

Sharanya Sudhakar

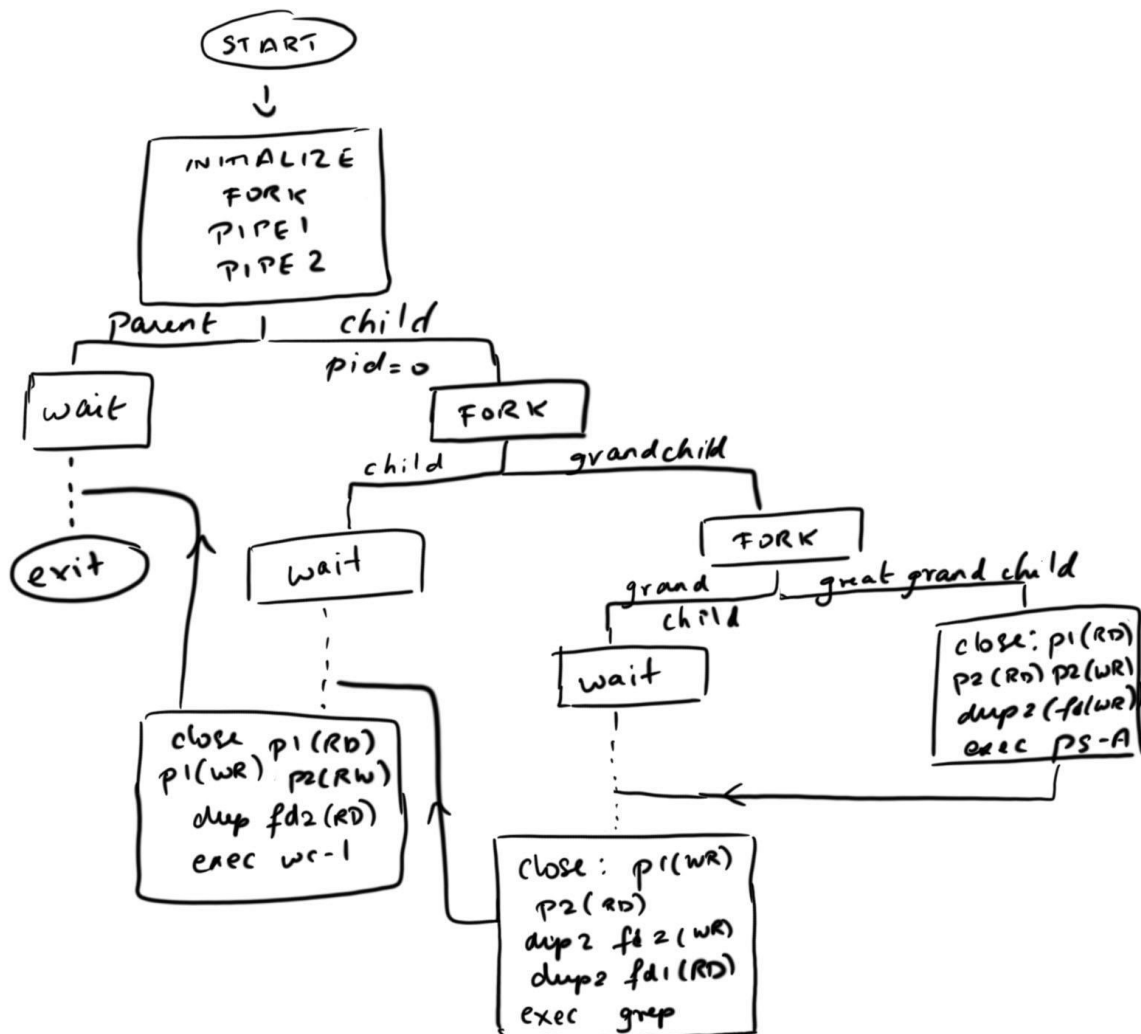
CSS430

Project 1

## Program Documentation

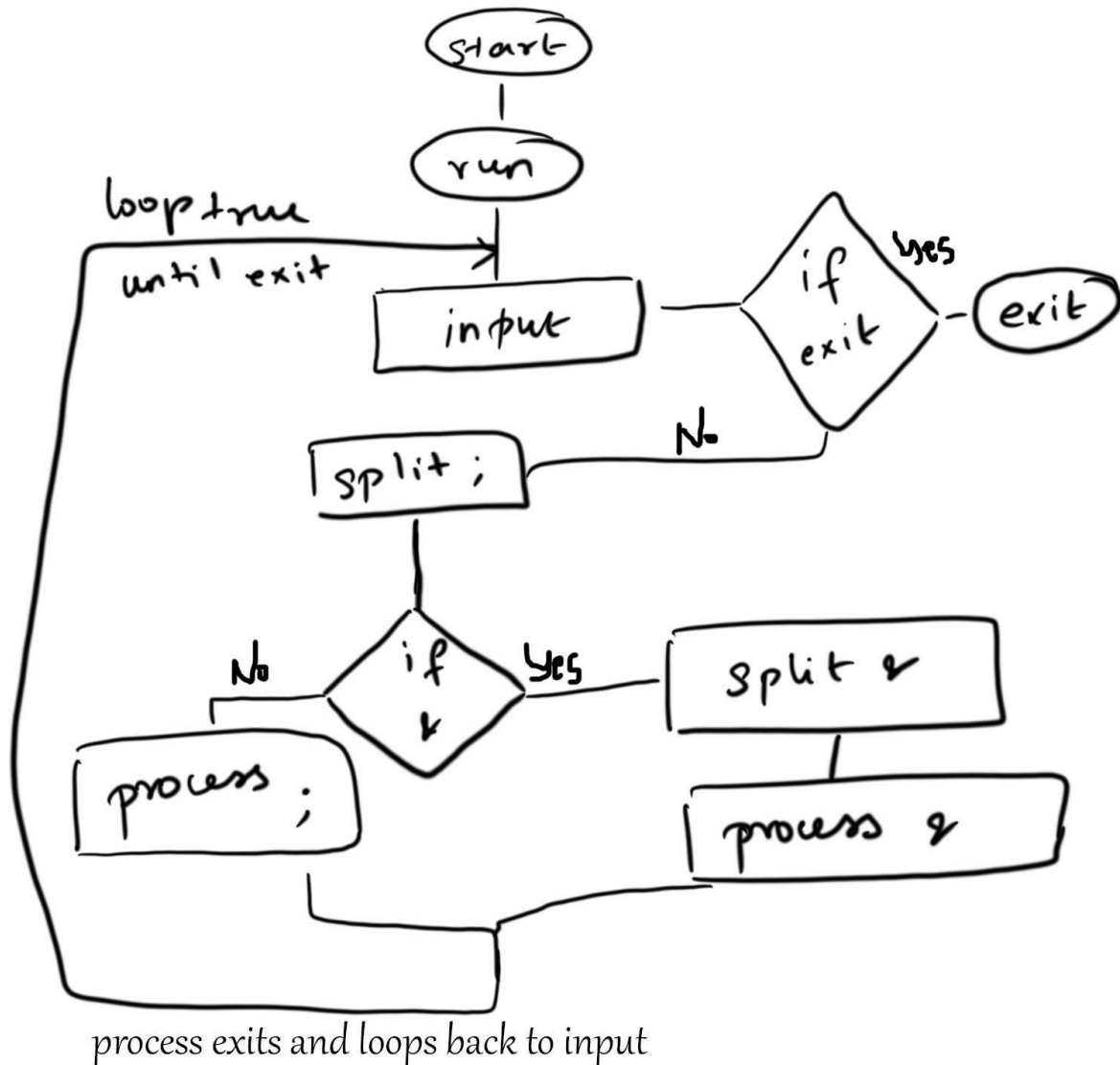
process.cpp

This program simulates the c++ execution of the following shell commands `ps -A | grep kworker | wc -l`. In this `kworker` can be replaced by a run time argument of the user's choice. To execute system calls, this program utilizes the `fork`, `pipe`, `execlp`, `wait`, `close` and `dup2` from `unistd.h` and `sys/wait` built in directories. The logic behind the program is to have it execute forks where the commands are executed by the child process, while the parent waits. The output from the child is then piped to the parents input and so on until all commands are processed.



MyShell.java

This program is a java simulator of the shell execution. The two condition to be process by the shell is Thread OS's parallel and sequential command execution identified by the delimiter. The command is read into the program, split first for sequential execution then branched off for parallel execution if the & symbol is identified.



#### Testing the Shell:

```
javac MyShell.java
```

```
java Boot
```

```
→ / MyShell
```

```
Shell[1]% PingPong abc 100 ; PingPong xyz 50 ; PingPong 123 100
```