**Aishwarya Madbal Gowda**

Project 1: Develop your own shell

**ReadMe**

* **In this project, we developed our own shell with following requirements:**

1. Our shell will be invoked from the Ash shell provided with MINIX.
2. Our shell will first execute a PROFILE file which defines a PATH variable that will allow you to access programs provided in/bin and /usr/bin. Once the PROFILE file is executed, we will be in a home directory specified by us in the PROFILE file. The path and the HOME variables do not replace those of the Ash shell from which our shell is instantiated.
3. In a command line of our shell we will be able to exercise any executable programs including the utilities provided in /bin and /usr/bin.
4. Our shell will set and alarm which fires 5 seconds after it has launched a command. After the alarm is fired, our shell will ask the user whether he/she wants to terminate the command and will terminate the command in the PROFILE file or in a command.
5. Our shell will remember the commands that the user has entered. In the future, when the user wants to enter a command again, he/she has to press the tab key, our shell will fill in the rest of the command line. If he/she does not like the suggestion, the user will have to press the UP and DOWN arrow key, our shell will stop executing until the next command. If the user suggests a new command, the shell will remember that too. The memory of the past commands survives after the shell exits.
6. Our shell will support a command line with parentheses and the sequence and parallel execution operators (“;” and “&”). Therefore cmd & (cmd1; cm d2) is a valid command line specifying that cmd will be executed in parallel with (cmd1; cmd2).

* **How to use our shell?**
* The minix version used during the project was 3.2.1
* We used VMware workstation 12 Player on Windows OS to install Minix OS.
* Our executables are stored in “MinixProjectCode” folder under directory **/home/Ganesh.**
* So, firstly need to go to the above folder using the command

***cd /home/Ganesh***

* There are totally ------number of files out of which 7 are .c files
* All \*.c files must be compiled before executing our shell using “cc” or “clang”command as shown below.

***[#]clang CommandExecution.c CommandExecutionControler.c CommandMemory.c Parser.c UserInputVer2.c conio.c main.c -o main***

* To run, use the **main** file as shown below.

***./main***

* Executing the above command, now you will be prompted to our shell.
* You can execute all the commands from **/bin** and **/usr/bin** like

1. cat
2. echo
3. ls
4. pwd
5. shutdown

* Our shell has set an alarm which fires 5 seconds after it has launched a command.It will ask the user whether he/she wants to terminate the command and will terminate if the user approves.

**Eg:- [#] *sleep 10***

*The above command will sleep foer 10 seconds as a result of which our shell throws to the user “Do you want to terminate?[Y/N]”.*

*If user enters ” Y “ then, the shell will exit.*

*The user is also provided with a feature to turn on and off the alarm*

*in the PROFILE file using the variable named “SetAlarm” which is set to 5 seconds by default.*

* While entering commands, which have already been executed, you have to press tab and the similar commands will be displayed on screen and if you are not satisfied with the commands, you can press up and down arrow keys and select the desired command.
* If you enter a new command, it will be stored in memory after the shell exits.

Eg:- [#] ls

[#] ls -l

[#] ls -la

[#] ls -lr

When the user has executed the above mentioned commands, and then again try to enter these commands, say [#] ls ,and then press tab, the user will be provided with the options “-l ,-la, -lr ” which he choose by using UP and DOWN arrow keys.

* Our shell supports a command line with sequence and parallel execution operators [;] and [&] respectively.

Eg:- [#] ls;pwd&whoami

The above example will illustrate both sequence and parallel execution operators

* If you wish to boot up our MINIX , you have to use the command “*shutdown*”.