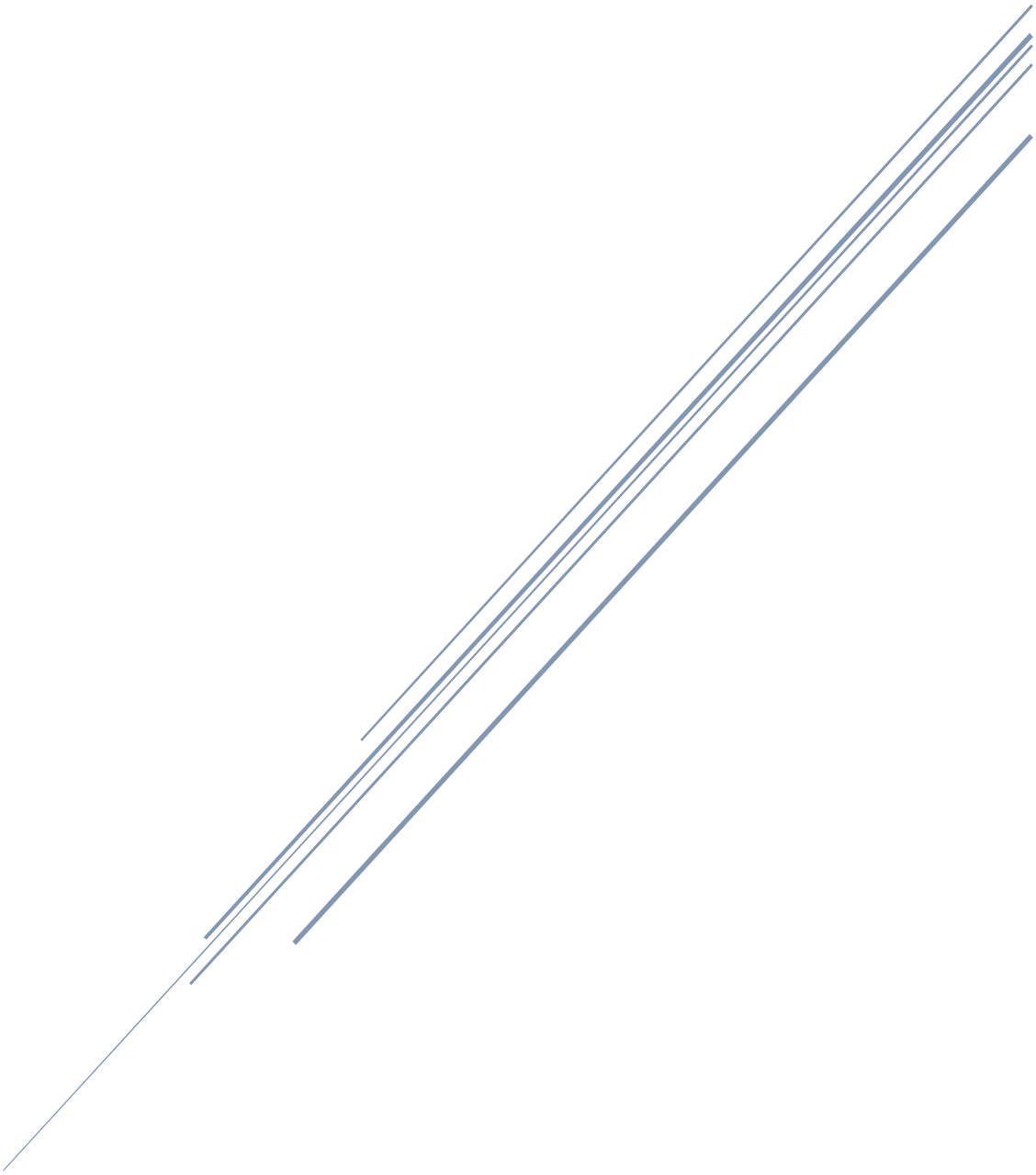
PA1 REPORT

CS 307 – Operating Systems



Due 6th November, 2023

İpek Akkuş - 30800

Introduction to the Problem

For this assignment, we are asked to code the pipeSim.c program that simulates piped execution of man and grep commands on the shell. We needed to randomly choose a *command* from the ***manual*** page and *option* name from that page to find it with ***grep*** command to simulate the command template **man** <*command*> <*-extra_option*> **| grep** <*option*> <*-extra_option*> **> output.txt.**

Chosen Command Details

I chose the *command* as "***diff***" and *option* as *"-**i**"*, therefore the command has become

"*man diff | grep -i -A1 -m1 > output.txt ".*

"Diff" command is used to compare objects and *"-i*" flag is used to ignore-case, more specifically ignore case differences in file contents. I randomly picked the command "*diff*" by plugging all options provided in the PA1 assignment document to the wheel of luck on the web and spinning it. The diff command appears on the wheel at the end. I picked *"-i*" option as my name starts with the letter i and the capital letter version is different in English alphabet. -A1 and -m1 added to satisfy the requirements of the Programming Assignment 1.

"*man diff | grep -i -A1 -m1 > output.txt "* means that the program opens *manual* file for the command *diff* and pipes the output with the *grep* command. With the help of grep command, description of the *-i* option is searched in the manual page of the *diff*. The description of *-i* option in the *manual* page of *diff* is given below:

-i, --ignore-case

ignore case differences in file contents

The description contains 2 lines. Thus, the program should prompt both lines, not only the first line. That's why, -A1 added to the simulation command as the <*extra_option*>. -A is used for "after" and 1 stands for 1 line. "-m1" is used to guarantee that the program stops searching for the other -i's as it limits maximum occurrence to 1.

Program Flow and Structure

My program has 3 processes in total and they are different in terms of hierarchy and purposes. The program contains 1 shell process as a parent and two children processes of this parent Child1 and Child2, which are for *man* and *grep*, respectively. Read end and write end of file descriptor play role in carrying information.

The shell process starts execution and then it forks for child1, which calls man command. Child1 executes for the opening manual page of diff command, then it sends the output to the write end of the file descriptor. Therefore, output produced in Child1 (man command) is transferred through the pipe to Child2 (grep command) to execute the search for description of chosen option, *-i*. There are some limitations in PA1 homework document, iso in order to satisfy that all -A1 and -m1

extra options are added to grep command and executing by the Child1, as well. See figure 1 for visualization. In that schema, inner arrows stand for reading step and outer arrows stand for writing step, for each process box.

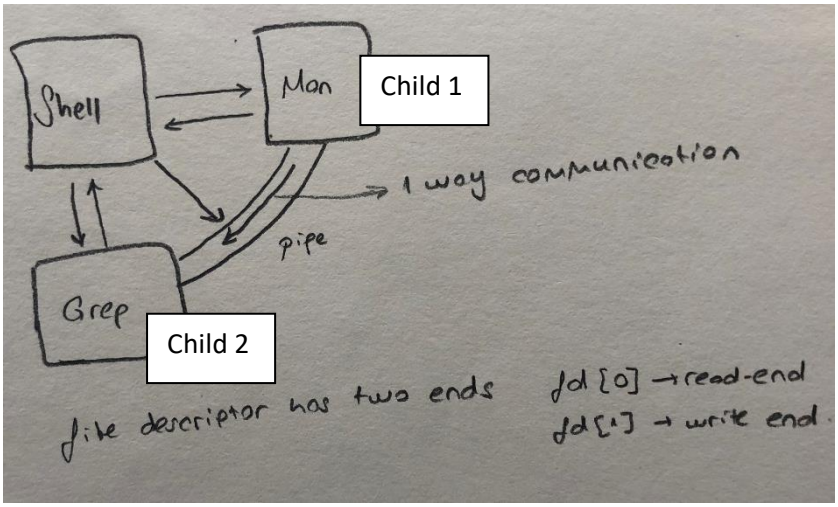


Figure 1. Schema for the hierarchy of processes

The processes run concurrently as the children are running in parallel. Neither Child2 runs without waiting the Child1 nor Child1 runs without waiting the Child2.

- 1: My mand and grep processes has parent-child relationship.
- 2: My man and grep processes has sibling relationship.
- 3: My man and grep processes has another kind of relationship.
- a: My man and grep processes can run concurrently
- b: One of my man and grep processes wait for the other to finish

Figure 2. Criteria from PA1

Thus, according to the criteria in Programming Assignment 1 (See figure 2 above), my program is 2a. They are children of the same parent, as detailly explained above and also they can run concurrently as the Child1 and Child2 processes are running in parallel without waiting each other. However, at the end the shell process waits for boths of them to finish.

Then, the result is printed to the file named output.txt.

My program correctly interleaves as the initial process, shell, waits until Child1 and Child2 processes to finish execution.

You may see comments on the pipeSim.c file for better explanations with reasons line by line.