

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

A) (2 points)

If you create a `main()` routine that calls `fork()` twice, then `execlp()` once, i.e. if it includes the following code: `pid_t x=-11, y=-22;`

```
x = fork();  
if(x==0) y = fork();  
if(y==0) execlp("/bin/ls", "ls", "-l", NULL);
```

Assuming all `fork()` and `execlp()` calls succeed, draw a process tree showing the parent-child relationships. Clearly indicate which processes execute `‘/bin/ls’` and which execute the original `main()` routine. 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. Each node should contain the PID (you can assign arbitrary positive PIDs) and indicate whether it’s running `‘/bin/ls’` or the original `main()`.

B) (4 points)

Write a C program that creates the following process tree:

```
1 / 2 3 / 4 5
```

Each process should print its process ID (PID) and its parent’s process ID (PPID). Use the `‘getpid()’` and `‘getppid()’` system calls. The parent process (PID 1) should wait for all its children to finish before exiting.

C) (4 points)

Write a C program that takes two command-line arguments: an integer 'n' ($n \geq 0$) and a string 'str'. The program should create 'n' child processes. Each child process should print the string 'str' followed by its process ID. The parent process should wait for all its children to finish before exiting. Ensure proper error handling for invalid input (e.g., insufficient arguments or non-integer input for 'n').

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 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.