**My Shell Program**

**Description:**

An attempt to create a \*nix shell that acts as exactly how shell works: taking in arguments and execute input. Shell is able to execute built-in commands, redirection, pipe, and background execution.

**Built-in Functions:**

* cd: change the current working directory.
  + Arguments:
    - “.” For current working directory
    - “..” Go up one directory
    - “*path”* Move you to desired path
* clr: clear the screen
* dir: list the contents of the directory
* environ: list all environment strings
* echo: will print out the input followed by a new line. Argument is a string of characters that user wants to print out
* help: display user manual using more filter (hit Enter to navigate through the manual)
* pause: pause the shell until Enter is hit
* quit: quit the shell

**External Functions:**

Shell is able to execute all commands that of \*nix shell.

See <https://www.geeksforgeeks.org/basic-shell-commands-in-linux/> for more information

**Redirection:**

Shell is able to take in input from a file and write to a file using this function. User will have to include special character like “>” and “<” so shell knows to execute redirecton:

* > - put the name of the file after the command then shell will create a file and write the result of the command to that file. If file exists shell will overwrite the file.
* >> - same as “>” but if the file already exists, shell will append the result to the file instead of creating new one
* < - shell will open the file and read whatever in the file be the argument to execute the command

Format: command argument1 argument2 .. < filename1 > filename2

**Piping:**

Shell will take the output of 1 command to be the input of the next command. The format has to follow this order:

Command1 argument | command2 argument

**Background Execution:**

Shell will execute command in the background and user will still be able to use shell in the meantime. User will have to add “&” at the end of the command line to be able to use this function.

Format: command argument1 argument2 .. &