ECE 471 – Embedded Systems
Lecture 29

Vince Weaver

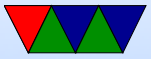
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

14 November 2022

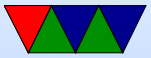
Announcements

- HW#9 was posted
- News: Tesla had recall because firmware update caused power steering to fail when go over bump
- Midterm on Friday
- Don't forget projects. 23 groups.



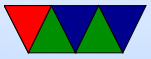
HW#8 / Linux 1-wire note

- Why not use “temperature” file?
- Had to look in linux source code, w1_therm.c
- git log / git bisect
- Added May 11 2020, first appeared Linux 5.8
- Not even the main feature, as an afterthought
- Better fits the sysfs philosophy
- Any reason not to use it? What if kernel version too old...

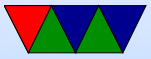


HW#9 – Modular Code

- What is modular code?
- In C you can compile each C file into its own object file, link together at end
- Each C file can act as self-contained module
- Can make code-reuse easier, can treat like a library
- API defined in a header .h file



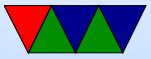
- Can be easier to follow code than in one huge program
- Easier when working in shared git repository



HW#9 – Converting Floating Point to Digits

- Re-use i2c display code from earlier homework
- Re-use temp code (either TMP36 or the 1-wire)
- Display the temperature on display
- Code coverage, Writing good testable code
- Converting a double to characters to print
  - Use sprintf()

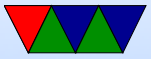
```
char string[128];
double temperature;
sprintf(string,"%.1lf",temperature);
/* Now string[0] has first digit, string[1] second, etc */
```



- Use division/modulus

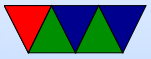
```
double temperature=23.4;
int hundreds,tens, ones,remainder;

hundreds=temperature/100;
remainder=temperature%100;
tens=remainder/10;
ones=remainder%10;
```

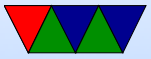


HW#9 Notes – Building Separate Files

- In previous homeworks we put everyting in one C file
- This isn't really practical for large projects
- By splitting things up into smaller files you can have some benefits
  - Easier to organize/find code
  - Can re-use code easier
  - Less chance of merge conflicts when multiple people working on project
  - Can take common code and make libraries

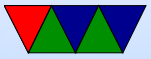


- For example in the homework, we could put temperature read code into its own file with a `double get_temperature(void)` interface
- For other C files to see this, you need to export the definition. Usually this is done by putting the advance definition `double get_temperature(void);` in a .h header file and then including it in the other files
- Note: don't put full C functions in header files. I know this is a C++ thing but it's usually frowned upon when programming in C
- Each file does not need a `main()` function, you only



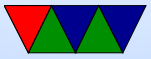
need one per combined program.

- To link the various .o files together involves the "linker". However it's easier to just let gcc do it (gcc knows how to run the linker for you) `gcc -o display_temp display.o temperature.o`
- The linker merges the .o files into one big executable, and makes sure the placeholders to functions/variables in all of the files get the right addresses/pointers to where things live in the finished executable.
- How do you make sure when you change one C file that everything that uses it is also rebuilt? A well-crafted



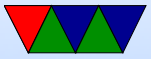
Makefile will have all these dependencies in place and will rebuild everything properly.

- Static vs Dynamic linking redux (we did discuss that earlier in the semester)

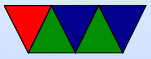


Midterm #2 Preview

- Booting on the Pi
  - What a bootloader does
  - Why Pi is unusual
- Real Time
  - Definitions
  - Is this hard, soft, firm
- i2c/SPI/1-wire
  - Know the tradeoffs between i2c, SPI, 1-wire
  - Be able to follow the C code for them

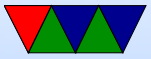


- Security
  - Buffer overrun, why it is bad
- Coding Practices
  - Be aware of the case studies we suggested
  - Know of some of the recommended ways to write safer C code



HW#6 Review

- What is a "safe" amount of overhead? 10% of max observed? Twice max observed?
- As saw when under load there might not be any guarantees
- Using real-time priorities helped a lot
- Why not let regular users be able to specify this?
- People handled hard/soft/firm realtime fine



Finished up the parts we didn't finish from Lecture 28