ECE 471 – Embedded Systems
Lecture 11

Vince Weaver

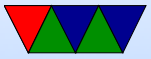
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

4 October 2016

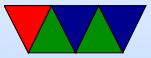
Announcements

- Midterm is Thursday.



Homework #4 Review

- Review of `read()` and `write()`:
  - File descriptors are integers, not FILE *. Be sure to not ignore compiler warnings!
  - `result=write(file_descriptor, buffer, length)`
    buffer is a pointer to the source of the values to write
    arrays are sneaky because they are secretly pointers so you don't need the &
    Be sure you use the proper length



(many people cut-and-paste "in", changed to "out" but forgot to update the length to be 3 instead of 2)

- `result=read(file_descriptor, buffer, length);`

  again buffer is a point to where the data goes

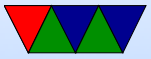
  to get the value out, you'll want to look at buffer[0] or similar, and *remember it's ASCII*

  If you just compare against buffer, what do you get?

  The pointer value, usually not what you want

- Control-C and what it does.
- Error checking! Silent fail is bad.

  What to do on an error?

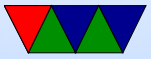


Print message? Exit? Try again?

Charging on through with invalid fd is not a good answer likely. Also if fails can't write to invalid fd.

You get an error, you should probably clean up/free anything that is open and then exit, rather than just printing an error and pushing on. In some cases though pushing through can be OK (GPIO export sometimes gets stuck).

- If printing, be sure to have linefeed (slash-n) on end. Otherwise will be buffered and not appear.
- Don't ignore compiler warnings! Were real bugs!



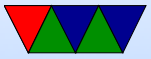
Initialize variables, if a local (stack) variable it might only be luck that you start out at 0.

- `fopen()` issues
  - Most of the issues come with buffering getting in the way.

    You need to run `fflush()` if not closing the file (closing always flushes the buffers)

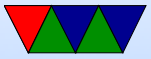
    linefeed is apparently not enough to flush buffer

    Also need `fflush()` when reading to re-load value from backing file.
- For the switch question, wanted it to print on press and



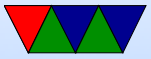
on release. Have a state bit that remembers last state. Remember the value you read is the ASCII value of result.

- Debounce
  If waiting, need to check again after the wait.
  Was it needed in practice?
  Comment your code about how you debounce, makes grading easier.
  One way is to when changes, sleep a small amount (1ms?) and re-read and see if the value is the same. Keep reading until get same twice in a row.



Also saw just once it changes, sleep some, in the assumption the first change is right and other spurious. Simply sleeping not enough, what if was a a spike?

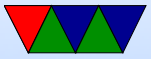
- Questions
  - 5.a Why usleep? Not udelay, how come no one caught this past three years? Less resources (not busy sleeping), cross-platform (not speed-of-machine-dependent), compiler won't remove, other things can run, power saving
  - 5.b Layer of abstraction. In this case, not having to bitbang the interface or know low-level addresses,



portability among machines. Note: You can use high-level languages w/o an OS.

- 5.c Limitations : higher overhead, not all features exposed, uncertain timing.
  superuser permissions? when no OS you run everything as super user, though this depends on HW and is complicated.

- 5.d. Web browser part of OS? Microsoft law suit.
  Interesting comments on google/chrome
- 6.a Machines from dmesg: Zero (1) B (2) Bplus (2)

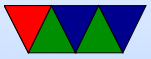


Pi2ModelB (12), Pi3 (6) dmesg a good place to find error messages, etc.

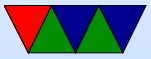
- 6.b Kernel versions. Current Linus kernel (upstream) is 4.7/4.8-rc8
  Uname syscall, what the parts mean

```
Linux rasp-pi 4.1.19+ #858 Tue Mar 15 15:52:03
Linux orvavista 4.5.0-2-amd64 #1 SMP Debian 4.
```

3.18.11 (1) 4.1.13 (2) 4.1.18 (2) 4.1.19 (5) 4.1.21(1) 4.4.1 (1) 4.4.9 (1) 4.4.11 (6) 4.4.13 (1) 4.4.20 (1) 4.4.21 (1)

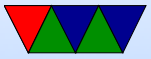


- 6.c. Disk space. Why `-h`? Human readable. what does that mean? Why is it not the default? At least Linux defaults to 1kB blocks (UNIX was 512) Lots of large disks.



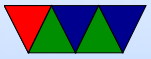
C String Review

- What is a C string? An array of chars, nul-terminated
- strcpy, strcat
- strncpy, strncat
- sprintf



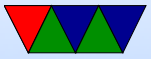
Midterm Review

- Be sure you know the four 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
  - Real time constraints
- Benefits/downsides of using an operating system on an embedded device

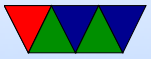


- Cost, time to market, helper libraries, overhead, timing

- ARM assembly language
  - Have you look at some assembly language code and know what it is doing
  - Only really need to know some of the more common instructions (add, cmp, mov, ldr, strb, swi). Also be aware of conditional execution.
- Code Density
  - Why is dense code good in embedded systems?
  - What changes were needed to ARM32 to make it fit into 16-bit THUMB?



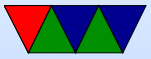
- i2c
  - Know some of its limitations (speeds, length of wires, number of wires, etc)
  - Don't need to know the raw protocol
  - Know the Linux interface (open, ioctl, write) and be familiar with how those system calls work



Stuff I missed last time

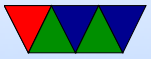
- How do you hook up multiple displays? Often i2c interface chips will have pins you can short to change the low-order address bits. Solder pads.

- insmod vs modprobe vs lsmod vs dmesg vs rmmod



Why is Linux used in Embedded Systems?

- Linux popular in embedded space because it is cheap/free and source code is available.
- Linux on ARM is widely supported (although upstream support is a mess)
- Licensing issues
  - Linux under GPLv2.
  - The Free Software Foundation has moved most of its software (including gcc compiler) to the less popular



GPLv3 which most companies don't like.

- Companies often prefer BSD type license which has fewer restrictions; companies can use code and release binaries without having to release the source (a GPL requirement).
- Apple and Google both trying to replace as much code as possible with BSD versions.