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

Your Name

September 11, 2025

Problem 1 (2 points)

If you create a main() routine that calls fork() twice, followed by a call to execl(), 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, draw a process tree showing the processes just after the second fork but before the execl() call. Clearly indicate the values of x and y for each process (0, -11, -22, or ¿0). Include a brief explanation of what happens after the execl() call.

process_tree1.png

Figure 1: Process Tree

Problem 2 (4 points)

Write a program that creates the process tree shown below:
(Insert the process tree diagram here as an image - a tree with 4 processes, parent with 3 children)

```
//Insert your code for Problem 2 here.  Use appropriate comments.  Example:
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    pid_t pid;
    // ... your code ...
    return 0;
}
```

Problem 3 (4 points)

Write a program whose main routine obtains two parameters, a and b, from the user (passed when invoked from the shell). It then creates a child process. The child process calculates a + b and prints the result. The parent waits for the child and then prints a - b. Do not use IPC.

```
//Insert your code for Problem 3 here. Use appropriate comments. Example:
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s <a> <b>\n", argv[0]);
        return 1;
    }

    long long a = atoll(argv[1]);
    long long b = atoll(argv[2]);
    // ... your code ...
    return 0;
}
```

Problem 4 (2 points)

Explain the difference between fork() and vfork() in the context of memory sharing and scheduling. What are the potential pitfalls of using vfork()?

Problem 5 (2 points)

Describe a scenario where using pipes would be more advantageous than using shared memory for inter-process communication. Conversely, describe a scenario where shared memory would be preferred over pipes.

Problem 6 (3 points)

Write a C program that uses two pipes to implement a simple inter-process communication where the parent process sends a string to the child process, and the child process reverses the string and sends it back to the parent. The parent then prints the reversed string.

Problem 7 (3 points)

Write a C program that uses shared memory to allow two processes to concurrently update a shared counter. Implement appropriate synchronization mechanisms (e.g., mutexes, semaphores) to prevent race conditions.

Problem 8 (2 points)

Explain the concept of a zombie process. How can zombie processes be avoided?

Problem 9 (2 points)

What are the differences between a process and a thread? Discuss the advantages and disadvantages of using threads versus processes.

Problem 10 (3 points)

Write a C program that creates three child processes. Each child process calculates the factorial of a different number (e.g., 5, 10, 15). The parent process waits for all children to finish and then prints the sum of the factorials calculated by the children. Use appropriate error handling.

What to hand in (using Brightspace)

1. Source file(s) with appropriate comments. 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.