CSCI 4730 Operating Systems | Project 1

Ruben Cristea
811427744

Program Structure and IPC

My program accepts the number of children processes and stores that number in nChildProc. The main loop of nChildProc will create all the pipes and initiate the forks. It will read the file for the specified amount for each child processes and all children will run asynchronously. Then the crash handling is checked (described in crash handling section). Afterwards, the parent will continue reading the children processes messages and increment the counter with the data received.

The IPC uses message passing where the child processes are the producers, and the parent is the consumer. The IPC is asynchronous and non-blocking so the child processes will send the message and continue (which is terminating normally). The parent will then read the children processes and use that data to increment the counter.

Crash Handling

Forks crashing are handled by waiting for the child process to finish with waitpid. The status is taken from the waitpid function and is checked against WIFSIGNALED. If the signal returned is an abnormal signal, a new fork will be created to run the same child process. It will keep retrying the child process until there is a normal signal returned from the process.