

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, clearly indicating the values of `x` and `y` for each process. The process tree should be a snapshot just after all forks completed but before any process exits.

**Process Tree Diagram should be inserted here as an image.**

**Problem 2 (4 points)**

Write a program that creates the process tree shown below:

**Process Tree Diagram should be inserted here as an image.**

**Problem 3 (4 points)**

Write a program whose main routine obtains parameters `n` and `m` from the user (passed when invoked from the shell, `n_i2`, `m_i0`) and creates `m` child processes. Each child process shall print a different line containing "Child process ID:

pid\_t Printing number: i" where i is a unique number from 0 to m-1. The parent waits for all children to exit before printing "Parent process exiting." Do not use IPC.

### Problem 4 (2 points)

Explain the difference between a zombie process and an orphan process. Provide an example scenario for each.

### Problem 5 (1 point)

What is the purpose of the 'wait()' system call? What happens if a parent process doesn't call 'wait()' for its child processes?

### Problem 6 (2 points)

Consider the following code snippet:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    pid_t pid;
    pid = fork();
    if (pid < 0) {
        fprintf(stderr, "Fork failed\n");
        return 1;
    } else if (pid == 0) {
        printf("Child process ID: %d\n", getpid());
        execl("/bin/ls", "ls", "-l", NULL);
        printf("This will not print.\n");
    } else {
        printf("Parent process ID: %d\n", getpid());
        wait(NULL);
        printf("Parent process exiting.\n");
    }
    return 0;
}
```

Explain the output of this program. What is the role of 'execl'?

### Problem 7 (1 point)

What is the difference between ‘fork()’ and ‘vfork()’? Why is ‘vfork()’ generally avoided?

### Problem 8 (2 points)

Draw a process state diagram showing the possible states a process can be in, including transitions between states. Label each state and transition clearly (e.g., Running, Ready, Blocked, etc.).

### Problem 9 (2 points)

Write a C program that uses ‘fork()’ to create a child process. The parent process should send a signal (e.g., SIGUSR1) to the child process. The child process should catch this signal and print a message indicating that it received the signal.

### Problem 10 (2 points)

Describe a scenario where using signals could be beneficial for inter-process communication. What are the advantages and disadvantages of using signals compared to other IPC mechanisms (e.g., pipes, shared memory)?

## 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.