ECE 471 – Embedded Systems
Lecture 11

Vince Weaver

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

vincent.weaver@maine.edu

6 October 2015

Announcements

- Midterm is tentatively scheduled for October 22nd.
- Homework #5 issues?

Homework #4 Review

- Review of `read()` and `write()`:
  - `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.

- `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, compiler won't remove, other things can run
  - ○ 5.b Layer of abstraction. In this case, not having to bitbang the interface or know low-level addresses, portability among machines.

- 5.c Limitations : higher overhead, not all features exposed, uncertain timing
- 5.d. Web browser part of OS? Microsoft law suit. Interesting comments on google/chrome
- 6.a Machines from dmesg: BCM2708 (1), Model B (4), Model B+ (5), 2 Model B (7)

  dmesg a good place to find error messages, etc.
- 6.b Kernel versions. Current Linus kernel (upstream) is 4.2 / 4.3-rc4

  3.6.11 (1), 3.18.7 (3), 3.18.11 (8), 4.0.7 (1), 4.1.6 (2), 4.1.7 (3)

- 6.c. Disk space. Why `-h`? Human readable. what does that mean? Why is it not the default?

C String Review

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

Stuff I missed Last Time

i2c Reserved Addresses

Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode master code
111 10XX	X	10-bit slave addressing
111 11XX	X	Reserved for future purposes

10-bit addresses work by using special address above with first 2 bits, then sending an additional byte with the lower 8 bits.

Other

- I got the Linux address shifting backwards. It's 0xE0 to 0x70 rather than 0x38 to 0x70

- 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.