible for getting userspace ready to go.

- 7c. fat32 is used primarily because it is simple and widely used, as well as mostly patent free.



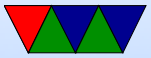
Real Time Constraints

- Time deadlines that hardware needs to respond in.
- Goal not performance, but response time



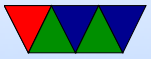
Types of Real Time Constraints

- Hard – miss deadline, total failure (people die?)
  Antilock brakes?
- Firm – result no longer useful after deadline missed
  lost frames in video, missed frames in video game
- Soft – results gradually less useful as deadline passes.
  Caps lock LED coming on?



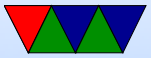
Constraints depend on the Application

- Hard or soft realtime can vary. Unlocking a car door taking an extra second?



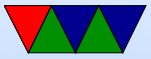
What can cause problems with real-time?

- Interrupts. Taking too long to run; being disabled (cli)
- Unpredictable nature of modern CPUs. Caches, branch-predictors, etc.
- Operating system. Scheduler.



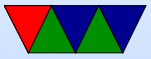
Determining worst case behavior.

- Hard on modern processors. Easier on stm32l than on raspberry pi.

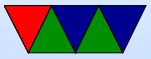


Scheduler example

- Static: Rate Monotonic Scheduling – shortest job goes first
- Dynamic: Earliest deadline first
- Three tasks come in. a. finish in 10s, 4 long. b. finish in 3, 2 long, c. finish in 5, 1 long
- In order they arrive, aaaabbccc bad for everyone
- RMS: cbbbaaaa works

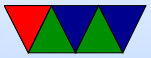


- EDF: bbbcaaaa also works.
- Lots of information on various scheduling algorithms



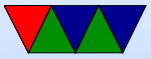
Priority Inversion Example

- Task priority 3 takes lock on some piece of hardware
- Task 2 fires up and pre-empts task 3
- Task 1 fires up and pre-empts task 1, but it needs same HW as task 3. Waits for it. It will never get free.
- Space probes have had issues due to this.



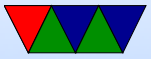
Real Time OS

- Who uses realtime?
- realtime used by musicians, important to have low-latency when recording
- High-speed trading



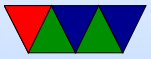
PREEMPT Kernel

- Linux PREEMPT_RT
- Faster response times
- Remove all unbounded latencies
- Change locks and interrupt threads to be pre-emptible



Linux PREEMPT Kernel

- What latencies can you get? 10-30us on some x86 machines
- Depends on firmware; SMI interrupts (secret system mode, can't be blocked, emulate USB and like)' Slow hardware; CPU frequency scaling; nohz
- Special patches, recompile kernel
- mlockall() memory in, start threads and touch at beginning, avoid all causes of pagefaults.



Co-operative real-time Linux

- Xenomai
- Linux run as side process, sort of like hypervisor