PA1 Report

Ethan Lamitie
2/13/16
CSI-285

This assignment was an interesting exercise in simulating interprocess communication. The three parts all encapsulated a different aspect of interprocess communication; the first demonstrated simply defining a process struct and simulating forking by copying a parent process five times. Each child receives a copy of the data and variables the parent does (much like real IPC) and each child increments the variable until it equaled 100. The second part was a simulation of pipes, and the third part was the same still but with emulated shared memory.

In the first part, each child has it's own copy of the global variable, and as a result there is no synchronization. Since the parent simply copies all of it's data to the children, and the children do whatever they want with their own local copies, nothing actually gets communicated to the other "processes".

The second part was a lot more sophisticated (and was much closer to an accurate emulation of IPC) in which pipes were used to allow the children to access the same global variable. This time there was no copying, because a data structure called pipe, meant to emulate an actual pipe, had one integer and two functions for reading from and writing to that integer. This way each child does an operation on the integer and it is "synchronized" across the multiple processes.

The last part was a simulation of shared memory rather than using pipes. The results of this part were identical to the pipes, because each process gets to change the same global, which is the same concept as with the pipes. To simulate shared memory there is a global flag, and each process checks the flag to see if the global is in use, and if it isn't, the process sets the flag to true, edits, and when it's done, turns the flag back to false. Obviously this is hard to accurately simulate since without doing actual multi-threading (not a simulation) we can't run functions in parallel so technically the processes won't get in each other's way. Still, the logic is sound and accurately represents what should happen.