

New York University Tandon School of
Engineering - Introduction to Operating Systems
- Assignment 4

Fall 2024

Assignment 4 (7 problems, 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). The process tree should only have one node for each process. The process tree should be a snapshot just after all forks completed but before any process exits. 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 2 (2 points)

Write a program that creates the process tree shown below:

[Insert Process Tree Diagram Here - A different tree than Problem 1]

Problem 3 (2 points)

Write a C program that creates three child processes. Each child process should print its process ID and the ID of its parent process. The parent process should wait for all three child processes to finish before exiting.

Problem 4 (1 point)

Explain the difference between the `fork()` and `exec()` system calls in terms of their effect on a process's memory space and execution.

Problem 5 (1 point)

Describe a scenario where using `fork()` followed by `exec()` would be preferable to only using `fork()`.

Problem 6 (1 point)

What is the purpose of the `wait()` system call? How does it help avoid zombie processes?

Problem 7 (1 point)

Write a short C program that uses `fork()` to create a child process. The child process should execute the `ls -l` command, while the parent process waits for the child to complete and then prints "Child process finished".

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.