COP290 Assignment 2 Report

Navneel Singhal

January 2020

Description

The assignment was to implement a shell in C.

The following commands can be used:

1. `make` : This command generates the executable for the program.
2. `make run` : This command runs the executable, generating it from the source code if necessary.
3. `make clean` : This command can be used to clean up the executable.

Build Instructions

The executable can be built using either of the given options:

1. `make` (In the case that we need to run the program, it should be followed by `make run`)
2. `make run` (This is self-sufficient, and also gives the result of a run of the program)

The cleaning can be done using `make clean`.

Working of code

1. `Makefile`: It has four targets: `all`, `exe/shell`, `run` and `clean`. `all` has one dependency which is `exe/shell` and has an empty recipe. `exe/shell` generates the executable using `gcc`. `run` has a dependency which is `exe/shell` and it runs the file `exe/shell`. `clean` cleans up the `exe` folder.
2. `src/2018CS10360_sh.c`: It consists of the code for the shell in `C`. It has the following functions:
   (a) `cleanstring`: This function replaces all tabs by whitespaces, removes initial and final spaces and duplicate spaces.
   (b) `runrawcommand`: This command tokenizes the given command with the delimiter as the piping operator. Then it runs each command using the function `runandgetdestination`.
   (c) `runandgetdestination`: This function runs a given tokenized command (without pipes) given the source of input and information about whether it is the first or the last command, and returns the file descriptor of the destination of this command. It also accounts for the presence of redirection (and the format for redirection in a given command should be `<command> [< infile] [> outfile]` (i.e. if both redirections are present, then the input redirection should be mentioned first).
   (d) `main`: The main function runs the shell by running an infinite loop where it keeps on taking the input and running the command using `runrawcommand` till the user types `exit`, whence it exits from the shell.
   (e) `clear`: This is a helper function to free two pointers at a time (just for ease in debugging unfreed memory allocations).