Ashton Callender
OS Batch Simulation

WRITTEN AND TESTED ON:
This program was written to be run on linux, it should work on windows subsystem but has not been tested. Only has been tested on Ubuntu.

I believe the comments will explain most questions. I used the notes to do the main loop but then I kind of forgot, so I did a lot of system calls to basically bash.

COMPILER INFO:
This does not implement a linker or loader; everything is done with GCC.

BIGGEST STRUGGLES:
Honestly Remembering how to use pointers properly and the constant forgetting of semi-colons. Also learning how to use some C functionality with the system calls. I learned a lot though and am having a blast with the system() function.

BATCH MONITOR CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>

//THIS IS A PROGRAM FOR RUNNING OTHER C PROGRAMS AS BATCH PROCESSES
//IT WORKS FOR LINUX AND WAS TESTED USING UBUNTU

//I have been using too much python lately and had to come back and add
nearly every semi-colon after atttempting to compile this program.



//Functions
int MainMenu() {
   printf("\n");
   printf("i. List Jobs\n");
   printf("ii. Set Jobs Directory\n");
   printf("iii. Compile and Run a Specific Job\n");
   printf("iv. Compile and Run All Jobs in Directory\n");
   printf("v. Shutdown\n");
   printf("vi. List Program Options\n");
   printf("vii. HELP\n");
```

```c
    printf("\n");
}

int readDir(char * dir) {



    FILE *fp;
    char path[100];
    char directory[100] = "";


    //Build file path access string
    strcat(directory, "/bin/ls ");
    strcat(directory, dir);

    //Opens Directory for reading
    fp = popen(directory, "r");
    if (fp == NULL) {
        printf("Error: Failed to run the command\n");
        exit(1);
    }

    //Formatting
    printf("List of Files:\n");
    printf("\n");
    //Reads directory
    while (fgets(path, sizeof(path), fp) != NULL) {
        printf("%s", path);
    }

    pclose(fp);
    printf("\n");
    return 0;
}

char * setDir() {
    char * dir;

    printf("Please Type a Directory:\n");
```

```c
   scanf("%s", dir);
   //clear buffer
   getchar();


   printf("Is this the Directory you mean't to type?: %s\n", dir);
   printf("Type y or n to Confirm: ");

   char answer = getchar();
   getchar();

   if(answer == 'y' || answer == 'Y') {
       char * returnDir = &dir[0];
       printf("%s : THE WHOLE STRING", returnDir);
       return returnDir;
   }
   else {
       printf("You have declined to change the directory returning to Main
Menu\n");
   }
}

int runProgram(char * dir) {
   FILE *fp;
   char path[100];

   printf("Please Input a Program to Run from the Set Directory:");
   char * answer;
   scanf("%s", answer);
   //clear buffer
   getchar();

   //For storing the full command
   char function[200] = "";
   //Select Compiler
   char compiler[] = "/bin/gcc";
   //Program To be Run

   strcat(function, compiler);
   strcat(function, " ");
```

```
    strcat(function, dir);
    strcat(function, "/");
    strcat(function, answer);

    system(function);
    printf("\n");
    printf("Program Output:\n");
    system("./a.out");

}

int runAllDirectoryPrograms(char * dir) {
    char function[200] = "";

    //Honestly its basically bash, I hope this is how you want it done\
    //It was so easy to handle this part
    strcat(function, "for f in ");
    strcat(function, dir);
    strcat(function, "/*.c; do gcc $f; ./a.out; done;");

    system(function);

    return 0;
}


int main() {

    //Padding
    printf("\n");

    //Variables
    char * dir = "./jobs";

    int choice = 0;
    bool executing = true;


    while (executing) {
        bool validSelection = false;

        while(!validSelection) {
            //Show Main Menu and allow selection
            MainMenu();
            printf("Please Select an option: ");

            //Read an option from the User

            scanf("%d", &choice);
            //Padding
            printf("\n");


            //Verify valid selection
            if (choice > 0 && choice <= 7) {
                validSelection = true;
            }
            else {
                printf("\n");
                printf("Error: You have Entered an Invalid Selection Please Enter a Number from the List of Options\n");
                printf("\n");
            }
        }

        switch(choice) {
            //List Jobs
            case 1:
                readDir(dir);
                break;

            //Set Jobs Directory
            case 2:
                dir = setDir();
                break;

            //Compile and Run a Specific Job
            case 3:
                runProgram(dir);
                break;

            //Compile and Run All Jobs in Directory
            case 4:
                runAllDirectoryPrograms(dir);
                break;

            //Shutdown
            case 5:
                printf("Shutting Down\n");
                //Honestly I did this just to be annoying, pretty sure you
meant to shutdown the program not the computer
                system("shutdown now");
                break;

            //List Program Options
            case 6:
                MainMenu();
                break;

            //HELP
            case 7:
                printf("This is a Batch Simulator for Running C Programs as
Batch Processes\n");
                printf("\n\nIf you need help with this please press CTRL-C
and do not touch this program\n\n");
                break;

        }

    }

    //Padding
    printf("\n");
}
```