ECE 471 – Embedded Systems
Lecture 17

Vince Weaver

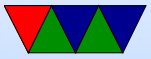
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

11 October 2024

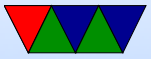
Announcements

- Midterm on Friday, the 18th, in class
- Am trying to grade all homeworks by then
- HW#6 will be posted, have two weeks
- Hang on to displays, needed for HW#9



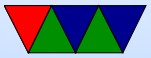
Midterm Notes

- The midterm will be in-person during class time
- Closed book/notes but you are allowed one page (8.5"x11") full of notes if you want

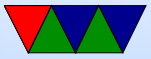


Midterm Content

- Be sure you know the characteristics of an embedded system, and can make an argument about whether a system is one or not.
  - Inside of something (embedded)
  - Fixed-purpose
  - Resource constrained
  - Sensor I/O
  - Real time constraints (if you use this, be sure you can explain)

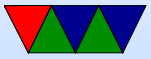


- Benefits/downsides of using an operating system on an embedded device
  - Benefits: "Layer of Abstraction"
  - Downsides: overhead, timing
- C code
  - Have you look at some code and know what it is doing
  - Fill in missing comments
  - Look at code and find bugs
  - Mostly know what file I/O, loops, usleep, open/ioctl (things we've done in the homeworks)
- Code Density



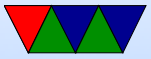
- Why is dense code good in embedded systems?
- Know why ARM introduced THUMB/THUMB2

- GPIO & i2c
  - Know some of its limitations (speeds, length of wires, number of wires, etc)
  - Don't need to know the raw protocol
  - Know the Linux interface (open, ioctl, write) and be familiar with how those system calls work
- Realtime won't be on this midterm

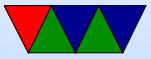


Project Preview

- Posted a PDF with full details to the website
- Can work in groups
- Embedded system (any type, not just Pi)
  Pi Pico, Beagleboard, Orange Pi, 271 STM boards, TS-7600, etc.
- Written in any language (asm, C, python, C++, Java, Rust, etc.)
- Do some manner of input and some manner of output using the various capabilities we discussed

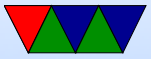


- I have a large amount of i2c, spi, and other devices that you can borrow if you want to try anything interesting.
- Past projects: games, robots, weather stations, motor controllers, music visualization, etc.
- Note: this will have to be distinct from your ECE498 or senior projects
- Will be a final writeup, and then a short minute presentation and demo in front of the class during last week of classes.
- Deadlines:
  - November 8: pick topic (send e-mail with group



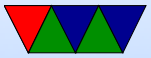
members and preliminary topic)

- November 25: progress report (e-mail with summary, how it's going, and which day you'd prefer to present)
- Last week of classes: project presenations
- December 20: Writeup due

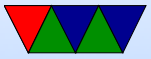


Homework #4 Error Checking

- What do you do if there's an error?
- Ignore it? Why could that be bad?
- Retry until it succeeds?
- Print an error message and continue?
  Can you continue?
  What if continuing with a bad file descriptor breaks things?
  What if printing too many error messages fills up a log, swamps the screen, hides other errors?

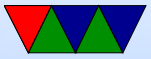


- Good error message
  Can't be confused with valid input (airlock)
  If displayed to user, make it easy to understand
- Print an error message and exit?
  What if it's a critical system?
- Crashing is almost never the right answer.
- Can get more info on error with errno / strerror()

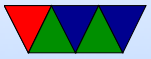


Homework #4 Permissions

- We haven't really discussed Linux permissions
- List file, "user" "group" "all"
- `drwxr-xr-x`
- Often in octal, 777 means everyone access
- Devices under /dev or /sysfs might be set to only root or superuser
- Traditional UNIX /dev you can set with chown (to set user/group) or chmod (to set permissions)
- Group under /etc/group, so gpio group



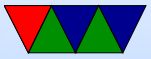
- Why is it better than using “sudo”? Why might I as grader not want to run your code using “sudo” if I can avoid it?
- How to set up sudo? /etc/sudoers file



Homework #4 – LED Blinking

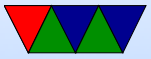
- Blink frequency. Remember, 1Hz is 500ms on / 500ms off

  not 500us, not 1s
- Blink correct GPIO. Does it matter? Want to fire engines, not engage self destruct.



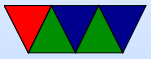
Homework #4 – Switch

- Debouncing
  - 100ms or even 10ms is long time
  - Tricky as we are detecting levels not edges here
  - Reading and only reporting if you say have 3 in a row of save val
  - Reading, sleeping a bit, then report the value after has settled
  - Just sleeping a long time after any change? If a short glitch happens this might misreport.



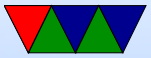
- Sleep too long, might miss events
- Debounce if using interrupt-driven code

  In that case debouncing might be to ignore repeated changes if they happen too close together



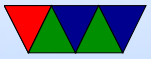
Homework #4 – Something Cool

- How can you read/write at same time (say to let switch activate LED)
- Need to make copy of data structures
- If you do re-use, make sure you close(), especially if you open multiple times. Either will get EBUSY or else fd leak



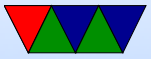
Homework #4 Question – usleep()

- Less resources (not busy sleeping),
- cross-platform (not speed-of-machine-dependent)
- compiler won't remove
- other things can run,
- power saving
- Be careful saying accuracy! usleep() guarantees a minimum time delay, but it is best effort how long the delay actually is. So if you really need *exact* time delays you probably want some other interface.



Homework #4 Question – OS

- provides layer of abstraction
- In this case, not having to bitbang the interface or know low-level addresses, portability among machines.

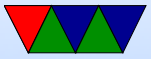


Homework #4 Question – Linux stuff

- 6.a Machines from dmesg: 2023: Pi4 (12) Pi3B+ (9) Pi3B (1) Pi2B (3) dmesg a good place to find error messages, etc. grep
- 6.b Kernel versions. Current Linus kernel (upstream) is 6.0
  Uname syscall, what the parts mean

```
Linux linpack-test 4.14.50-v7+ #1122 SMP Tue Jun 19 12:26:26 BST 2018 armv7l GNU/Linux\\
Linux orvavista 4.5.0-2-amd64 #1 SMP Debian 4.5.5-1 (2016-05-29) x86_64 GNU/Linux\\
```

  2023: 6.1.62 (1) 6.1.21 (18) 5.15 (2) 5.10(1) 64-bit (7)
- 6.c. Disk space. Why `-h`? Human readable. what does



that mean? Why is it not the default? At least Linux defaults to 1kB blocks (UNIX was 512) Lots of large disks.