ECE 598 – Advanced Operating Systems
Lecture 21

Vince Weaver

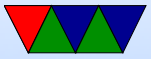
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

16 April 2015

Announcements

- HW#6 was due, HW#7 due Friday

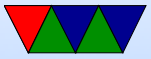


HW#6 Review

1. Benefit of a system call: fast
   Downside: need to update interfaces, can't be in module, new code cannot depend on it

2. HDMI display

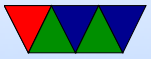
   (a) Bandwidth of display: 4096x2160x32bppx60Hz=17.0Gbps
   Complicated, used 8b/10b conversion for various reasons. Assume you can ignore this.
   36 and 48-bit color?



(b) HDMI 1.0 165MHz/channel
(c) HDMI 1.3
(d) HDMI 1.4 340MHz/channel = can do 4096x2160x24Hz
(e) HDMI 2.0 600MHz/channel * 3 = supports 18Gbps
    (for reference, Rasperry Pi is HDMI 1.4)
(f) One of reasons for HDMI 2.0 was for 4k displays

3. Mailbox

(a) Only top 28 bits sent, so has to be in multiple of 16 bytes if you pass an address.
(b) In gcc you can use `__attribute__ ((aligned(16)));;`



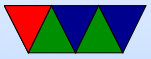
4. Framebuffer

   framebuffer points to an 800x600x24bpp RGB framebuffer?

```
void do_something(int x, int y1, int y2, char *framebuffer) {
    int y;
    for(y=y1;y<y2;y++) {
        framebuffer[(y*pitch)+(x*3)+1]=255;
    }
}
```

   three dimensional array framebuffer[x][y][3];

   `vline_green(int x, int y1, int y2, char *frameb`

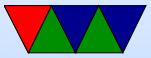


Homework 7

- Don't put it off!

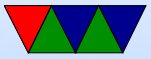
```
kernel_main.c:(.text+0x2e4): undefined reference to `__aeabi_idiv'
```

- ARMv6 has no divide instruction. What can you do about this?
- Link against libgcc.a (maybe)
- Why isn't this always a problem?
  If a constant, the compiler can cheat. Reciprocal multiply, shift if power of 2, etc.



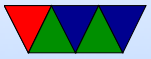
- Easier to hit if dividing by a variable so it can't fix at compile time.

- Same issue with mod operator %
  If power of 2 can implement with an and instruction



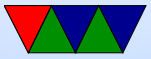
Misc

- adding multithreading to code?
  How to handle global variables (errno?)
  Thread-safe functions. Is strtok thread-safe? malloc? any routine that might not be re-entrant
  How are multiple stacks handled? One option each thread gets own copy of global variables. This can't be expressed by default in C, you need special routines, thread-local variables.

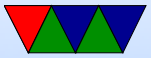


IPC – Inter-Process Communication

- Processes want to communicate with each other. Examples?
- Two issues:
  getting the message across
  synchronizing
- signals
- network, message passing (send, receive)



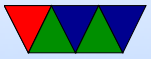
- shared memory (mmap)



Linux

- Signals and Signal handlers
  Very much like interrupts
  Concurrency issues much like threading

- Pipes
  stdout of one program to stdin of another
  one-way (half duplex)
  ls — sort
  pipe system call / dup



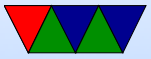
C library has popen()

- FIFOs (named pipes)
  exist as file on filesystem

- SystemV IPC
  shared memory, semaphores ipcs

- Just use mmap

- Unix domain sockets
  Can send file descriptors across



- Splice – move data from fd to pipe w/o a copy? VM magic?
- Sendfile. zero copy?