CSCI 3323 (Principles of Operating Systems), Fall 2022

Homework 1a

Credit: 25 points.

1 Reading

Be sure you have read, or at least skimmed, Chapters 1 through 6 of the textbook.

2 Problems

Answer the following questions. You may write out your answers by hand and scan them, or you may use a word processor or other program, but please turn in a PDF or plain text file. (No links to shared files on Google Drive please, and no word-processor files.) Turn it in by putting it in your course "TurnIn" folder on Google Drive. Please be sure to include your name somewhere in the file, so when I print it for grading I know whose work it is. (In the pledge is fine.)

1. (5 points) Suppose you have a computer with four processors, and at some point in time there are 100 processes. For each of the three basic states (ready, running, and blocked), what's the largest and smallest number of processes that can be in that state (assuming that every process has to be in one of those states)?

2. (5 points) The textbook discusses various benefits of having an operating system (providing a virtual machine, managing system resources). Can you think of circumstances in which it would be advantageous not to have one? If so, what?

3. (5 points) Chapter 2 mentions that the illusion of every process having its own "address space" can be provided by virtualizing memory, which involves space-sharing physical memory. The very simplest scheme for doing this assigns one contiguous chunk of physical memory to each process and keeps track of the location and size of the current process's chunk using two registers (sometimes called "base" and "limit"). Clearly(?) switching to a new process requires changing the values in these registers. Should the instructions for assigning new values to them be restricted, or should they be executable from user mode as well?

4. (10 points) Most UNIX systems include some command that allows you to trace all system calls made by a process or command. Under Linux, this command is `strace`. For example, to trace all the system calls made during execution of the command `ls -l` and record the output in `OUT`, you would type

   ```
   strace -o OUT ls -l
   ```

   Your mission for this problem is to run `strace` for a command of your choice, capture the output, and then describe what some of it means. Specifically, I want you to:

   - Tell me what command you're tracing.
   - Show output (all or selected lines).
   - Briefly explain at least four lines of the output, each using a different system call: What is the call supposed to do? What do the parameters and return value mean?

The `man` page for `strace` explains the general format of the output. To find out what the individual system calls do, you will need to read their `man` pages. Some of these are easy to find — e.g., the first call is usually to `execve`, and `man execve` will tell you about it. Some are a little harder to track down — e.g., `man write` produces information about a `write` command rather than a system call — but `man` with a section number of 2[1] (e.g., `man 2 write`) should show you the `man` page for the `write` system call.

As an example of what I have in mind, here is a line from a trace of the command `ls /users/cs4320` with commentary.

```
execve("/usr/bin/ls", ["ls", "-l", "/users/cs4320"], [/* 40 vars */])
= 0
```

The call to `execve` creates a new process to run the command. Parameters are the command to execute, the arguments to pass to it, and an array of environment variables (quite a lot of them, apparently!). The return value of 0 probably doesn't mean anything, since the `man` page for `execve` says that the function doesn't return if the call is successful.

(You should of course choose system calls other than `execve`.)

3 Essay and pledge

Include with your assignment the following information.

For programming assignments, please put it a separate file. (I strongly prefer plain text, but if you insist you can put it in a PDF — just no word-processor documents or Google Drive links please.) For written assignments, please put it in your main document.

3.1 Pledge

This should include the Honor Code pledge, or just the word "pledged", *plus* at least one of the following about collaboration and help (as many as apply). Text *in italics* is explanatory or something for you to fill in; you don't need to repeat it!

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*

- I worked with *names of other students* on this assignment.

- I got help with this assignment from *source of help — ACM tutoring, another student in the course, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*

- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*

- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

[1] `man` pages are organized into "sections" — one for commands, one for system calls, one for library functions, etc.

3.2 Essay

This should be a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy.