ECE 471 – Embedded Systems Lecture 21

Vince Weaver

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

vincent.weaver@maine.edu

29 October 2018

Announcements

- Project coming
- Don't forget SPI homework HW#7

HW#6 Review

- Questions:
  - Brakes – hard real time?
  - Tuner – soft real time
  - Video – firm real time
  - Interrupts. Doorbell.
  - Yes command – mostly to answer things like fsck that ask a lot of obvious questions.
    Load testing, maybe, but that wasn't really the original design.

PREEMPT Kernel

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

Typical kernel, when can you pre-empt

- When user code running
- When a system call or interrupt happens
- When kernel code blocks on mutex (lock) or voluntarily yields
- If a high priority task wants to run, and the kernel is running, it might be hundreds of milliseconds before you get to run

- Pre-empt patch makes it so almost any part of kernel can be stopped (pre-empted). Also moves interrupt routines into pre-emptible kernel threads.

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, etc.)
  Slow hardware; CPU frequency scaling; nohz
- Special patches, recompile kernel
- Priorities
  - Linux Nice: -20 to 19 (lowest), use nice command
  - Real Time: 0 to 99 (highest)

- Appears in ps as 0 to 139?

Changes to your code

- What do you do about unknown memory latency?
  - mlockall() memory in, start threads and touch at beginning, avoid all causes of pagefaults.
- What do you do about priority?
  - Use POSIX interfaces, no real changes needed in code, just set higher priority
  - See the `chrt` tool to set priorities.
- What do you do about interrupts?
  - See next

Interrupts

- Why are interrupts slow?
- Shared lines, have to run all handlers
- When can they not be pre-empted? IRQ disabled? If a driver really wanted to pause 1ms for hardware to be ready, would often turn off IRQ and spin rather than sleep
- Higher priority IRQs? FIR on ARM?
- Top Halves / Bottom Halves
- Unrelated, but hi-res timers

Co-operative real-time Linux

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

Other RTOSes

- Vxworks
- Neutrino
- Free RTOS
- Windows CE
- MongooseOS (recent LWN article?)
- ThreadX (in the Pi GPU)