Liam Robertson, Khushi Savaliya, Sachin Prabhakar, Kavya Sharma, Ethan Shenassa
CSEN 383 - Advanced Operating Systems
November 11th, 2025

Project 6 - UNIX I/O

When implementing project 6, only two main issues arose that took longer than average to debug and finally solve. The first issue we ran into was having Child 4 read from standard input to obtain the string that the user wrote in the terminal. During initial testing, we were getting EIO errors (Error Input/Output) and had trouble tracing the root cause of this error. Initially, we thought the problem had to do with how the standard input file descriptor was being handled in the multi-process simulation. We tried different methods to handle this, such as manually closing the stdin file descriptor for all processes except Child 4, using different functions to read the terminal, like fgets(), scanf(), and read(), but we landed on manually giving Child 4 access to the controlling terminal. However, in the end, we discovered that the Parent was exiting prematurely, which was causing the observed issues with respect to stdin and access to the controlling terminal. Regardless, we still gave Child 4 direct access to the controlling terminal to ensure that no other unintended errors occurred.

The second issue we ran into arose when the parent was reading from its end of the pipe. Initially, we had the parent call read() on its pipe, which was determined by select(), and the output we were expecting was for the parent to append its timestamp to the beginning of the message from the child. However, in our output.txt file, some of the messages did not have the appended timestamp. We discovered that this issue was occurring because the read() system call will read the entire buffer in the pipe, and there is a high probability that when the parent reads from the pipe, the child has already sent multiple messages to it. Our solution was to add more processing to the buffer received from the read() system call and split messages by the new line character, so each message sent to the parent has the parent's timestamp appended to the beginning of it.