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, i.e., if it includes the following code:

```
pid_t x=-11, y=-22;
x = fork();
if(x==0) y = 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 x and y for each process in the tree (i.e., whether 0, -11, -22, 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.

[Insert Process Tree Diagram Here - Problem 1]

Problem 2 (4 points)

Write a program that creates the process tree shown below:

[Insert Process Tree Diagram Here - Problem 2]

Problem 3 (4 points)

Write a program whose main routine obtains parameters n and m from the user (passed to your program when invoked from the shell, n¿2, m¿0) and creates m child processes. Each child process shall then create and print a sequence of n consecutive even numbers starting from 2. The parent waits for all children to exit and then prints the sum of all the even numbers printed by its children. 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 a process and a thread. Give an example of when you might prefer to use threads over processes, and vice versa.

Problem 5 (3 points)

Consider a system with three processes, P1, P2, and P3, and two resources, R1 and R2. Initially, P1 holds R1, P2 holds R2, and P3 holds neither. P1 requests R2, P2 requests R1, and P3 requests R1. Illustrate the situation using a resource allocation graph. Is there a deadlock? Explain your answer.

Problem 6 (2 points)

What is a race condition? Provide a simple code example demonstrating a race condition in C, and explain how you would prevent it.

Problem 7 (2 points)

Describe the concept of a critical section. Why is it important to protect critical sections? Give an example of a solution to protect critical sections.

Problem 8 (3 points)

Write a C program that uses signals to handle Ctrl+C (SIGINT). When Ctrl+C is pressed, the program should gracefully terminate, printing a message "Program exiting gracefully..." before exiting.

Problem 9 (3 points)

Write a C program that simulates a producer-consumer problem using a bounded buffer with a maximum size of 5. The producer should generate random integers

and add them to the buffer, while the consumer should remove and print integers from the buffer. Use semaphores to synchronize the producer and consumer.

Problem 10 (3 points)

Explain the differences between the following scheduling algorithms: First-Come, First-Served (FCFS), Shortest Job First (SJF), and Round Robin. Discuss the advantages and disadvantages of each.

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 c 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 form 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.

Operating Systems - Prof. Omar Mansour