

Operating System Project

[10 points]

An interactive shell program such as Cygwin shell or cmd prompt or terminal in Mac or linux takes command line input from the user and then execute the command/program specified by the user.

In this exercise, you will implement **closh** (Clone Shell), a simple shell-like program designed to run multiple copies of a program at once.

Like any other shell, closh takes as input the name of the program to run (e.g., hello, etc.). However, closh also takes two additional inputs:

1. The number of copies (processes) of the program to run. This is an integer from 1 to 9.
2. Whether the processes should execute concurrently or sequentially.

Closh executes the given program the specified number of times, then returns to the prompt once all processes have either completed or timed out. Here is a simple example of using closh (italic is user input and hello.exe is the hello world example presented in class):

```
oelkhat@loyno-Laptop$ ./closh
closh> ./hello.exe
  count> 3
[p]arallel or [s]equential> p
hello world
hello world
hello world

closh>
```

To write such a shell in C++, first review programs in the Process slide.

As noted in the slide, you can use `execlp()` to create a new child process and have it execute a command. Use also the `waitpid` in the parent process to wait for the child process to finish in case sequential execution is selected.

Due Date: Sunday 2/9/2020 at 11:59 pm

For simplicity, assume that the user specifies the full path name for any command/executable that they wish to execute. Thus you do not need to deal with path name completion issues. You can test your shell on programs you write in C++ (after compiling in machine language).

What to submit:

Submit your cpp program through blackboard.