

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()` four times, i.e., if it includes the following code:

```
pid_t w=-11, x=-22, y=-33, z=-44;  
w = fork();  
if(w==0) x = fork();  
if(x==0) y = fork();  
if(y>0) z = fork();
```

Assuming all `fork()` calls succeed, draw a process tree similar to that of Fig. 3.8 (page 116) in your textbook, clearly indicating the values of `w`, `x`, `y` and `z` for each process in the tree (i.e., whether 0, -11, -22, -33, -44, or larger than 0). Note that the process tree should only have one node for each process and thus the number of nodes should be equal to the number of processes. The process tree should be a snapshot just after all forks completed but before any process exists. Each line/arrow in the process tree diagram shall represent a creation of a process, or alternatively a parent/child relationship.

Process Tree Diagram (Insert here)

Problem 2 (4 points)

Write a program that creates the process tree shown below:

Process Tree Diagram (Insert here – A tree with a root and three children, each child having two children)

Problem 3 (4 points)

Write a program whose main routine obtains a parameter n from the user (i.e., passed to your program when it was invoked from the shell, n_1) and creates a child process. The child process shall then calculate the sum of numbers from 1 to n (inclusive) and print the result. The parent waits for the child to exit and then prints double the sum calculated by the child. Do not use IPC in your solution to this problem (i.e., neither shared memory nor message passing).

Listing 1: Example C Code

//Insert your C code here

Problem 4 (2 points)

Explain the difference between a process and a thread. Discuss the advantages and disadvantages of using threads compared to processes.

Problem 5 (2 points)

Describe the concept of a race condition. Give a simple C code example demonstrating a race condition involving two threads accessing and modifying a shared variable. Explain how to prevent this race condition using mutexes.

Problem 6 (4 points)

Write a C program that uses threads to compute the factorial of a number entered by the user. The program should create two threads: one to compute the factorial iteratively and the other to compute it recursively. Both threads should store their results in a shared variable, and the main thread should then print both results (iterative and recursive) along with a message indicating whether they are equal or not. Appropriate synchronization mechanisms (like mutexes) should be used to prevent race conditions.

Listing 2: Example C Code

//Insert your C code here

Problem 7 (2 points)

Describe the producer-consumer problem. Explain how semaphores can be used to solve this problem, and provide pseudocode to illustrate your solution.

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.