Shubham Sharma

Section L

CPRE 308

Project 1: Report

This project is my very own version of the UNIX shell. It is capable of running any built in UNIX command and a lot more. This shell can run processes in the background, show the exit statuses of each process, and also has a few built-in commands which make it a pretty useful tool.

The project follows the given specs up to all the point that ALL the given test cases in the specification sheet works flawlessly. The design of this project was made to be very modular, in the sense that new features and easily be added, current features can be modified with ease. The whole project is heavily documented for any assistance.

[Note: This submission does not include the extra credit]

The *makefile* handles, compiling the program and cleaning up code. To run the program, execute:

```
make clean && make && ./shell "userName> "
```

There are multiple functions in this program designed to handle and execute code. The are listed and described as follows:

- ***int main(int argc, char** argv)***

This is the main function of the program. It handles all the arguments to the given specs. Will return 0 when 'exit' is typed.

- ***void printUserName(char **argv);***

Prints the username of the shell. Handles a custom prompt *'-p'*

- ***void executeCommand();***

Executes given built-in-commands and all Linux commands. This method is responsible for exiting the program.

- ***void printPID();***

Prints the current PID of the current process.

- ***void printPPID();***

Prints the PPID of the current process.

- ***void printPWD();***

Prints the path of the current working directory.

- ***void printCD();***

Handles and prints the *cd* command. cd moves the current working directory to the users HOME directory. Also, it is able to change directory from its relative path.

- ***void printSET();***

Sets and resets an environment variable.

- ***void printGET();***

Prints the given environment variable.

- ***void buildArgs(char** list);***

Builds the *args* array needed for *execvp* command. It does this by splitting the whole command to usable sections in to a char array[].

- ***void handleUnixCommand(bool isBackground);***

Handles all UNIX commands. It will check if the process exits with or without an error. Also, it handles processes if it is to run in the background.

- ***bool isBackground();***

Returns true if the user wants to run the command in a background process. Else the functions returns false.

- ***void checkBg();***

Handles the termination of a background process. Background process can end with or without an error, which is handled accordingly.

Conclusion:

This project has taught me a lot on how intricate and complex the native UNIX shell is, and how effortlessly it handles any form of errors. I learn a bunch on how processes work, terminated and created. I'm now familiar working with arguments, *execvp*, and environment variables. Looking forward for the next project/lab in this course.

For any issues in running the program or in grading the project, please do not hesitate to contact me at: shubham@iastate.edu