

Dalton Avery, Casey Porter, Kirk Sarrine

Dr. Bo Sun

COSC 4302

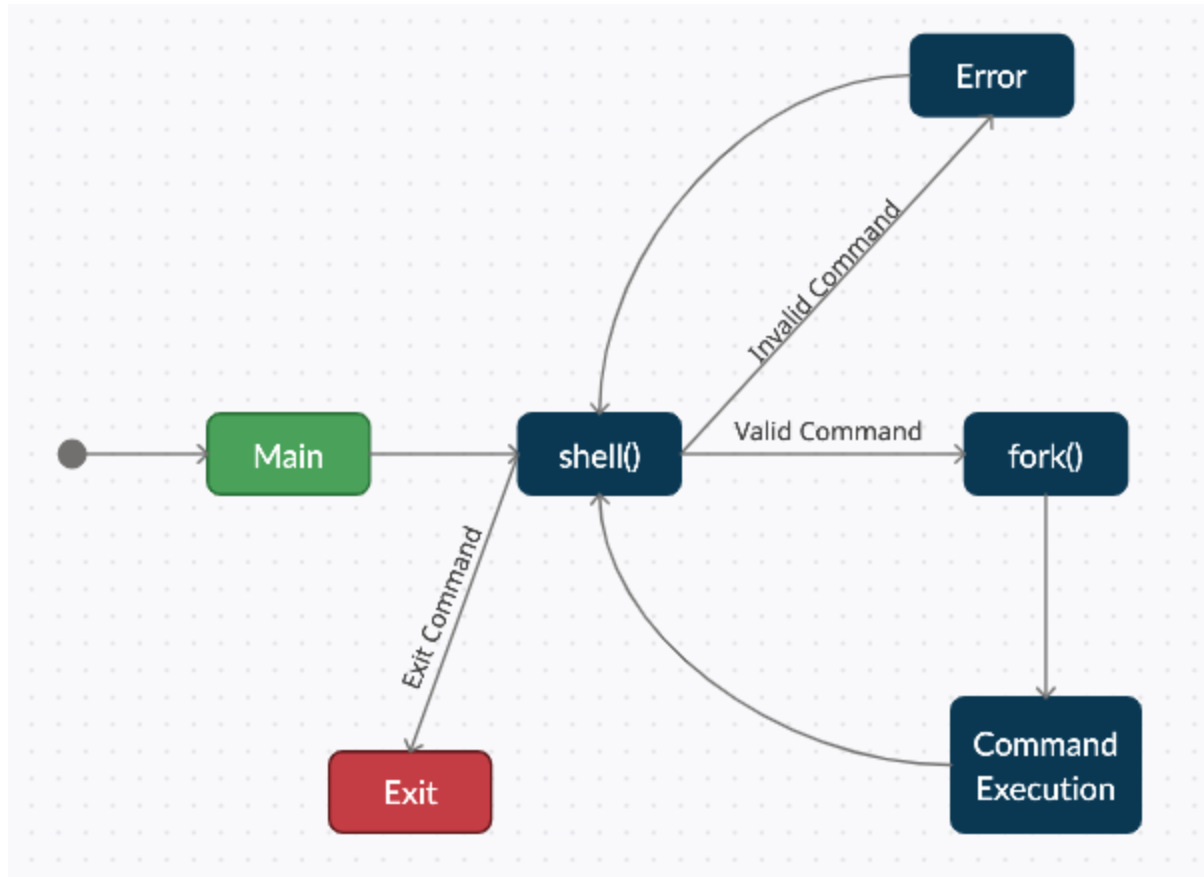
A Simple Shell Group Project

## Project Report

The Simple Shell Project uses a high level language, in our case we used C++, to implement a shell within basic functionality including sending commands and using arguments. Using the template provided to guide the project, our program includes a command class that is used for convenience when parsing file paths as well as reading and printing commands. Our methods for functionality utilize the environment paths for commands and accessing different arguments.

To start using our project, you must first compile and run the program using the commands “g++ -o <executable name> program.cpp”, if running on Lamar’s linux server add “-std=c++11” to the end, then type “./<executable name>” to run. After running the project the user's name along with “K-\$HELL” and an introduction message will be printed to the screen and will begin waiting for user input.

The main method in our code launches the shell function which uses the helper methods written above to create the shell and display its data. The shell function holds an infinite while loop that starts by printing an introduction message to the user before waiting for the user’s command input. By reading the input, the command and the arguments are gathered, the method then proceeds to check if the command is the exit command, on which will terminate the shell program. Assuming the command is not the exit command, we use the command class to look up the environment path and proceed with a validity check to ensure the command is valid, printing errors should it be invalid or not exist. After the validity check the main process creates a child process which executes the command function before exiting, which the parent process waits for before starting the loop of waiting for user input again. The command execution uses `execv` command which takes the full path and a c string of the arguments from input as parameters to execute the command.



**Figure 1: State Diagram**

In conclusion, this Simple Shell Project was a nice little project to show the basic inner workings of how the structure of the coding language is set up. The shell can modify the environment in which commands run. Input and output can be redirected to files, and processes that communicate through pipes can be invoked. Commands are found by searching directories in the file system in a sequence that can be defined by the user. Commands can be read either from the terminal or from a file, which allows command procedures to be stored for later use. Whether the program runs into an error, or it is a valid command, the Simple Shell Program is able to handle all the different outcomes from the user interaction via commands.