Yiğit Karamanlı

28105

yigitkaramanli@sabanciuniv.edu

COMMAND

man grep | grep “-A” -A 3

grep command searches for *patterns* (e.g., a string or substring) in a given file and prints each line that matches the given pattern. -A <*NUM*> option prints *NUM* lines that comes after the initial line that contains a matching to the *pattern*. I picked this command and option because while thinking on a possible solution for the assignment and how to print adequate number of lines I read the whole manual page for grep command and when I found the “-A” option the feeling of success gave me the idea that I had to select this option.

PROGRAM HIERARCHY

The program starts with printing the required “SHELL” message to the terminal under the parent process. After that 2 file descriptors are created (as an array named fd) and piped. If the pipe is successful the first child process is created with fork().

The first child process starts by printing the necessary “MAN” message to the terminal. After that, using the dup2() command, the reading file descriptor (fd[0]) is closed and STDOUT_FILENO file descriptor is closed and replaced (duplicaded) by the writing file descriptor (fd[1]). Following that, required arguments (“man, grep”) are put into an array and executed using the execvp() command, writing the manual page of grep command into the file descriptor.

During the execution of the “MAN” child, the parent process waits. After the execution is complete, another child process is created using fork().

The second child process starts execution by printing the necessary “GREP” message to the terminal. After that, a new file descriptor, named new_fd, is created, writing file descriptor (fd[1]) is closed, STDIN_FILENO file descriptor is replaced by the reading file descriptor (fd[0]) and STDOUT_FILENO file descriptor is replaced by the newly created descriptor (new_fd) using the dup2() command. The required arguments (“grep” , “-A” (used as “\\-A” in order to solve the special character problem), “-A” , 3 (number of lines to be printed following the match)) are put into an array and executed using the execvp() command, printing the required output into the newly created file named “output.txt”.

During the execution of the “GREP” child, the parent process waits and the file descriptors (fd[]) are closed. After “GREP” child process finishes execution, necessary “SHELL” message is printed on to the terminal and the program terminates (returns) by value 0. Execution is complete and the required output can be accessed through the “output.txt” file.