Anıl Şen - 29556

CS307 PA1 REPORT

Chosen command: man grep | grep -A 4 -n "\-r, --recursive" > output.txt

First, I have decided on which command I pick for MAN command to work on. It turns out to be GREP command since in this assignment, it is expected to deal with grep command, therefore, I thought it would be more helpful to dive deep into the characteristics of GREP command while also implementing the command itself in a C program. As a result, the manual page of GREP provides a broader perspective on what configurations could be done on the occurrences of the given search string. GREP command takes a text input and prints out the occurrences of the string that user will provide. In this case, GREP's input text is what MAN yields and the search string will be "\-r, --recursive". When it comes to options, because there exists 4 lines after the title of an option of the chosen command, which is "-r", "-A" and "4" options have taken place as options for GREP. Moreover, to have more detailed output and to see how far our findings are away from the starting point of the document, "-n" option helps for printing out the associated line numbers for the lines of the result.

Process Hierarchy

My MAN and GREP processes have sibling relationship.

As it can be seen in the code, SHELL process is forked and its child, which is to be MAN command, performing MAN execution. After that fork, SHELL performs it's second fork and its child is GREP process. Because both MAN and GREP are a child of parent SHELL, they are siblings.

My MAN and GREP process can run concurrently.

In the first fork, we have two processes going on which are MAN and SHELL process. After that SHELL process performs its second fork and creating its child, GREP. So, in total, we have 3 processes running. In SHELL process, SHELL waits for its two child, MAN and GREP, to finish because 1 wait statement is not enough since we want SHELL to wait for MAN and GREP processes to be done and inform the user that results can be found in txt file. Moreover, Between MAN and GREP, there is a second pipe which enables GREP to print out its statement, which is "I am GREP", after MAN prints out its own statement. So, we want GREP to wait a little bit until MAN's print so that the output on the console is as desired if GREP comes to the print statement first. Other than that, there is nothing that might prevent GREP or MAN to wait for one another. After printing to the console period is over, they are just left free to perform their operations through "execvp" function. Since if GREP comes to the "execvp" statement first, it waits until an input is provided by MAN and once the input is provided, two processes can run concurrently using "execvp" function, concurrent execution of these two is achieved.