ECE 471 – Embedded Systems
Lecture 19

Vince Weaver

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

vincent.weaver@maine.edu

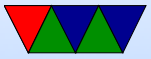
16 October 2017

Announcements

- Issues on HW#6

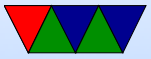
  Be sure you remember not to drive GPIO high, let float high

  Use the predefined code.

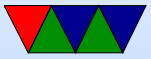


HW#5 Review

- Coding: Seemed to go OK
  Error handling – exit, not write to not opened fd
  Print all 1's meant 88:88 not 11:11
- Comments: `buffer[0] = (0x2 << 6) | 0x1;`
- Raspberry Pi boot odd: GPU does it
- Fat32: gave lots of good reasons for Fat32, but the reason boot partitions often use it is it's simple enough to be read by firmware at extreme early boot.

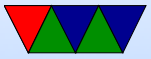


- Program that loads kernel and jumps to it is called the bootloader
- Skip i2c – those addresses are reserved
- wc, diff, piping



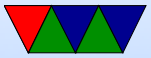
Real Time Operating System

- Can it be hard real time?
- Simple ones can be mathematically provable
- Otherwise, it's a best effort

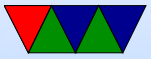


Priority Based, like Vxworks

- Each task has priority 0 (high) to 255 (low)
- When task launched, highest priority gets to run
- Other tasks only get to run when higher is finished or yields
- What if multiple of same priority? Then go round-robin or similar

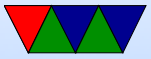


Real Time Linux



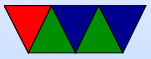
PREEMPT Kernel

- Linux PREEMPT_RT
- Faster response times
- Remove all unbounded latencies
- Change locks and interrupt threads to be pre-emptible
- Have been gradually merging changes upstream



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.