SOURCE CODE

```
/*
William Kelley
CS415-Operating Systems
Command Line Interpreter

Sources:
https://stackoverflow.com/questions/16285623/how-to-get-the-to-get-path-to-the-
current-file-pwd-in-linux-from-c/16285723#16285723
*/

#ifdef __unix__
    #define IS_POSIX 1
    #define _BSD_SOURCE
#else
    #define IS_POSIX 0
#endif

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <iostream>
#include <iomanip>
#include <string>
#include <iostream>
#include <fstream>

using namespace std;

ifstream fin;
ofstream fout;

string showPWD();

int main (int argv, char **argc)
{
    fstream file("output.txt", ios::in | ios::out | ios::app);

    cout << "--- Wheel's Shell ---\n\n\n";

    string input;
    string command;
    string pieces[2];

    while(input != "exit") {
        cout << "wheelsh::";

        getline(cin, input);

        command.clear();

        if(input[0] == '>') {
            pieces[0] = input.substr(0, input.find(' '));
            pieces[0] = pieces[0].erase(0,1);

            if(pieces[0] == "cd") {
                pieces[1] = input.substr(input.find(' '));
```

```
                pieces[1] = pieces[1].erase(0,1);
                int chDirSuccess = chdir(pieces[1].c_str());
                if(chDirSuccess >= 0) {
                    cout << showPWD() << endl;
                }
                else {
                    cout << "chdir() failed" << endl;
                }
            }
            else if (pieces[0] == "pwd") {
                file << showPWD();
                cout << "pwd printed to output.txt" << endl;
            }
            else {
                cout << "No valid command entered" << endl;
            }
            input.clear();
        }
        else if (input[0] == '<') {
            input.erase(0,1);
            cout << "Requested file: " << input << endl;
            fin.open(input.c_str());
            getline(fin, command);
            cout << command << endl;
            fin.close();
        }
        else if (input == "exit") {
            cout << "Good bye!" << endl;
        }
        else {
            cout << "No valid command entered\n";
        }
    }

    return 0;
}

string showPWD() {
    string Result;
    if (IS_POSIX == 1) {
        char buffer[500];
        FILE *output;

        // read output of a command
        output = popen("/bin/pwd", "r");
        char *pwd = fgets(buffer, sizeof(buffer), output);

        // strip '\n' on ending of a line
        pwd = strtok(pwd, "\n");

        Result = "\nPath info:";
        Result += pwd;
    }
    return Result;
}
```

TESTING DOCUMENTATION

```
Step 1) Unzip files submitted alongside the PDF
Step 2) Place all files in the same folder locally
Step 3) Use command below to run the program locally
./command_line
Step 4) Supported commands are:
    >pwd
        - will print working directory
    >cd /path/to/change/directory/to/here
        - will change directory to any correctly entered directory
    <{DESIRED_FILE_NAME}
        - will output the contents of the first line of any .txt file

*Still to implement, running code from input files
*Creating file from >>, currently my program creates file if it doesn't exist
    - If file does exist, it will append to that file.
*Use requested structure
*Use fork, exec, and wait system calls


--- Wheel's Shell ---


wheelsh::>pwd
pwd printed to output.txt
wheelsh::>cd /home/parallels/school/CS415-Operating Systems

Path info:/home/parallels/school/CS415-Operating Systems
wheelsh::>pwd
pwd printed to output.txt
wheelsh::>cd /home/parallels/school/CS415-Operating Systems/Terminal Stuff

Path info:/home/parallels/school/CS415-Operating Systems/Terminal Stuff
wheelsh::>pwd
pwd printed to output.txt
wheelsh::<input.txt
Requested file: <input.txt
cd /home/parallels/school/CS415-Operating Systems
wheelsh::<input2.txt
Requested file: <input2.txt
pwd
wheelsh::<input3.txt
Requested file: <input3.txt
cd /WHEEEEEL
wheelsh::exit
Good bye!
```

SCREENSHOT OF RUNNING CODE