CSCI 3323 (Principles of Operating Systems), Fall 2021

Reading Quiz 2

Credit: 17.5 points.

1 Reading

Be sure you have read, or at least skimmed, Chapters 4 and 5 of the textbook.

2 Instructions

Answer the questions below using *only* the course textbook — no Web searches. Please work independently rather than in groups, and include the Honor Code pledge in what you turn in, either the full pledge or just the word "pledged". (Please put this in the same document as your answers, so I don't overlook it.)

You may write out your answers by hand and scan them, or you may use a word processor or other program, but please submit PDF or plain text in the "turn-in" folder I set up for you on Google Drive. (So, no word-processor files and no links to other Google Docs. This is a change from how I've asked students to turn in work in previous semesters, meant to reduce both the chance of mistakes on my part and the amount of time I spend managing multiple file formats.)

3 Questions

1. (2.5 points) What are two basic techniques used in operating systems to share resources?
2. (2.5 points) What are "policy" and "mechanism" as used in the textbook, and why separate the two?
3. (2.5 points) When you execute, from the command line, a compiled C program, from an application programmer's point of view "the system" does some mysterious things and then transfers control to its `main()` function with `argc` and `argv` set appropriately. Which system call ("API" in the textbook's usage) does this? sketch what it does, as briefly as you can. (You can get full credit here even if you don't mention *everything*.)
4. (2.5 points) What keystroke combination do UNIX shells use to interrupt the currently running process? To suspend it?
5. (2.5 points) Under UNIX, can users kill any process whose process ID number they know?
6. (2.5 points) The "limited" part of "limited direct execution" relies on several hardware features. List as many of them as you can, based on this chapter.
7. (2.5 points) The "trap" instruction transfers control to — somewhere. How does the system (hardware and/or operating system) know where to go?