CS 111: Operating System Principles
Lecture 12

Midterm Review

1.0.0

Jon Eyolfson
April 27, 2021

The Midterm Will Be Open for 24 Hours

Anytime on the April 29 in PST
    (12:00 AM to 11:59 PM)

The midterm will be 2 hours from when you start
    No pausing and coming back

It'll be open book
    I'm assuming there's no way to stop that

The Exam Will Have 3 Types of Questions

There will be:

- 15 fill in the term / multiple choice questions
- 6 short answers
- 2 long answers

It'll cover everything up and including Lecture 11

It'll be graded out of 100

Winter 2020 Midterm is Good Indicator

We'll go over that, except for question 6 and 7

For question 6, it's the clock algorithm we went over

It was basically FIFO

For question 7, we didn't talk about emulation

Maybe we'll talk about Windows Subsystem for Linux later in the course

Question 1

What is the benefit of using the copy-on-right optimization when performing a fork in the Linux system?

Question 2

Round Robin, First come First Serve, and Shortest Job First are three scheduling algorithms that can be used to schedule a CPU.

What are their advantages and disadvantages? Which one is likely to have the largest overhead? Why?

Question 3 (1)

Assume you have a system with three processes (X, Y, and Z) and a single CPU. Process X has the highest priority, process Z has the lowest, and Y is in the middle.

Assume a priority-based scheduler (i.e., the scheduler runs the highest priority job, performing preemption as necessary). Processes can be in one of five states: RUNNING, READY, BLOCKED, not yet created, or terminated.

Given the following cumulative timeline of process behavior, indicate the state the specified process is in AFTER that step, and all preceding steps, have taken place. Assume the scheduler has reacted to the specified workload change.

Question 3 (2)

For all questions in this Part, use the following options for each answer:

- RUNNING
- READY
- BLOCKED
- Process has not been created yet
- Not enough information to determine
- None of the above

Question 3 (3)

- Process X is loaded into memory and begins; it is the only user-level process in the system. Process X is in which state?
- Process X calls fork() and creates Process Y.Process X is in which state?
- The running process issues an I/O request to the disk. Process X is in which state? Process Y is in which state?
- The running process calls fork() and creates process Z. Process X is in which state? Process Y is in which state? Process Z is in which state?
- The previously issued I/O request completes. Process X is in which state? Process Y is in which state? Process Z is in which state?

Question 4

For the next two questions, assume the following code is compiled and run on a modern linux machine (assume any irrelevant details have been omitted):

```
main() {
  int a = 0;
  int rc = fork();
  a++;
  if (rc == 0) { rc = fork(); a++; }
  else { a++; }
  printf“(Hello!\”n);
  printf“(a is %d\”n, a);
}
```

- Assuming fork() never fails, how many times will the message `Hello!\n` be displayed? Explain how you get this result.
- What will be the largest value of “a” displayed by the program? Explain how you get this result.

Question 5

Consider the following set of processes, with associated processing times and priorities:

Process	Burst Time	Priority
A	4	3
B	1	1
C	2	3
D	1	4
E	4	2

All of the processes arrive at time 0 in the order Process A, B, C, D, E.

For each scheduling algorithm, fill in the table with the process that is running on the CPU (for time slice-based algorithms, assume a 1 unit time slice).

Table says to do: FIFO, Round Robin, SRTF, Priority

We'll Quickly Run Over Lectures 1 to 11

Most of the concepts are important, you'll have to know the terminology

Don't rely too much on the exam being open book

If you're looking up everything you'll run out of time

You'll want to understand what you did in Lab 0 and 1