A **PROJECT REPORT** Submitted

in Partial Fulfilment of the Requirements

for the Degree of

**Bachelor of Technology**

In

**Computer Science Department**



Submitted to

Dr. PRADEEP ARYA

SCHOOL OF ENGINEERING AND TECHNOLOGY

BML MUNJAL UNIVERSITY GURGAON

March 06, 2021

**OS Mini Project by Group 6**

**Title of the Project**

**SHELL BASED WORKING ENVIRONMENT**

(Mini OS)

MENTORED BY

**Dr.** **PRADEEP ARYA**

SUBMITTED BY

|  |  |  |
| --- | --- | --- |
| **S No.** | **Enroll No.** | **Name** |
| 1 | 190C2020028 | PAVAN KUMAR BELLAMKONDA |
| 2 | 190C2020029 | PRABHATH AKULA |
| 3 | 190C2030088 | MEKALA SHARATH CHANDRA |
| 4 | 190C2030096 | G VENKATESH |
| 5 | 190C2030086 | GOTTAM ROHITH REDDY |

**TOPIC**

1. Summary of the project
2. Problems that have motivated to do this project
3. Problem that is solving by the project
4. Detailed explanation of the project
5. Technical features of the project
6. Block diagram
7. Hardware and software
8. New in our project
9. Alternative methods
10. Status of the project
11. When and how we first conceived to this idea
12. References

**SUMMARY OF THE PROJECT:**

User Interface is the term used for specifying how a user interacts with a device, particularly computer. CLI and GUI are the various types of user interfaces. Mainly the interfaces differ in the graphics that are employed within the OS. To do an operation on the CLI system one has got to write a command. On the opposite side, GUI users provided the visual aids/graphics with the help of icons and images, which makes the user perform his task directly.

CLI systems require users to have good knowledge within the commands for performing tasks whereas GUI does not require that much knowledge about the interface, it might be operated by novice users too.

This project makes our command-based user interface easy to use for any person who knows basic English. The script we write for this project takes the input the normal English words like commands like

1. Calc
2. Clock
3. Create
4. Display
5. Dup
6. Info
7. Media
8. Restart
9. Search
10. Shutdown

and convert them into the respective command by itself which is accepted by the computer and make the machine to do action/task according to the user’s requirement. As it is based on the command it is faster than that of GUI at same time it is making the user not required to remember the command. The outcome of the project in a single line.

**“ATTAINING SPEED WITH LESS MEMORIZATION”**

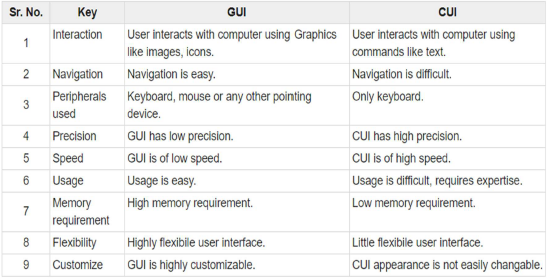
**PROBLEMS THAT ARE MOTIVATED US TO DO THE PROJECT:**

Main Goal of our Project is to attain the speed of the operating system since Graphical User Interfaces are comparatively slower than the Command Based Interface. But the remembering of all the commands is quite hard for a non-technical person. So, we found a method which is used by many companies used to solve the same problem decades ago for communicating with the microprocessor, Generally, to communicate with the Microprocessor we must write the command in hexadecimal format they solved the issue by introducing the assembly code(mnemonics).

By inspiring from the above method, we created a working environment in which it accepts commands which are like the terms used by us in general communication which helps the person for no need of memorization of exact command. Even remembering all commands makes the person to restrict only to a single Operating System as commands for each O.S is different.

**PROBLEM THAT IS SOLVING BY THE PROJECT:**

We solved the problem that motivated us for doing the above project i.e., the memorization of different commands. Even We created a new type of commands which are not present as a command in any previous working environment. Like find feature which is generally available only in GUI operating Systems like Mac OS and windows but not present in the command based operating systems like MS Dos etc.

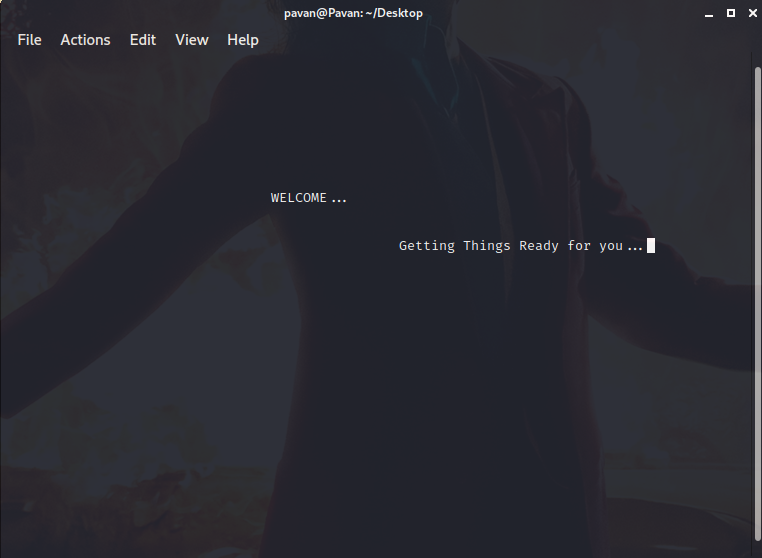


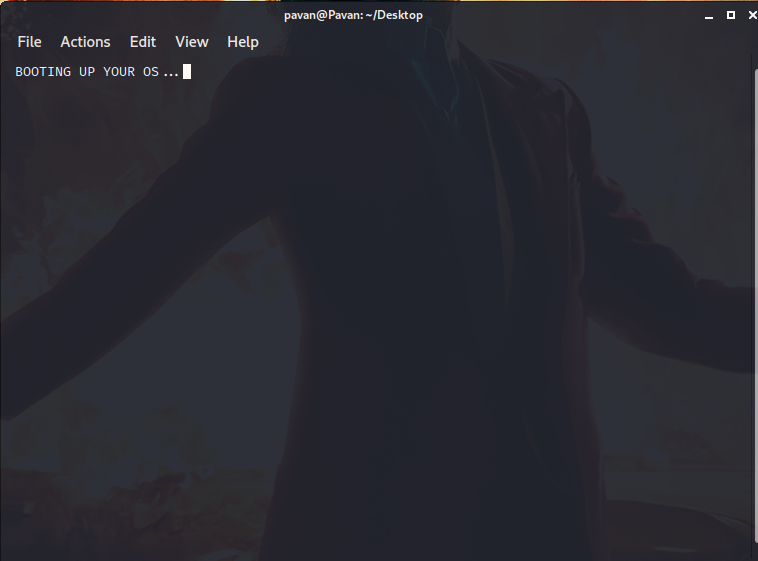
We are solving the issue of flexibility, customize and usage of CLI slightly with the help of our project which helps the user in gaining the advantages of the CLI i.e., precision and the speed.

**DETAILED EXPLANATION OF HOW THE PROBLEM IS SOLVED:**

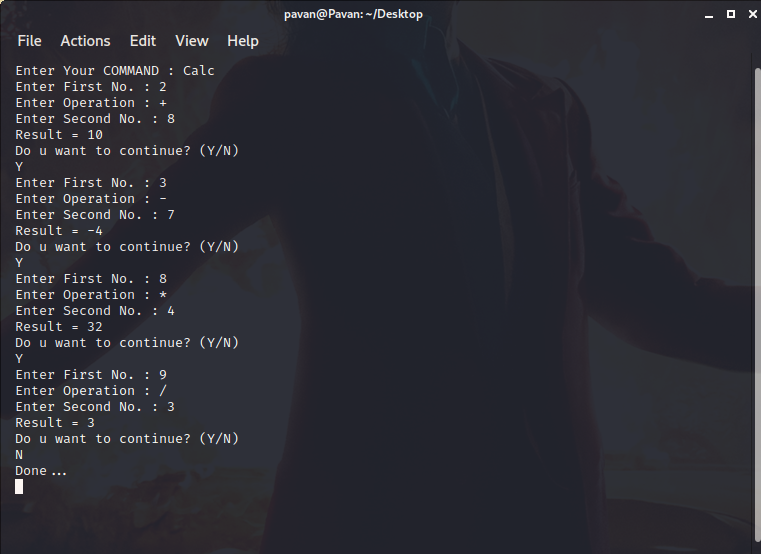
There are many commands in command prompt that are not easy to understand, so this project solves the problem of remembering those commands. This project was created to ease the life of the user by creating some commands that use the easily understandable words that we use in our daily life, we assign the names by the functions they do. Explanations about each command we use in our script.

**Boot:** We created tis command to display that the Mini-OS is booting up

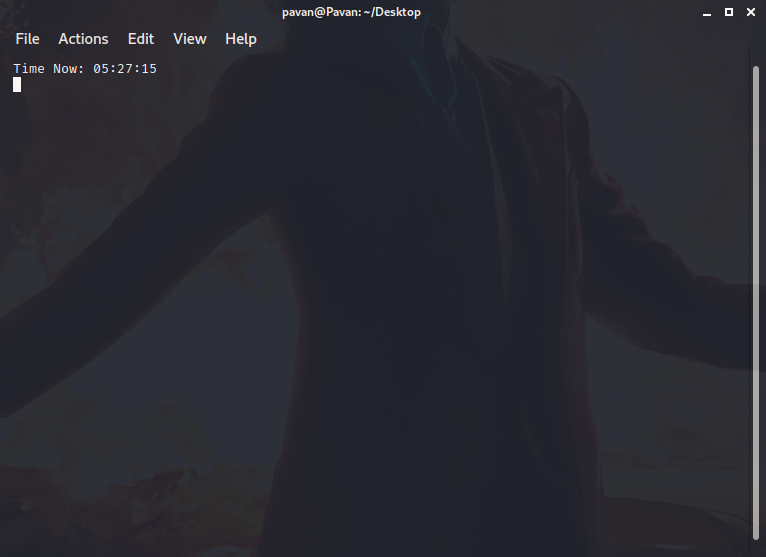




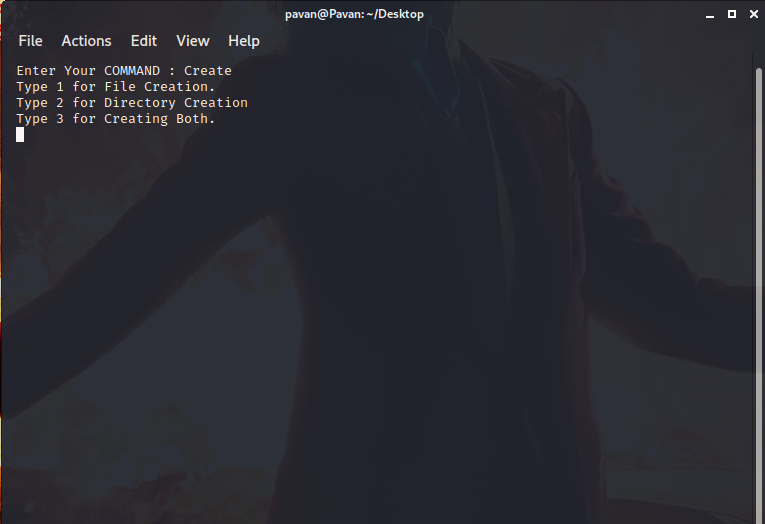
**Calc:** We created this command to perform different Arithmetic operations.



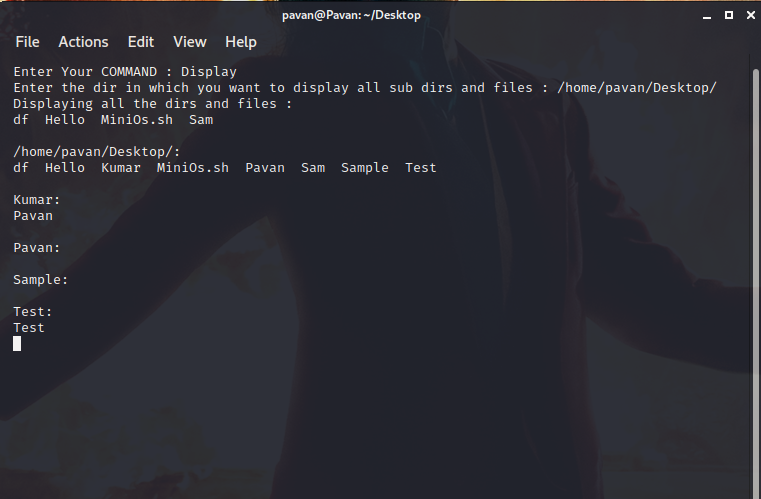
**Clock:** We created this command to display the present time now.



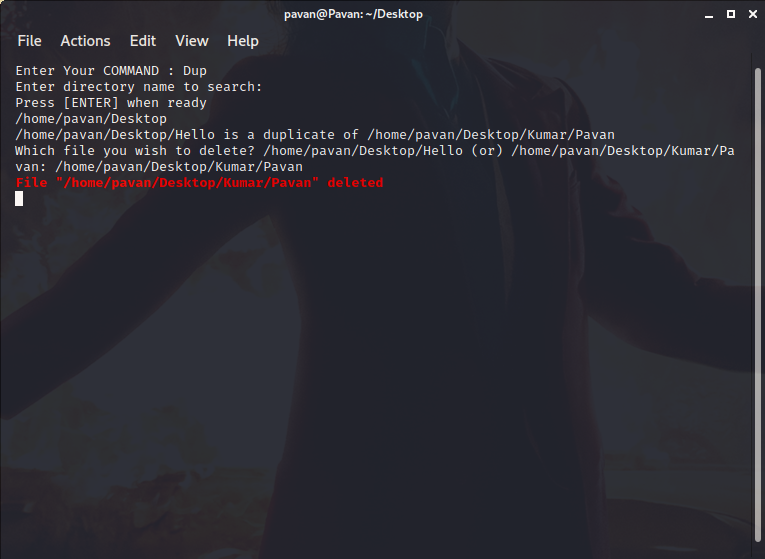
**Create:** We created this command to create a folder or a file or both in a single command, this solves the problem of entering the command repeatedly for the single action and even with the help of us create command it is not required to type two different commands i.e., one for directories and other for creating the files.



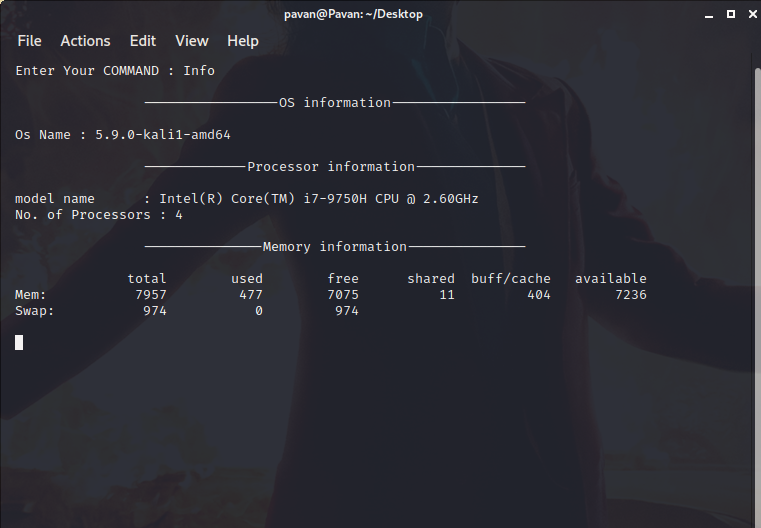
**Display:** We created this command to display all the sub directories and files in the directory entered.



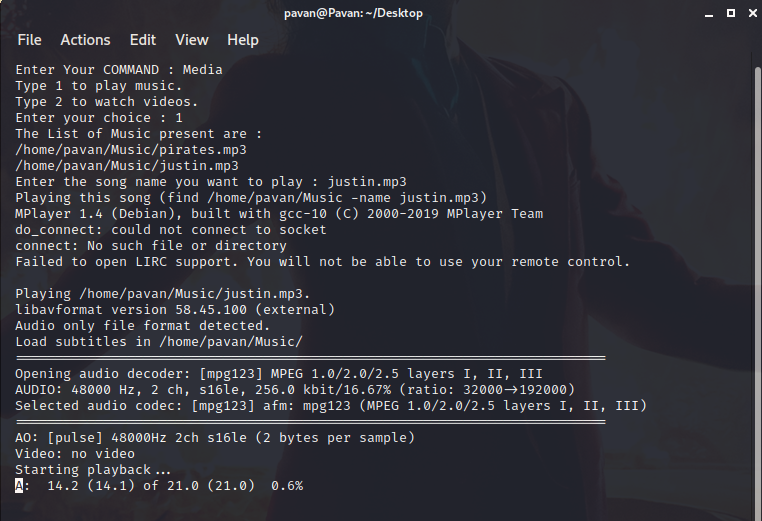
**Dup:** We created this command to print all the duplicates in the given input directory and delete any one of them. This solves the problem of just showing all the duplicates in the present working directory and not deleting them in windows. In windows we must remove those duplicate files manually by going back to the directory present with the duplicates. The output of the following command is as follows.



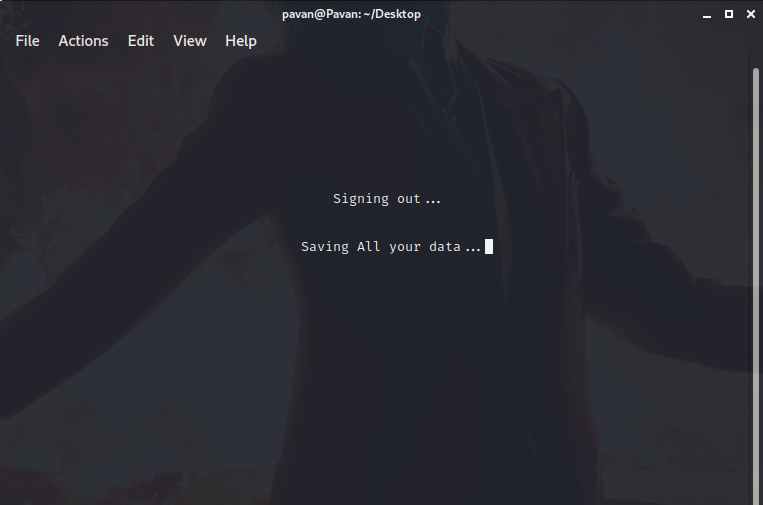
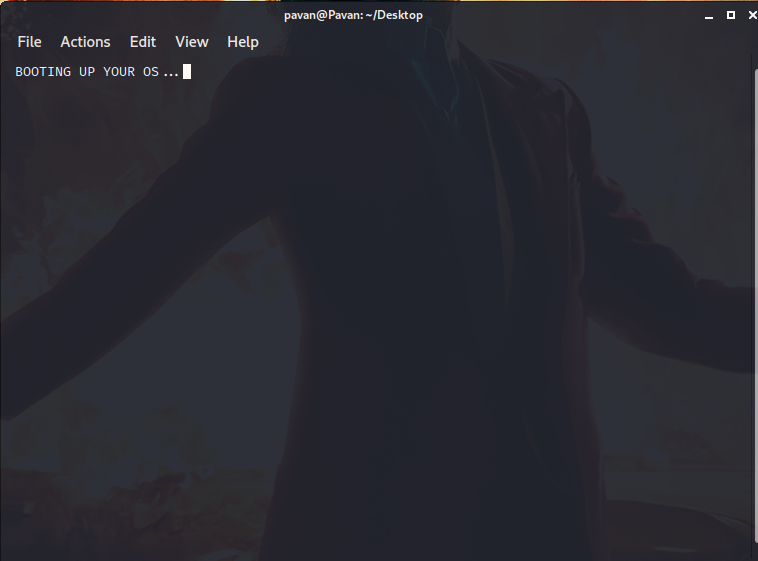
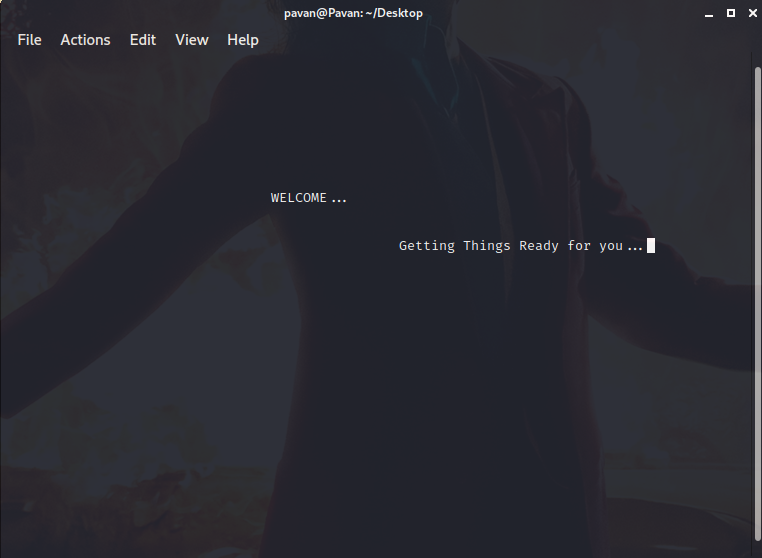
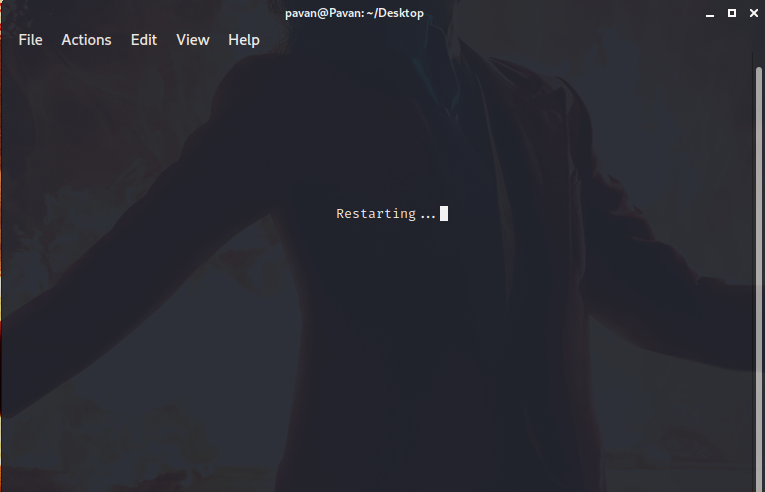
**Info:** We created this command with the help of cat commands (which are very difficult to remember) present in the Linux systems. Our About command helps the user to know about the hardware & software, Kernel, Bios versions, cache capacity, processor cores and threads, main memory and many other things related to them.



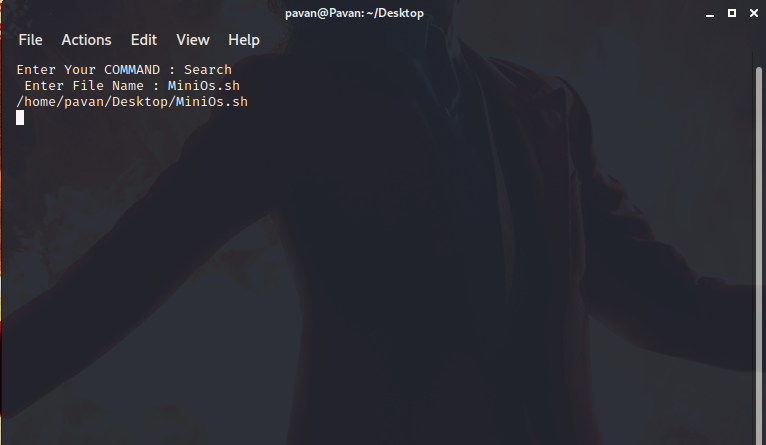
**Media:** We created this command in order the play and media file in the command prompt itself and it is very easy to use this command solves the problem of traversing to the media files location and then play it, the command we created displays all the media files that are present in the whole Operating system and plays the media file which is given as input. The output of the command is as follows.



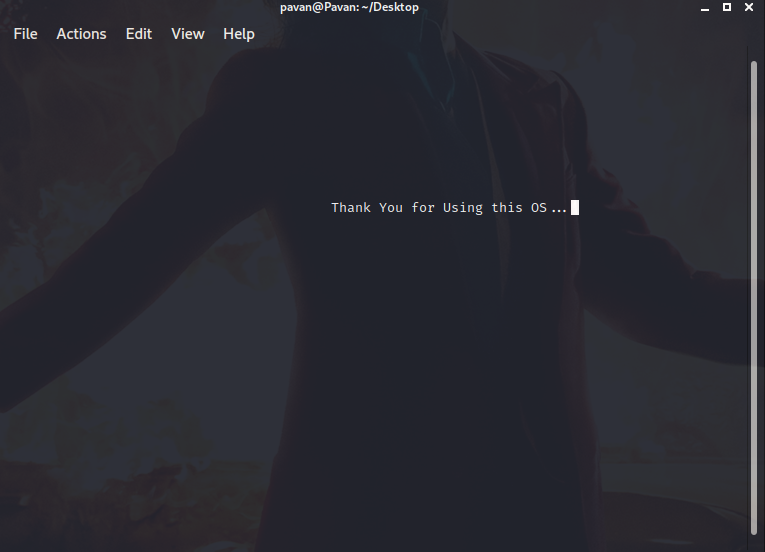
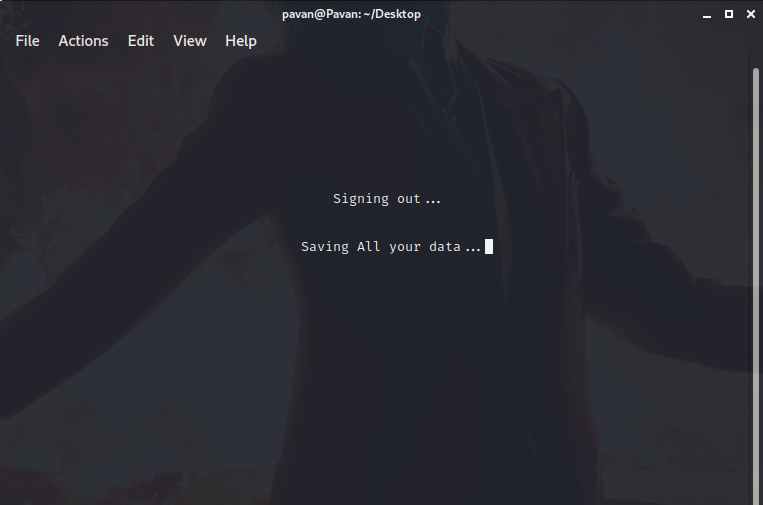
**Restart:** We created this command to display that the mini-OS is getting restarted.

**Search:** We created this command to search for a file from the home directory itself because we wanted to solve the problem of traversing to the location of the file and search for it (whether it is present or not). This command displays the whole location of the file from the Home directory itself.



**Shutdown:** We created this command to display that the mini-OS is getting shutdown.

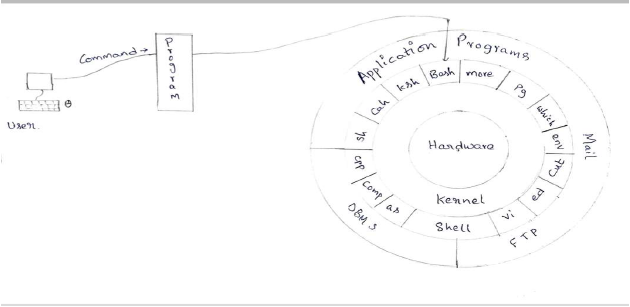


**TECHNICAL FEATURES OF THE PROJECT:**

To create this project, we used bash shell script, GIT bash, Kali Linux. We use shell script to create a bridge between our environment and operating system. Bash shell script can be directly implemented on the Linux machine to perform actions. As bash script cannot be used directly on windows machines, so we used GIT Bash which helps the windows machine to run any shell scripting in its domain. For complete testing and implementation of our script we use Kali Linux which can completely match with the shell scripting nature i.e., having home as main drive and not having other sub drives (Local Disks) which makes the script easier to find the required contents.

**Shell Script:** A shell script is a list of commands in a computer program that is run by the Unix shell which is a command line interpreter. A shell script usually has comments that describe the steps. The different operations performed by shell scripts are program execution, file manipulation and text printing. A wrapper is also a kind of shell script that creates the program environment, runs the program etc.

**BLOCK DIAGRAM OF THE PROJECT:**



**HARDWARE AND SOFTWARES THAT ARE USED:**

1. **Git Bash:** Git Bash is command line programs which allow you to interface with the underlying git program. Bash is a Linux-based command line while Shell is a native Windows command line.
2. **Kali Linux**: Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security.
3. **Leaf Pad:** Leaf pad is an open-source text editor for Linux, BSD, and Maemo. Created with the focus of being a lightweight text editor with minimal dependencies, it is designed to be simple and easy-to-compile. Leaf pad is the default text editor for the LXDE Desktop environment, including Lubuntu up to version 18.04 LTS. After Lubuntu moved to the LXQt desktop Leaf pad was replaced by Feather Pad.
4. **Mouse Pad:** Mouse Pad is a notepad-like graphical text editor for Xfce based on the Leaf pad text editor that runs on any Linux desktop. It is very handy as a go-to tool when you do not have any heavy lifting to do.
5. **Vim:** is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi-editor distributed with most UNIX systems. Vim is often called a "programmer's editor," and so useful for programming that many consider it an entire IDE.

**NEW IN OUR PROJECT & IT AND ITS NEAR TECHNOLOGY:**

In our project we used different types of functions that are modified of the present functions. Such as

1. We created a “Media” command which plays any media in a command prompt from the home directory itself no need to go to the file’s directory but the existing function in windows command prompt is that we need to go to the files directory and then play the file. So, the function “Media” that we created is very useful and easy to use.
2. We created a “Create” function which is useful to create a file or directory or both of them in one command but when we use the existing function “mkdir” it creates a directory but if we need to create a file, we need to type another command.
3. We also created “Dup” function which asks for the directory and prints all the duplicate files present in the given directory and then it asks you whether to delete the duplicate files or not, But the existing function “fdups” will just show all the duplicates that are present in the present working directory and we cannot be able to delete the files just using these functions we have to do it by handy.
4. We created a function “Search” which is very easy to use and it is different from the existing function “find” Because in “find” we need to specify the path and the filename to search the file but, in our function, we just need to mention the file name which displays the directory of the file.

**ALTERNATIVE SOLUTIONS:**

As discussed earlier, the main motto of our project is “Attaining Speed with Less Memorization”. And according to fact, command-based interfaces are comparatively faster than graphical user interfaces, so it is not possible in GUI.

When we think of other alternatives, the only other alternative is only by making some changes in the kernel. Kernel is nothing but the main core of the operating system which controls whole tasks of the system. It is very different from the shell as the shell is the interface that allows users to communicate. But it will be very difficult for them to do the changes directly as it involves many interlinked operations which are designed in a complex manner and assigned to various functions.

So, there is very less chance for others to solve it in a better alternative way than our way of solving represented problem.

**STATUS OF THE PROJECT:**

This project has been created, tested, implemented. There are currently no other working environments that implements the idea we implemented in our project. We found many shell scripts on many sources which are restricted to a particular use only, but we created many new techniques and made them to work under a single roof.

We first came across this idea of using shell scripting in our OS lab session. We thought that it would be great helpful to the students and many users as they could experience the CLI interface without remembering many commands.

**WHEN AND HOW WE FIRST CONCEIVED TO THIS IDEA:**

When we are working with Linux, MySQL, and UNIX there are several commands that we must enter to do our work and the commands are very difficult to remember and they do not make any sense with the work they do. So, we came up with an idea that we should create a project that makes the life of the user easy to work and that the commands should make sense of the work they do.

**REFERENCES:**

1. Abraham Silberschatz, A. and Galvin, P.B., (1994), “Operating System Concepts “, 4th ed Addison-Wesley, MA.
2. http://www.cs.ecu.edu/sartipi/courses/OS/f12/3.LabProjects/2.Stallings-TextBook- Projects/html/shells.html
3. https://www.macs.hw.ac.uk/~hwloidl/Courses/LinuxIntro/x945.html
4. https://opensource.com/article/19/12/automation-bash-scripts
5. https://www.coursera.org/learn/cybersecurity-roles-processes-operatingsystemsecurity/home/week/3