New York University
Tandon School of Engineering
Department of Computer Science and Engineering
Introduction to Operating Systems
Fall 2024
Assignment 4 (10 points)

Problem 1 (2 points)

If you create a main() routine that calls fork() twice, then execl() once in one of the child processes, i.e., if it includes the following code:

```
pid_t x=-11, y=-22;
x = fork();
if(x==0) {
    y = fork();
    if (y == 0) execl("/bin/ls", "ls", "-l", NULL);
}
```

Assuming all fork() calls succeed, and execl() replaces the child process, draw a process tree similar to that of Fig. 3.8 (page 116) in your textbook, clearly indicating the values of x and y for each process in the tree (i.e., whether 0, -11, -22, or larger than 0). The process tree should only have one node for each process, and each line/arrow in the process tree diagram shall represent a creation of a process, or alternatively a parent/child relationship. The process tree should be a snapshot just after all forks completed but before any process exits. (Insert process tree diagram here)

Problem 2 (4 points)

Write a program that creates the process tree shown below: (Insert process tree diagram here) The tree should have a parent process that forks three child processes. Each of these children should then fork two children each. Clearly show the parent-child relationships in your code.

Problem 3 (4 points)

Write a program whose main routine obtains two parameters, n and m, from the user (passed to your program when invoked from the shell, n¿2, m¿2). It creates two child processes. The first child process shall create and print the first n prime numbers. The second child process shall calculate and print the factorial of m. The parent waits for both children to exit and then prints the sum of n and m. Do not use IPC in your solution to this problem (i.e., neither shared memory nor message passing).

Problem 4 (2 points)

Explain the difference between the fork() and vfork() system calls in terms of memory management and process creation. When might you choose to use vfork() over fork(), and what are the potential risks associated with using vfork()?

Problem 5 (2 points)

Draw a process state diagram showing the possible states a process can be in, and the transitions between these states. Include states such as running, ready, blocked, and terminated, and label the transitions with the events that cause them (e.g., I/O completion, time slice expiry).

Problem 6 (4 points)

Write a C program that simulates a simple producer-consumer problem using two threads. The producer thread generates random numbers and adds them to a shared buffer (e.g., a circular buffer or a queue). The consumer thread removes numbers from the buffer and prints them. Use appropriate synchronization mechanisms (e.g., mutexes, condition variables) to prevent race conditions and ensure that the producer doesn't try to add to a full buffer and the consumer doesn't try to remove from an empty buffer. The program should run for a specified number of iterations.

Problem 7 (2 points)

Describe the concept of a zombie process and an orphan process. Explain how each arises and how they can be handled or avoided.

What to hand in (using Brightspace)

Please submit the following files individually:

1. Source file(s) with appropriate comments. The naming should be similar to "`lab#_$.c`" (# is replaced with the assignment number and $ with the question number within the assignment, e.g., `lab4_b.c`, for lab 4, question b OR `lab5_1a` for lab 5, question 1a).
2. A single pdf file (for images + report/answers to short-answer questions), named "`lab#.pdf`" (# is replaced by the assignment number), containing:
   - Screenshot(s) of your terminal window showing the current directory, the command used to compile your program, the command used to run your program, and the output of your program.
3. Your Makefile, if any. This is applicable only to kernel modules.

RULES

- You shall use kernel version 4.x.x or above. You shall not use kernel version 3.x.x.
- You may consult with other students about GENERAL concepts or methods but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied from other students, the internet, or any other source).
- If you are having trouble, please ask your teaching assistant for help.
- You must submit your assignment prior to the deadline.