* One double sided hand written cheating sheet
* Bring ID

Scheduling

Kernel programming

* Analyze code (parse it and say what it does)
* No “write a kernel module”
* More like conceptual stuff

Virtual memory

* MMU (memory management unit) takes virtual memory and transfers it to physical memory
* Instead of interacting directly with physical memory, we provide a translation layer
* A page is the smallest unit of virtual memory
* Virtual pages are mapped to physical memory
* Study mapping pages to physical memory and filling out a page table simulation thing
* **For sure a virtual memory question like the one on slide 26**
  + 14 bit virtual address, 64 byte page size
  + Log2(64) = 6 bits for offset
  + Tlb hit and miss
  + Page fault
  + Vpn acts as index of page table
* TLB – lecture 14 slide 9

Check some examples of the topics on slide 31

Process and signal

* Fork creates a new process from the current process
* Example slide 37
  + Kill is just sending a signal to the getppid() id – it is not terminating a process
  + Why is the output 1?
    - We cant use signals to track events because when you send a bunch of signals they are buffered because the OS can only handle 1 signal at a time
    - Signal is not designed for this and the remaining 4 signals are buffered
  + Wait(NULL) is the parent waiting for the child to be terminated

I/O

* Slide 40
  + The first 3 are taken by standard
  + Then fd1 gets 3
  + Then fd1 is closed and 3 is available
  + Then fd2 gets 3

Concurrency

Semaphore

* For sure one semaphore problem