CSCI 4730 – Project 1

Gregory Woolsey

The purpose of this program is to read a file and print out the number of lines, words, and characters are in the file. This has since been modified to run using a multi-process structure and IPC (Inter-Process Communications). The way the modified program works is by taking in a user-defined number between 1 and 10, and with that number splits up the work as evenly as possible between that many child processes. This is done though a simple for-loop that creates a fork() each iteration. Each child then calls the word_count() function as given in wc_core.c, and counts the number of lines, words, and characters within a range as specified by the number of child processes. Once it has completed the counting, each child writes their collected information to their respective pipe that was created at the start of the program. Once each child has fully completed, the parent process concatenates all of the children's data together, and prints out the final result of the total number of lines, words, and characters in the given file.

The last argument of the program allows the user to specify the chance, in percentage, of a child process crashing. The program handles this through the waitpid() function. If the program encounters one of these crashed processes, it will run a conditional check with the second parameter and the WIFSIGNALED(*status*) option to check if the child was indeed terminated by a signal, as it is if aborted through a crash in the word_count() function. The program will then de-increment the main for-loop creating the child processes, which will allow a new child to be created picking up where the previously aborted one left off.