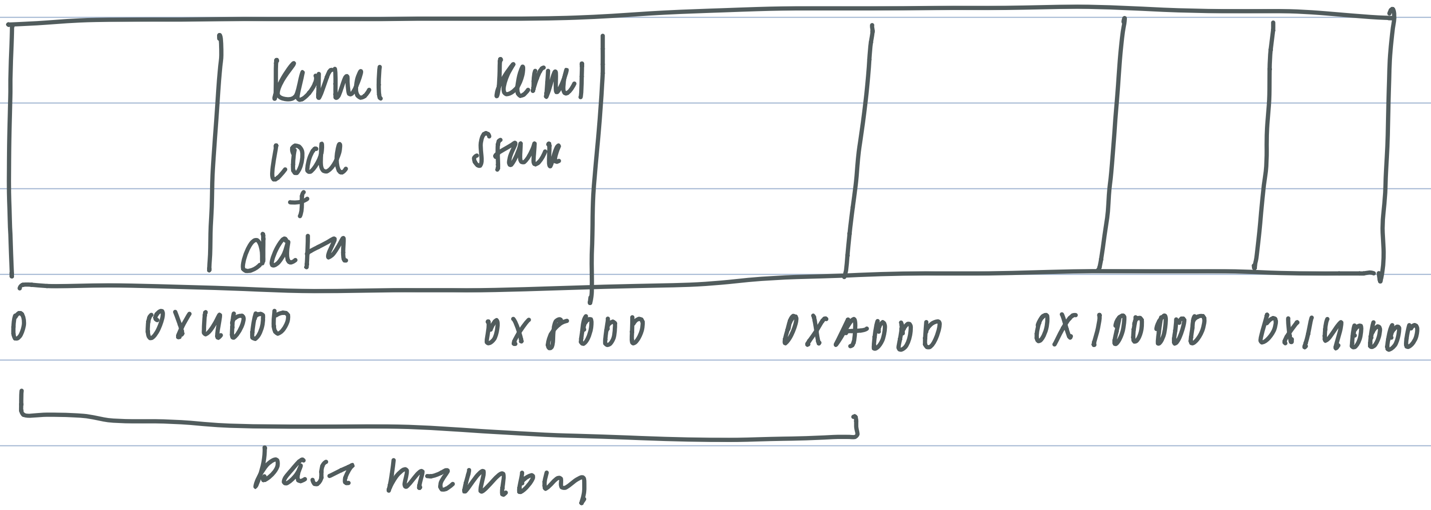
CS61 Section 3

Oct 9

- attempting to dereference random memory
  → segfault
- QEMU: allows for execution of small, slow x86-64 processor
  - has exterior GDB debugging

- WeensyOS: sys_write(message, len(message))
  - have full privilege over system (can step into syscall
    ↳ syscall pushes onto the stack but can be treated as pointing to a register file



syscall: takes register call and put it into standard output

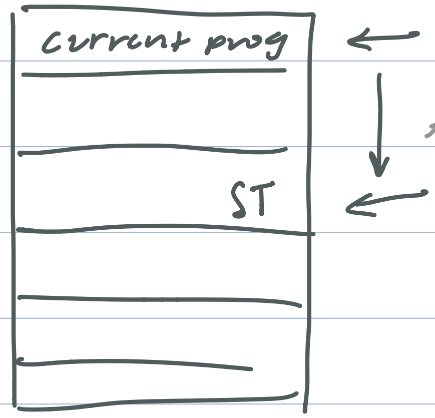
- PID (process ID number)
- Yield : runs the next process

  schedule ( ) : pauses current process, calls another process, and resumes
- write

```
console_puts(-1, 0x0700, (const char*) regs->reg_rdi
        regs->reg_rsi);
```

schedule ( )

NPROC



find some process that is stopped

move saved register file into current register and return

run(ptable[pid] ...)

run( )

Errors in WeensyOS

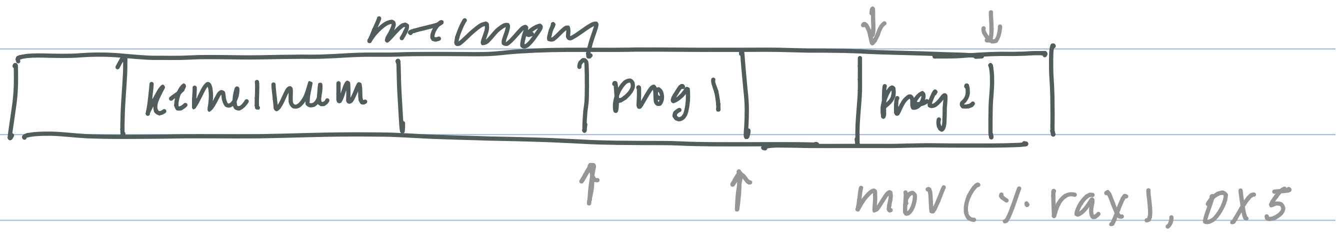
log_printf("hr") from log.txt

- Writing to kernel memory

Kernel page fault for 0x0 (read missing page, rip = 0x0x4f0d)!

segmentation

ex solutions: boundary check the process' memory



* difficult to give a program more space

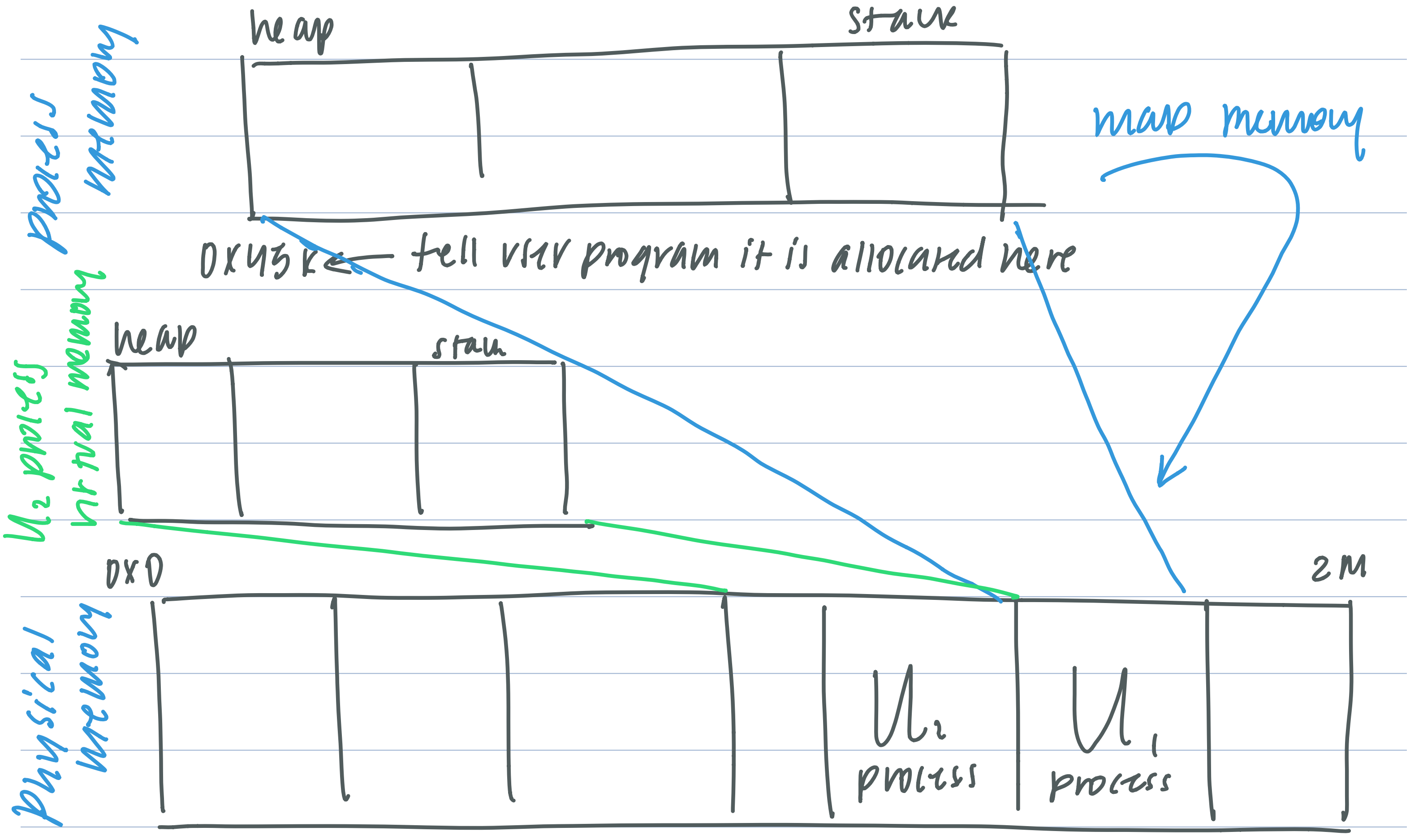
- violates kernel property that it is invisible to user programs
- stack and heap are ~ 2^{47-20} apart, so how would you make a continuous allocation this big?

NOT GOOD.

Virtual memory

one program's address space cannot interfere with anothers

- memory protection
  - kill program when it tries to write outside its bounds/allowed access areas

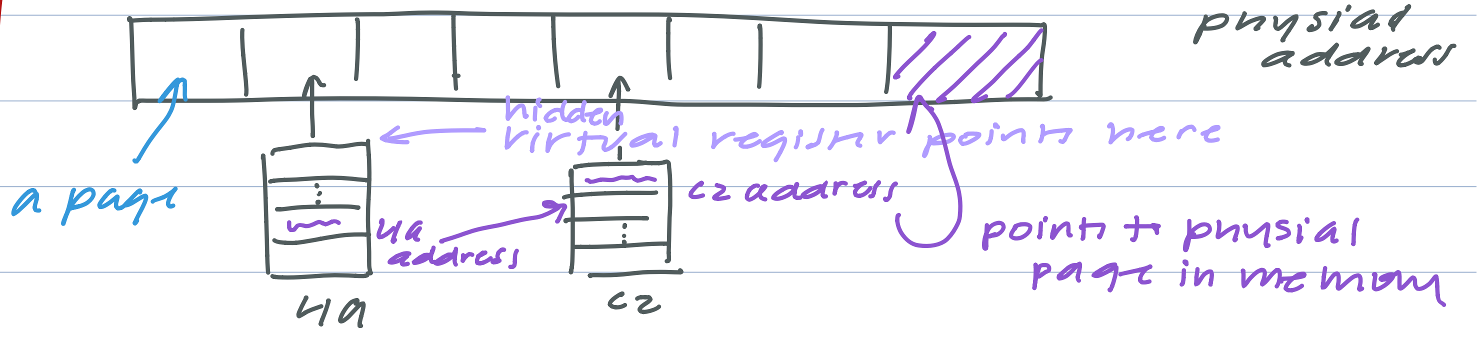


- divide memory into "pages" and each process gets physical pages mapped to virtual space

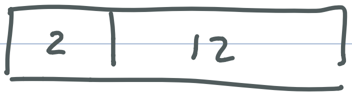
Page Table (x86-64 data structure that helps w/ data allocation

For one process

4ac2400: page table - page-table - 400 bytes into physial address

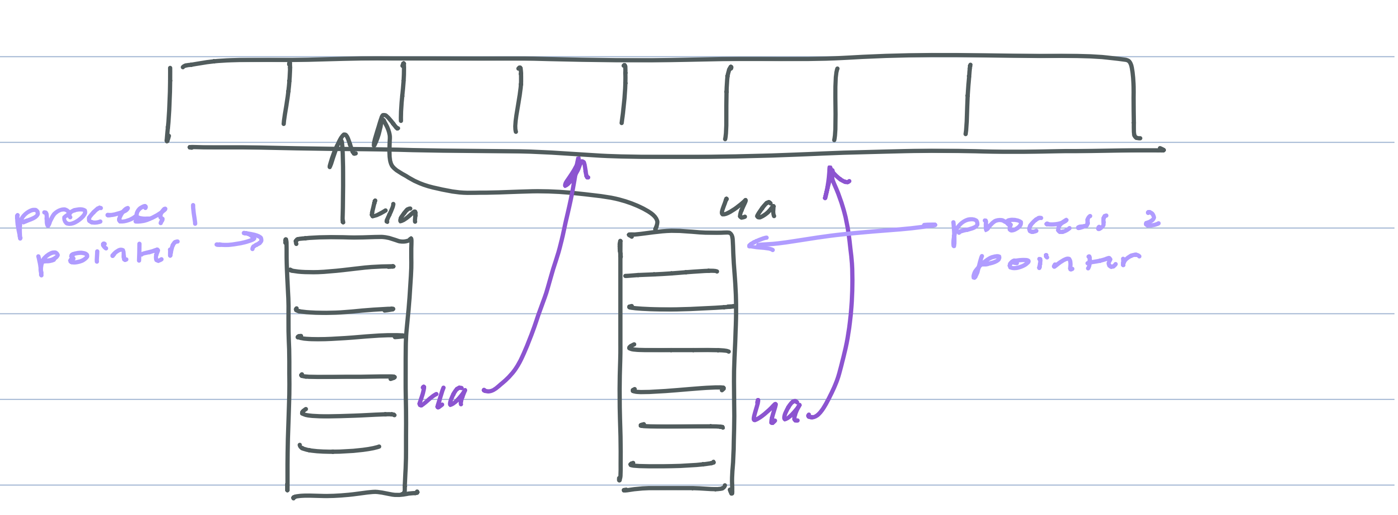

a page
hidden virtual register points here
4a address
c2 address
points to physial page in memory
4a
c2

Kernel maps 4ac2 to 140K



2 byte offset
12 byte pointing to where to go

Two processes



- Can dynamically add entries to page tables
- new process = new level 1 page table
- Kernel has its own page table (has an identity mapping, aka 4K → page table → 4K

Problem set 3

- Vmiter