

Project 3 Practice Process APIs

In this Project, you are to gain some familiarity with the process management APIs.

Submission to GitHub: source code (programs and related files for running the programs) and a separate Word file with your answers to the following questions. Please name your programs with the question number.

Submission to Canvas: a text indicating your project is ready on GitHub.

Questions:

1. Write a program that calls `fork()`. Before calling `fork()`, have the main process access a variable (e.g., `x`) and set its value to something (e.g., `100`). What value is the variable in the child process? What happens to the variable when both the child and parent change the value of `x`?

2. Write another program using `fork()`. The child process should print “hello”; the parent process should print “goodbye”. You should try to ensure that the child process always prints first; can you do this without calling `wait()` in the parent?

3. Write a program that calls `fork()` and then calls some form of `exec()` to run the program `/bin/ls`. Try **at least one** of the variants of `exec()`, including (on Linux) `execl()`, `execle()`, `execlp()`, `execv()`, `execvp()`, and `execvpe()`. Why do you think there are so many variants of the same basic call?

4. Now write a program that uses `wait()` in the parent to wait for the child process to finish. What does `wait()` return? What happens if you use `wait()` in the child?