CS 4375, Operating Systems, Test 1

Fall 2024, September 26 name _____

78 minutes 10:32 – 11:50

One page of handwritten notes is allowed. There are no room re-entry privileges.

/

1. [7 points] True or False
   T F The scheduler is the lowest-priority process.
   T F After an exec(), a process will go to the blocked queue.
   T F After its time slice expires, a process goes to the ready queue.

   T F System calls employ the trap mechanism.
   T F A process can change its process ID with pcb=os.getpcb(); pcb.pid=*n*;

   T F Different processes typically use disjoint (non-overlapping) segments of physical memory.
   T F Different processes typically use different data paths in of the processor.

2. [1] Process inspection tools, such as Windows Task Manager and ps, list many interesting statistics about each process, but never tell you exactly which processes are *currently* in the ready queue, which are blocked, and which is running. This is most likely because:
   a. This would be a security risk
   b. The OS does not have access to this information
   c. This changes too fast for any human to benefit from such a realtime display
   d. all of the above
   e. none of the above

3. [4] A process can obtain useful information in various ways. Which of the following are true?
   a) T F Getting the value of an environment variable requires a system call.
   b) T F Asking the user requires a system call.
   c) T F Parsing its arguments requires a system call.
   d) T F Reading a configuration file requires a system call.

4. [1] Most of the time, the return value of a system call is
   a. 0
   b. 1
   c. A non-zero integer
   d. A short string
   e. undefined
   f. This question makes no sense, since system calls almost never return.

5. [1] Many supercomputer installations charge per minute of runtime. If process X causes a system call C, how should the time the OS spends handling C be charged?
   a. to the person running Process X
   b. amortized (averaged) across all processes that ran that day
   c. neither

6. [2] The shell command cat -n secret1 secret2 | grep spy > names.txt creates how many processes? How many pipes?

7. [3] The kernel keeps track of many things for each process. These include all of the following except three. Which?
    a. The parent process.
    b. The pid (process ID number).
    c. The current working directory.
    d. The current time (UTC).
    e. The file descriptors.
    f. The current state (running, etc.)
    g. The name of the file that contains the executable.
    h. A pointer to the source code used to generate the executable.
    i. The amount of CPU time used so far.
    j. The start and end of the memory that is allocated to that process.

8. [1] If you start running a long program, but then delete it, for example, with

```
spinner.py 1000000000 &
rm spinner.py
```

the process (e.g. spinner) will end:
    a. Immediately
    b. Never
    c. The next time it makes a system call
    d. The next time it is suspended (so that another process can get a time slice)
    e. None of the above

9. [3] Imagine running the following program. (Note that kill(*pid*, 9) will kill the process with the specified process ID.)

```
import os
os.fork()
pid = os.fork()
if (pid > 0):
   os.kill(pid,9)
os.fork()
print("done")
```

How many times will "done" be output?

11. [3] An attacker with physical access to a running computer may execute a cold boot attack by abruptly turning off the power, then booting a lightweight operating system from a removable disk to dump the contents of pre-boot physical memory to a file. An attacker is then free to analyze the data dumped from memory to find sensitive data, such as encryption keys. (adapted from wikipedia)

True or False
    a. T F In a cold boot attack you can't get the values of program variables, since it only gets physical memory contents, but processes use virtual memory.
    b. T F Cold boot attacks can be prevented if the superuser always chooses a long complex password.
    c. T F Cold boot attacks can only obtain information about the memory used by one process (the one that was running when the power was turned off).

10. [7] Name some of the system call(s) that are most relevant to each of the following shell constructs. Hint: consider wait/waitpid, dup/dup2, open, pipe, exit, close, and exec.

a) & _____

b) < _____ and _____

c) > _____ and _____

d) | _____ and _____

12. [3] If a running process is suspended and then resumes, the OS ensures that it continues from the point it left off. Explain how this is accomplished.

13. [3] If the file ptk contains the word elephant, the output of the following code is pha. Explain why.

```
1 import os
2 inFd = os.open("ptk", os.O_RDONLY)
3 dpFd = os.dup(inFd)
4 ignore = os.read(inFd,3)
5 print(os.read(dpFd,3))
```

14. [3] It is the responsibility of the OS to prevent processes from snooping on or corrupting each other. Discuss the role of address spaces and memory protection *or* the role of kernel mode and trapping instructions.

15. [3] Can a process with an infinite loop freeze your computer, so badly that it needs to be rebooted? Explain.

16. [4 points] A scheduler is just a program that implements an algorithm. In Linux, you can modify the scheduler as you wish, then compile and link it to create a new kernel, which you can use for your computer's OS. Pick one of the following problems and explain how it might be fixable by modifying the scheduler.
    a) the computer starts to get slow every afternoon, but rebooting fixes this
    b) the disk is at 95% (meaning that 95% of the possible CPU-to-disk bandwidth is in use)
    c) the computer sometimes takes 2 seconds to even echo a keystroke
    d) your computation-intensive program for forecasting tomorrow's weather takes three days to run
    e) streaming video keeps halting

17. [5] When a computer is in supervisor model (= kernel mode), any instruction may be executed, including those reserved for use by trusted code of the operating system. When a computer is in user mode, only a limited set of "safe" instructions may be executed. Which three of the following are possible ways that you, as an applications programmer, could cause the system to shift (temporarily) to supervisor mode?
    a. enter an infinite loop
    b. make a system call relating to the process API, such as fork() or exec()
    c. attempt to write to a memory location outside the space allocated to the process
    d. make a system call relating to I/O, such as dup(), read() or write()
    e. run tetris as superuser (or example with sudo or after su or logging in as admin or root)

    For the one that you are least sure of, explain your thinking.