

ECE 471 – Embedded Systems

Lecture 18

Vince Weaver

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

`vincent.weaver@maine.edu`

15 October 2025

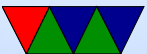
Announcements

- Don't forget RT homework HW#6 (finally posted)
- Midterm exam Friday, in class!
- Working on getting HW grades out. Managed to crash my grading Pi when testing HW#6.
- RT related: SpaceX second stage trouble, engine burned 500ms more than should have



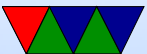
Midterm Notes

- Detailed list of material on midterm can be found in Friday's notes
- Closes notes except you are allowed 1 8.5" x 11" piece of paper
- I prefer hand-written, and for the notes to be your own. I'm not banning typed notes though.



Notes on HW#4

- Please don't cheat
- Please try to get assignments in on time
- Don't ignore compiler warnings
- Took forever trying to grade switch debouncing, gave up in end, need some sort of automated test
- What you **definitely** shouldn't do is sleep for 100ms or more, that's long enough that a normal person pressing buttons will see lost keypresses



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?

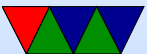
Can you continue?

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: 2025: Pi5 (5) Pi4 () Pi400 () Pi3B+ (1) Pi3 (2)

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\\
```

2025: 6.12 (6) 6.6 (2)

- 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.

Note this is another thing that changed, and a pain to keep up with. Asked for “rootfs” which it used to say but doesn’t anymore. Instead on Linux your “root” filesystem is the one mounted as “/”



HW#5 – Code Notes – Datasheet

- What does 'X' mean in this context? (don't care)
- Bits 15-8 was confusing, it's because we can ignore bits 7-0 (the i2c address and r/w) as Linux sends those for us



HW#5 – Code Notes – Constants

- Enabling oscillator. If want value 2 in top 4 bits, 1 in bottom 4? Just use 0x21?

`(0x2<<4) | (0x1)`

- Can we use hex or binary notation?

The shifts make it more explicit what's going on, compiler will optimize for you

- “Magic Constants”, you might instead want to do something like

```
#define HT16K33_OSCILLATOR_ON (0x2<<4)|(0x1) // p42 of datasheet
buffer[0]=HT16K33_OSCILLATOR_ON;
```



HW#5 Review – Questions – Why Use OS

- Why use OS?
- Why not?



HW#5 Review – Questions – i2c Reserved Address

- Skipped i2c – those addresses are reserved.
- For various things, not just “future purposes”
- What happens if you have a device living at address 0x0?
Would it work?



HW#5 Review – Linux

- wc, diff, piping

```
cat x.txt | sort | uniq | wc -l
```

- You may have seen this all before in ECE331
- diff – used when making patches, also git diff
Ask for wc -l which just shows lines. Can also show words, chars
- These days diff/patch more or less obsoleted by git pull requests



i2c Reserved Addresses Reminder

Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte (helps make polling cheaper)
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 + R/W, then sending an additional byte with the lower 8 bits.



Where do RT deadlines come from?

- Often from hardware, or from the spec, or laws of Physics
- Can you change them? Maybe, but might involve hardware changes
- Examples
 - Write to device, need to do all 4 memory accesses within 100us
 - Car notices brakes locking, start pumping them within 100ms
 - Drop laptop, notice and park hard-drive within 100ms



Priority Based Scheduling

- It's actually rare for an OS to let you specify a deadline
- Usually instead they are priority based (like in HW#6)
 - Have multiple tasks running, assign priority
 - In previous example, B highest, then C, then A
 - B can pre-empt C and A
- What can happen if overcommit resources? Starvation



IRQ	-	-	-	-	-	I	-	-	-	-
HIGH	-	-	-	-	B	-	B	-	-	-
MEDIUM	-	-	C	-	-	-	-	-	-	-
LOW	A	A	-	A	-	-	-	A	-	-
OS	!	!	!	!	!	!	!	!	!	!



Priority Inversion

- Task priority 3 takes lock on some piece of hardware (camera for picture)
- Task 2 fires up and pre-empts task 3
- Task 1 fires up and pre-empts task 2, but it needs same HW as task 3. Waits for it. It will never get free. (camera for navigation?)
- Space probes have had issues due to this.



Priority Inversion – What can you do about it?

- User code – be very careful writing code and taking locks, sharing resources (always do this, but be extra careful when different priorities involved)
- Operating system – have scheduler make sure all processes (even low priority ones) make occasional forward progress
- Hardware – have watchdog timer that always triggers and can check to see if system has frozen



HW#6 Preview

- Two weeks to do this one (due to Fall break and midterm)
- Involves testing out real-time / latency response on stock Raspberry Pis
- Connect a GPIO output pin to GPIO input. Raise the output in one thread, see how long it takes another to notice.
- On an idle system might get OK behavior, but that might change when under load



- Have you try out adding load, as well as messing with Linux Real Time priority support.
- There's a chance that messing with priorities can crash your system. If so just reboot and follow the directions in the assignment.

