P1. Bash-Like Shell

Concepts used

pipes, tee (system call), processes and fork, execv, path searching

Design

The program is a shell program that runs commands. The commands are read from stdin and the output is written to stdout or a file depending on what destination is specified. The shell supports many types of commands (listed in the section below). The shell can run programs present in directories mentioned in $PATH. The shell allows the user to run programs in the foreground or background (using &). Redirection is implemented by duplicating file descriptors before executing the commands and double/triple pipes are implemented using the tee system call (to duplicate the contents of a pipe without using read).

Files Included

shell.c, Makefile

How to run

- to compile, use make or make all
- run with ./shell
- all commands require a space between each token *(arguments, symbols like comma, pipe, etc.)*
- *types of commands supported*
    - single commands (`ls`, `cat`, `touch`, `clear`, ...)
    - commands with IO-redirection (`ls > file1`, `ls -la >> file2`, `cat < file3`)
    - commands with single pipes (`ls | wc`, `cat file1 | sort | uniq | wc`)
    - commands with single pipes and IO-redirection (`ls | wc > file3`)
    - commands with double pipes (`ls || wc , grep hi`)
    - commands with triple pipes (`ls ||| wc , sort , uniq`)
    - daemon commands (`daemonize <cmd>`)
    - background commands (`cat file1 &`)

Limitations

- the shell is unable to run executables like a.out, etc. from the current working directory

Ashwin Kiran Godbole (2018B5A70423P)
Samarth Krishna Murthy (2018B2A70362)