```
************************************************************
*  Name      :  Sharvita Paithankar
*  Student ID:  108172438
*  Class     :  Operating Systems
*  HW#       :  Lab 2
*  Due Date  :  April 3 2019
************************************************************


                Read Me


************************************************************
*  Description of the program
************************************************************

The command line interpreter, called a shell, is an application
program that gets commands from a keyboard and uses the system call
interfaces to invoke OS functions.
Your shell should isolate itself from program failures by creating a
child process to execute each command specified by the user.
The following were the guidelines given
Printing a prompt – when myshell has begun, it display its own string
as its own prompt.
Getting command line – To get a command line, the shell performs a
blocking read operation so that the process that executes the shell
will be blocked until the user types a command line in response to the
prompt. When the user enters the command, the command line string
should be returned to the myshell.
Parsing the command: Take the user command and divide it into command
and arguments and check for errors.
Finding the file – The shell provides a set of “environment variables”
for each user such as “PATH”. The PATH environment variable is an
ordered list of absolute pathnames that specifies where the shell
should search for command files. Use getenv function to find the PATH
environment variable
Create a child process and launch the command – Once the command and
arguments are prepared, create a child process to execute the command
with its arguments and wait until the command is completed.



************************************************************
*  Source files
************************************************************

Name:  main.cpp
   Main program.  This is the driver program that uses for and exec to
   execute a program to allow the user to input commands into the
shell
```

and do what has been asked by the user. This file has all the code to
create the shell.

```
*********************************************************
*  Circumstances of programs
*********************************************************
```

The program compiles and runs successfully on linux

The program was developed and tested on gnu g++ 4.8.2. It was compiled, run, and tested on gcc

```
*********************************************************
*  How to build and run the program
*********************************************************
```

1. Uncompress the homework. The homework file is compressed.
   To uncompress it use the following commands
   ```
   % unzip Sharvita_Paithankar_Lab2.zip
   ```

   Now you should see a directory named homework with the files:
   ```
   Oslab2.cpp
   ```

2. Build the program.

   Change to the directory that contains the file by:
   ```
   % cd lab2
   ```

   Compile the program by:
   ```
   gcc OSlab2.cpp
   ```

3. Run the program by:
   ```
   % ./a.out
   ```

4. Delete the obj files, executables, and core dump by
   ```
   %./make clean
   ```