The first exam will be in class on Monday, February 23. If you have an unavoidable conflict with the date of an exam, please see me as soon as possible (before the exam date!) to discuss options for rescheduling. Last minute rescheduling/extensions will not be accommodated for something known about in advance.

The exam covers material from OSTEP chapters 3–8 (processes and scheduling, up to but not including fair share scheduling).

The exam is closed book. You may have a single page of notes (8.5x11", one side) which will be handed in with your exam. This page may be handwritten or typed and can contain whatever you would like, but it must be a hardcopy — on a piece of paper, not a laptop, tablet, phone, or other device — and must be personally prepared by you — you may not copy another student's page or hand out copies of yours to others. Creating your own notes is an essential part of the learning process — deciding what to include requires engagement with the material which reinforces understanding and improves long-term retention of the material, provides an opportunity for review in order to identify gaps in your knowledge in time to ask questions before the exam, increases confidence in what you do know, and encourages taking ownership of your own learning.

Terms and ideas that you should be familiar with:

- terminology: define/explain virtualization, process, mechanism, policy, process control block, system call, direct execution, trap, interrupt, fault, context switch, trap table, turnaround time, response time
- the hardware organization of a typical system and what happens when a program runs (fetch-decode-execute cycle)
- the role of the operating system
- what virtualization is and why we'd want to do it
- processes
  - what a process is
  - the machine state of a process
  - the main process states and when processes transition between them
  - what happens when a process is created
- process API
  - typical operations provided by a process API
  - the Unix process API: `fork()`, `exec()`, `wait()`/`waitpid()` — you should be able to explain what these system calls do and be able to trace through simple C code involving them to explain what happens
  - how `fork()` results in a tree of processes

OSTEP chapter 5 homework problems (simulation #1, #5, #6) and code (#1–3, #5–7) are good practice for many of the process API topics — copy the directory `/classes/cs331/cpu-api` and its contents to your `~/cs331` directory if you want to experiment with the `fork.py` program. You won't be asked to write code like in the second set of problems, but you should have a sufficient understanding of the behavior of `fork()` etc to be able to answer those questions.

- direct execution and limited direct execution
  - what direct execution is
  - what is limited about limited direct execution and why
  - kernel mode vs user mode
  - what kinds of things cause a switch to kernel mode and the mechanism involved (trap, fault, interrupt)
  - protected control transfer (why this is important, how traps and interrupts are handled safely)
  - system calls vs regular library or function calls (what happens when each occurs, what the differences are)
  - the role of the (user mode) program, hardware, and OS (kernel mode) in starting a process and executing system calls (limited direct execution protocol, figure 6.2)
  - context switches (what they are, when they occur, why they are needed, what happens (the role of the hardware and OS) during a context switch (figure 6.3))

- scheduling policies
  - performance, system efficiency, and fairness metrics for scheduling
  - preemptive vs non-preemptive scheduling — what each is, drawbacks of non-preemptive scheduling
  - scheduling algorithms — FIFO, SJF, STCF/PSJF, RR, MLFQ
    - what the acronyms stand for
    - what the algorithm is / how the policy works — be able to describe the idea of the algorithm as well as generate an execution trace and/or compute response time and turnaround time for a specific set of processes
    - the algorithm's behavior with respect to performance metrics
    - advantages, drawbacks, limitations

  OSTEP chapter 7–8 homework problems (hw2) are good practice for understanding most of the scheduling algorithms.