**Report**

The form of exec() function that I used was fork(). The purpose of using this function was to create a new process that mirrored all the current content and state of the targeted first process. However, one difference is that one would be indicated as the parent, and the other process will be known as the child.

Fork had to be called twice. Once for child 1 (Process ID\_1) and once more for child (Process ID\_2). This is used to duplicate the process’ code. The difference between the parent and child is that the child will return a 0 while the parent will return a 1. This way, the program won’t run into issues identifying the two processes. The parent would also have access to the child’s PID and could therefore communicate with the child process, kill it, or wait for it.

This assignment only requires two pipes. One for each child. This is so that the parent could communicate with their respective child processes directly.

Wait(NULL) was used to make the parents wait for the child processes to complete themselves and terminate before continuing with the parents. This way, we could prevent zombie children processes from lingering around and free of memory for new processes later on. For this code, the wait() was only called twice. If the program is small, we wouldn’t notice a few lingering processes but once it’s much larger, proper memory management will be required and is overall good practice to use the wait() call to terminate any unnecessary children processes before continuing.