Yigit Kaan Tonkaz

29154

Command Formation & Argument Selection

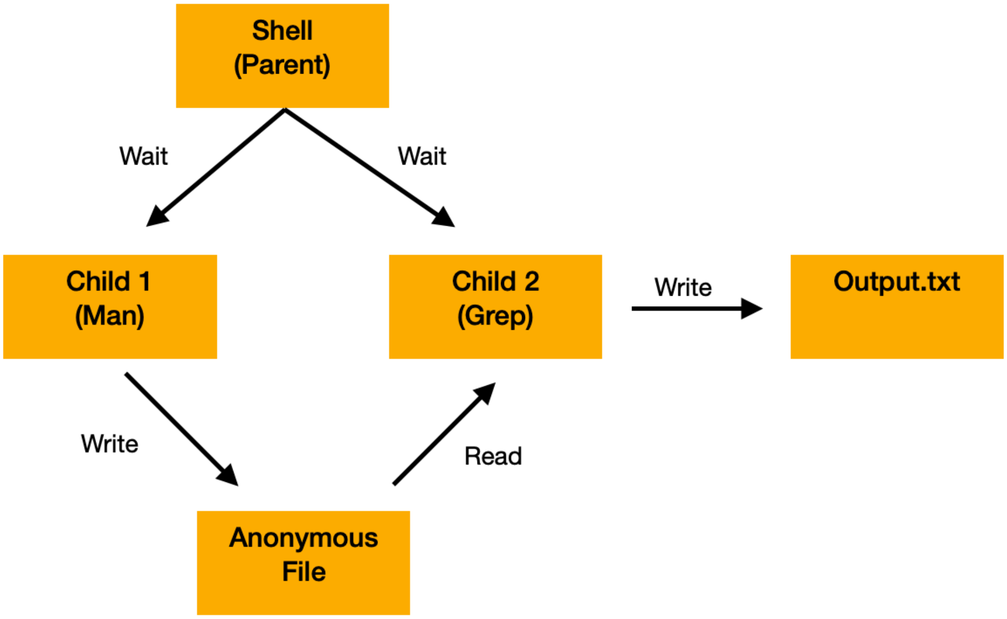
```
man ls | grep -A 1 -m 1 -- "-h" > output.txt
```

“ls” command will list information about the files and directories which are in the current working directory. In other words, it provides an easier way to view the contents of the current directory. Moreover, some of various options of this command can provide several simplicities such as sorting or filtering and displaying. I chose “-h” option which is lists files in a human readable format like displaying the size of file (kb, mb). I picked this command and option because I think that they are the fundamentals and used most of the time and also -h option is a kind of interesting due to display format. Furthermore, I used -A 1 which decides the number of lines are going to be displayed after a matching occurs, and -m 1 that notifies the grep to take only one occurrence, and – which is used for showing that any arguments should not be treated as options so I can separate “-h”.

Report Format

My program is 2a. Man and grep processes are works like a siblings and they run concurrently. They are siblings because the parent process, which is the initial shell process, forks two child processes, one for running the man command and the other for running the grep command. The parent process is responsible for managing the communication between the two child processes through a pipe. In other words, both man and grep processes generated by the shell that’s why they are siblings. They can run concurrently because man and grep processes are being forked separately. Both processes run concurrently, with child 1 (man) writes the output of man ls to the pipe's write end and child 2 (grep) reads its input from the pipe's read end. Also, the parent process (shell process) waits both child to complete using waitpid. This waiting mechanism confirms that parent process is going to progresses after both child have completed their tasks. In short, forking the children separately and allow communication through the pipe confirms the concurrency.

Process Hierarchy and Concurrency



The main process Shell creates 2 child which are man and grep. Then, pipefd[2] file descriptor is going to piped with pipe function which connects both child. Then pid1 is forked that starts the man process and man is executed and result is written by dup2 to end of pipe. After that pid2 is forked to start grep process. And then, output file is created and with the dup2, result acquired from read end. Then it is executed and result from grep is written to output file. Finally man and grep is waited and results printed.