Jacob Craffey

CIS 452

Professor Wolffe

Multi-Process Text Analysis Design Document

The program begins by prompting the user for the number of files he/she would like to analyse. Then the program asks for *k* file names and places them in an array.

The program is now in an infinite loop until the user exits with defined protocol. For every file that the user has input, there are two file descriptors arrays initialized. One pair of file descriptors is designated for the parent to child process communication, while the other pair is designated for child to parent communication.

Next, there are *k* child processes being spawned concurrently equal to the number of files. Upon creation, the child logs that it has been created to stdout. The child closes its pipes so it has two unidirectional pipes for IPC. The child now waits to read the filename and keyword that will be sent from the parent. The user is now prompted to enter a keyword. If they keyword is ‘$$$’ (or ctrl-C), then the program shuts down gracefully.

The parent process closes certain file descriptors so it has two unidirectional pipes for IPC. The parent process writes the keyword and a filename to each of the child processes. The parent then logs its write to child and closes its writing file descriptor since it will not be needed for the rest of the program cycle.

The child processes have now read their keyword and file name from the parent. They open and read their file and count each time the keyword appears. The behavior of the count is case-sensitive and removes any character that is not an upper or lower-case letter. Once the read has reached EOF, the children write to the parent how many times the keyword appeared in their text. The children processes close the remaining file descriptors and exit successfully.

The parent process reads the word count from the child processes and closes its remaining file descriptors. The word count is stored in an array the size of the number of files.

The formatted word count is logged to stdout where the user can now see the amount of times their keyword appeared in each text document.

The user is prompted to enter another keyword and process begins again until ‘$$$’ (or ctrl-C) is entered.