CS 307 Programming Assignment (PA) - 1
Shell Command Execution Simulation in C

Ceren Dinç 28220

The Command:

man diff | grep -A 2 -e -q

Variables:

I choose the *diff* command that I find really useful. When I was taking the course CS305 in the previous semester, I was trying to compare my output for an assignment with the 'golden output' that is provided to us. However I had trouble comparing 2 different files because I was not familiar with the *-q* option of *diff* which reports only when files differ. As a result, I decided to use this command in my CS307 assignment.

As for the *man* part, *man diff* displays the manual page of the *diff* command which is responsible from the comparison of files. With regards to the *grep* part, *-A 2* option with grep is used to display two lines after the matching line, providing context to the match found. The "-e -q" part of the command cleverly tells grep to look for the exact text "-q", instead of turning off its output with the quiet option. In this script, we're interested in the "-q" text because it's used in the diff command to indicate that it should only tell us when there are differences between files.

Process Hierarchy (1a):

My `man` and `grep` processes have a *parent-child* relationship, where the `man` process is the parent and the `grep` process is the child. Also, they are designed to *run concurrently*. The `man` process is created first with a fork() call and it becomes the parent. Within the parent process, another fork() call is made to create the `grep` process, which becomes the child. The parent (`man` process) writes to the pipe, and the child (`grep` process) reads from the pipe.

Since there's no call to `wait()` before the `man` process starts writing to the pipe, both processes can run concurrently. The `grep` process will start processing the data as soon as there's something to read from the pipe, even if the `man` process is still running and writing to the pipe. The `wait(NULL)` call is made after the pipe ends are closed in the parent, which means it waits for the `grep` process to finish after it has done writing all its data to the pipe.

The `man` process is the parent, which executes the command `man diff` and sends its output to the `grep` process through a pipe. The `grep` process, which is the child, filters the output using the command `grep -A 2 -e -q`. It reads the input from the pipe connected to the `man` process' output, enabling these processes to run concurrently. This concurrent execution is achieved because the `grep` process starts processing data from the pipe as soon as it is available, without waiting for the `man` process to finish.