COSC 4302 Written Report (Group Project: Simple Shell)

**Group Members:**

Ann Chapple

Nelson Paz

Noah Roden

**Introduction**

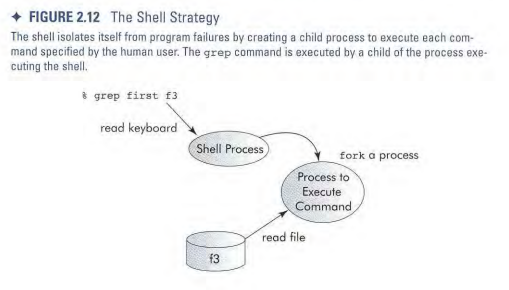
In this project, our group effectively designed and implemented a simple interactive shell program. This shell functions very much like a simple UNIX shell command line interpreter. The program allows the user to input simple Linux commands which are then parsed and executed by a child process.

**Technical content**

This shell is a simple shell designed in C. The code first prompts the user for a command. The prompt consists simply of "$". The program prompts the user for a basic Linux command, which the user will then input through the terminal. That command is then parsed, and executed with a child process using the execv() function. The implementation of execv() into the program requires that the program read in the PATH environment variable (termed simply in our code, pathEnvVar). It then searches each directory in the PATH for the command file name that was input by the user. The output is then displayed to the terminal. When the user is done, entering “quit” or “exit” will cause the child process to terminate, and the program closes.

However, in order to for the program to run correctly and efficiently, we implemented 5 different functions in addition to the main function: lookupPath, parseCommand, parsePath, printPrompt, and readCommand. lookupPath finds the directory path and returns it as a string. parseCommand validates the command name and constructs a parameter list, separating the whitespace in any of the commands as needed. parsePath reads the path variable and creates an array which holds the directories in that path. printPrompt merely prints the prompt to the user for a Linux command. readCommand does exactly what it sounds like by storing the user’s command input into a buffer using the C standard input library.

An excellent example of the shell strategy which we followed in designing this code is shown in the figure below.



We outlined the steps for our simple shell as follows:

step1: Parse the path

step2: Print the prompt

step3: Read the command

step4: Parse the command

step5: Find the command

step6: Execute the command

**Readme:**

Import code into C compiler and run. The user will be prompted to enter a command with “$.” Basic Linux commands such as ls, pwd, and clear are compatible with this program as it is only a simple shell. Simply type in the command and press “Enter.” When the user is ready to exit, type “exit” or “quit” to terminate the shell.