CS214 Systems Programming Project 2: My shell by Romil Patel (rp1247).

⌘ Purpose:  
 The purpose of this assignment is to design and implement a simple command-line shell.

This program reads user input as commands, then executes them using the **‘execvp’** system call. The **‘parseSpace’** function is used to split the input string into an array of strings, where each string represents an argument in the command. If the command contains a pipe character (|), then the **‘parse\_pipe’** function is used to split the command into two separate commands, which are then executed using **‘execute\_command\_piped’**. The **‘read\_command’** function reads a command from the user, and the **‘execute\_command’** function executes the command. The **‘execute\_command\_piped’** function executes two commands that are piped together. Finally, the **‘print\_cwd’** function is used to print the current working directory.

Overall, this program provides a basic shell that can be used to execute commands in Linux.  
  
 ⌘ Test Plan:

* To run the program, simply call the make command: **make**
* Once you run this command it will produce the object file of mysh. To execute the file, simply call: .**/mysh**
* The program also includes a few built-in commands such as:
  + **‘pwd’** for printing the current working directory.
  + **‘cd’** for changing the current directory.
  + **‘exit’** for quitting the shell.
* To check if my pipe is working, I created a text file with 10 fruits name and use command: **ls text | grep a** (Note: Once you get the output, press ctrl + c to get out of command and run back **./mysh** This is the error I couldn’t figure out and I understand if I get less point for this).
* To check if my home directory is working, I then use command **pwd** to check my current working directory and when I try command **cd** it will automatically switch the working directory to the user's home directory.
* To check my wildcards, First I use command **ls** to check the files in my directory, then I use **ls \*.c** to check only files that end with .c and the output was correct.

⌘ Extensions:

**Home directory:** The user's home directory is stored in the environment variable HOME and can be accessed with the function getenv("HOME"). If the cd command is used without any arguments, it will automatically switch the working directory to the user's home directory. Additionally, any tokens starting with ~/ are treated as being relative to the home directory and the ~ is substituted with the actual home directory path.

**Directory wildcards**: Asterisks may occur in any segment of a path. For example, \*/\*.c references files ending with .c in any subdirectory of the working directory (excluding files and subdirectories that begin with a period). You may allow more than one asterisk within a path segment, but this is not required.