Duygu Tümer 24842

“ls” command lists information about the files (the current directory by default). It sorts entries alphabetically. “-m” fills width with a comma separated list of entries. I picked this command and option because, “ls” is one of the most known one. “ls” gives the files and “-m” puts them in a list. We can always use when we want to see our files in a listed entry.

I use two “fork()” calls because one for the man and the other one for the grep. For the man process I use first “fork()” call at the beginning of the main function. Before “fork()” call, I create int fd[2] and put fd in a “pipe()” call because I want to connect the outputs and inputs of my child processes.

After the first “fork()” call, there will be three possibilities: If fork fails, else if it is equal to 0 (child process) and there will be else part (parent process).

In this child process, I run program and command with “execvp()”. For this, first I write “close(fd[0]);” and “close(STDOUT_FILENO);” and “dup(fd[1]);” to send the output to the pipe.

In the parent process, I use the second “fork()” call because now, I want to do grep process. In this grep process, again, there will be three possibilities: If fork fails, else if it is equal to 0 (child process) and there will be else part (parent process).

In this child process, first I write “close(fd[1]);” and “close(STDIN_FILENO);” and “dup(fd[0]);” because now, I do not want to write and I want the grep to read from the pipe. Then, I run grep with “execvp()” call.

In this parent process of the grep process, first I write “close(fd[1]);” because it has to close the writing. If we do not write this command it won’t execute(grep does not stop writing/running). At the end, I use two “wait()” calls because one for the man and the other one for the grep process.