

## Lab 4 Report

When I started this lab, I thought I was going to do a speed test between part 1 and 2 thinking they would execute the same however that was quickly disproven as the processes weren't printing "clarkson888". At first, I figured the parent wasn't executing and so p wasn't changing. I tested this hypothesis with a print that displayed "test" after the wait in parent and it still printed. This showed that it was running the parent but the change wasn't showing this led me to believe it freeing before the child could continue executing. This also did nothing to display the 888, however, it did solve another issue of printing the second "clarkson333" to the console rather than the command line, which means the code did return before the child had finished. I then looked up a solution to waiting for children to execute on stackoverflow (<https://stackoverflow.com/questions/19461744/how-to-make-parent-wait-for-all-child-processes-to-finish>). After changing it to fit my code, I ran it and it still didn't work. I then realized that maybe the parent and child run with different memory from one another and put a print for out in the parent which showed the 888 and then 7 seconds later the 333 displayed again. This made me realize what the lab was about.