

I chose "man df | grep -a" as my command to be simulated. As we have seen in the lectures, man command shows the information about the command that we give as input to the terminal. Then, we use grep to find the specific lines of information that we search for. I chose the command "df", which shows the statistics of free disk space of the filesystem or the filesystem of which the file is a part of. Then, I chose "-a" to be searched for in my grep part, which shows the mount points (directory that is linked to another system).

For this purpose, I designed my simulation c code as the following:

- Created the main, which acted as the SHELL process. It acted as the parent process of the command.
- Created the file descriptor to use the pipe function. Pipe is used to read and write to anonymous files by using fd[0] and fd[1].
- Created the "output.txt" file, which will be used to write the final output.
- Used "fork" to create the first child process. If the fork operation is not successful, it returns a value less than 0 and returns an error message. If it is equal to 0, it means that fork is successful. In this else if part, the pid of the MAN process is printed.
- Used "dup2" function to copy the file descriptors at the left side of the parenthesis to the given right side file descriptor. By using that, we are able to give the output of this MAN process to be used by the write end of the file descriptor.
- Created an argument list to execute the command by using "execvp" command.
- Then, the second child is created with fork function from the parent for the "grep" command. The same steps for the error handling is performed and printed the pid of the new child, if the fork is successful.
- The output file which was created at the beginning of the program is used in dup2 function, which directed the output to the given txt file. Again, used "execvp" command by giving the argument list.
- As the last step, used "waitpid" functions to wait for the child processes to end. When they finish, SHELL process again gives the pid and prints that the result is written to the output.txt file.