

Assignment 3-2-1: Working with UNIX Processes in C++

Assignment Instructions:

- You must use the virtual environment that you set up in Exercise 1-1-3 for this assignment.
- Be sure to compile and test each program to be certain it works as expected. If you aren't sure how to compile and run a C++ program, refer to the [Build and Execute Program](#) section of the [setup instructions document](#).

Important notes:

- At the top of your .cpp file, please include a comment with your **full name**. If your section uses **Lightweight Teams**, add the names of the teammates whom you worked with to the same comment.
- Add your own **individual comment** for each function / major portion of code that you add, briefly explaining what that part does.
- If you are asked to submit screenshots and your submitted screenshots do not match with your program's actual behavior, we will consider that to be a **violation of academic integrity** and pursue it accordingly.
- Make sure to **organize and format** your code in a consistent way.
- If you refer to any online resource to understand a concept, see examples of the use of a particular syntax, etc., add a comment **citing** that resource (i.e., specify website name and link).
- You must only submit **.cpp** files. If you have multiple .cpp files, upload them individually and **not** as a zip / compressed file.
- No screenshot(s) will mean no grade for this assignment.

Assignment Objectives:

- The purpose of this activity is to explore and use functions related to **process creation & handling** in Unix-based OSs.

Assignment Tasks:

An online version of the Linux manual can be found here:

<http://linux.die.net/man/>. For this activity, you will mainly need to refer to the **system calls** section of the manual.

If you need help with **navigating** the file system through a command line terminal, refer to this: http://linuxcommand.org/lc3_lts0020.php .

Assignment Setup (0 points)

Note:

- *You need to use the terminal to compile and run the program. Do not use your IDE's GUI.*

1. You will need to download, compile, and execute a small program using your virtual C++ environment.
2. Type the following command **into the terminal window** to pull the project repository from GitLab:

```
git clone https://cci-git.charlotte.edu/jbahamon/ITSC_3146_A_3_2.git
```

3. Change directory into the newly created directory (folder) named ITSC_3146_A_3_2.
4. Issue the following command to compile the code:
`g++ Assignment_3_2.cpp Processes.cpp -o Assignment_3_2`
5. Issue the following command to execute the program:
`./Assignment_3_2`

Part 1: Working With Process IDs (5 points)

1. Modify the `getProcessID()` function in the file named `Processes.cpp`
 - a. The function must find and store the process's own **process id**
 - b. The function must return the process id to the calling program.
Note that the function currently returns a default value of -1.
Hint: search for "process id" in the "System Calls" section of the Linux manual.

Take a screenshot of a sample output and upload the picture as part of your assignment submission.

Part 2: Working With Multiple Processes (10 points)

1. Modify the `createNewProcess()` function in the file named `Processes.cpp` as follows:
 - a. The **child** process must **print** the message *I am a child process!*
 - b. The **child** process must then **return a string with the message** *I am bored of my parent, switching programs now*
 - c. The **parent** process must **print** the message *I just became a parent!*
 - d. The **parent** process must then **wait** until the child process terminates, at which point it must **return a string with the message** *My child process just terminated!* and then return. *Hint: search for "wait for process" in the "System Calls" section of the Linux manual.*

IMPORTANT:

Do NOT type the messages, copy-paste them from here.
Your output must be exact.

Take a screenshot of a sample output and upload the picture as part of your assignment submission.

Part 3: Working With External Commands (15 points)

1. Modify the `replaceProcess()` function in the file named `Processes.cpp` as follows:
 - a. The parent process will use the `fork()` function to create a child process. *This step has been done for you.*
 - b. The **child** process must then change its memory image to a different program by using the **execvp** system call (<http://linux.die.net/man/3/execvp>). The parameter **args** that has been passed to the `replaceProcess()` function is the array of parameters to be passed to **execvp**, telling it which program to execute and what parameters to pass to that program.

For example, the test code provided to you executes the `"ls"` (directory list) program with the parameter `"-l"` by setting the **args** array as follows:

```
char * args[3] = {(char * )"ls", (char * )"-l", NULL};
```

IMPORTANT:

Although the test code executes the "ls" program, we must be able to change the command that execvp executes. So, DO NOT hardcode the command to be passed to execvp. Simply use the args array provided.

- c. Finally, in the **parent** process, you must **make sure to invoke the necessary system call to wait for the child process to terminate**. Once the child terminates, exit the program.

Take a screenshot of a sample output and upload the picture as part of your assignment submission.

Expected Output:

If you execute the code you pull from GitLab without any modifications, it will produce the following output:

```
Part 1: Working With Process IDs
Process ID: -1

Part 2: Working With Multiple Processes

Part 3: Working With External Commands
```

If you make all required changes correctly, your program should produce output similar to the following (**Note:** To verify your output for Part 3, execute the "ls -l" command directly in your terminal. Then match what you see there to the directory listing you get from Part 3 of your program. They should be the same.):

```
Part 1: Working With Process IDs
Process ID: 3329

Part 2: Working With Multiple Processes
I just became a parent!
I am a child process!
I am bored of my parent, switching programs now

My child process just terminated!

Part 3: Working With External Commands
total 52
-rwxr-xr-x 1 ubuntu ubuntu 14570 Jan 20 20:21 Assignment_3_2
-rw-r--r-- 1 ubuntu ubuntu 1124 Jan 20 20:18 Assignment_3_2.cpp
-rw-r--r-- 1 ubuntu ubuntu 4984 Jan 20 20:21 Assignment_3_2.o
-rw-r--r-- 1 ubuntu ubuntu 378 Jan 20 20:18 Makefile
-rw-r--r-- 1 ubuntu ubuntu 1466 Jan 20 20:20 Processes.cpp
-rw-r--r-- 1 ubuntu ubuntu 495 Jan 20 20:18 Processes.h
-rw-r--r-- 1 ubuntu ubuntu 5088 Jan 20 20:21 Processes.o
-rw-r--r-- 1 ubuntu ubuntu 905 Jan 20 20:18 Processes_SK.cpp
```

Caution: Before you submit, make sure that you have followed all the instructions under [Assignment Tasks](#) and [Important notes](#) and that you have taken screenshots as indicated in the assignment.

Assignment Submission Items:

The files that need to be submitted for this assignment are the following:

- Processes.cpp
- The necessary output screenshots for the cpp file.

Note: No screenshot(s) will mean no grade for this assignment.