1. INTRODUCTION

The project is built over GUI based script that behaves like an Operating System. The script has been written on ubuntu which is a free and open-source Linux distribution. The GUI displays objects that convey information and represent actions that can be taken by the user. The objects contain main menu with further functionalities when the user interacts with them which are detailed below

A GUI is considered to be more user-friendly than a text-based command-line interface, such as shell of Linux-like operating systems.

The Operating System have the following properties.

- It allows you to create folders and files.
- It allows you to change rights of your files.
- It can help you in searching files.
- It allows you to create processes and threads that perform specific tasks. **For example**
  - It allows you to create a process that sorts array in ascending order.
  - It allows you to create multiple threads that may help in solving matrix operations etc.
- It allows you to display processes like a task manager in Windows and should allow to kill any selected process.
- It allows to open applications like Firefox, Image viewer etc.
- It allows to share data between processes such that output of one process becomes input of another. Also provides sub menu to select Process1 and Process2 that gives input or input or vice versa.
- It allows to write Linux C programs to create your own process that can perform any desirable task. It also provides options in user sub-menu to execute and delete that program.

2. SPECIAL FEATURES

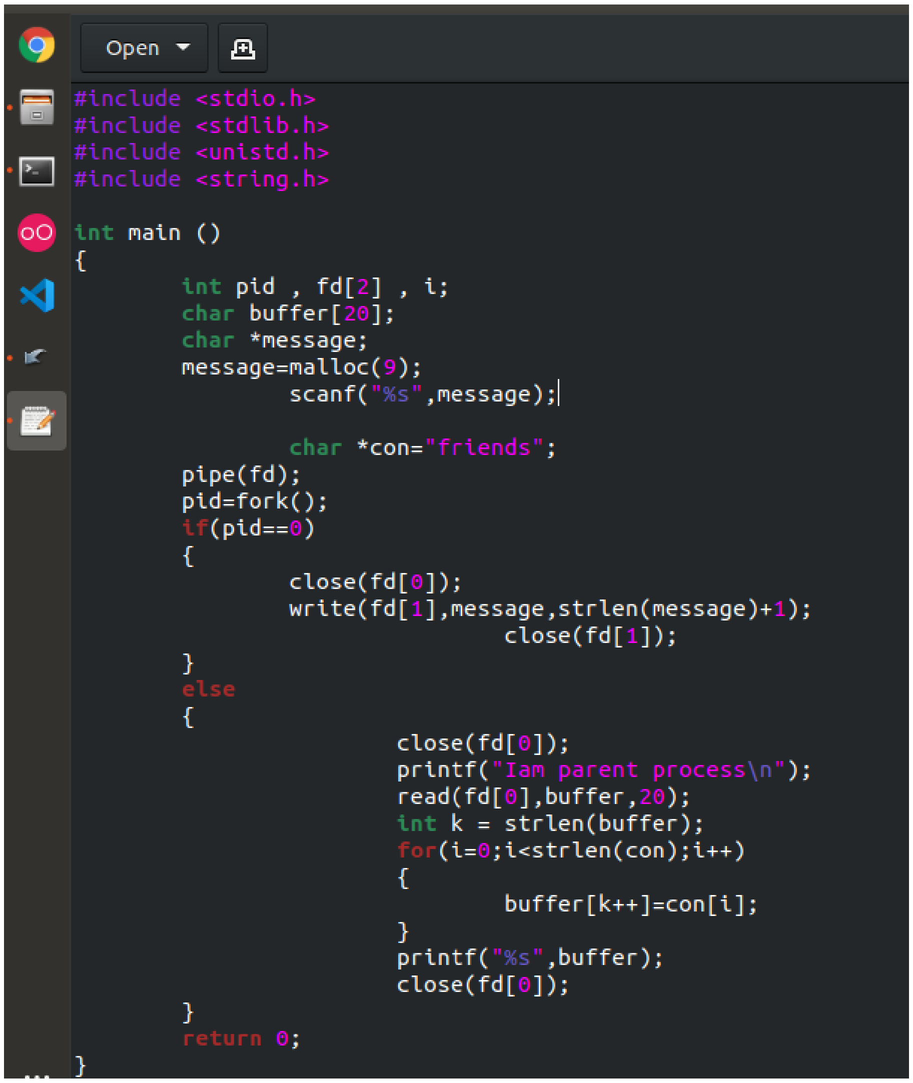
The GUI is built on zenity which allows you to incorporate a wide range of graphical interface elements in your Bash scripts. It’s a powerful toolkit that gives your scripts a modern feel and a contemporary, familiar appearance.

3. RESULTS

The results include the screenshots of code written in C or bash script, GUI, output/action performed written in C or bash script.

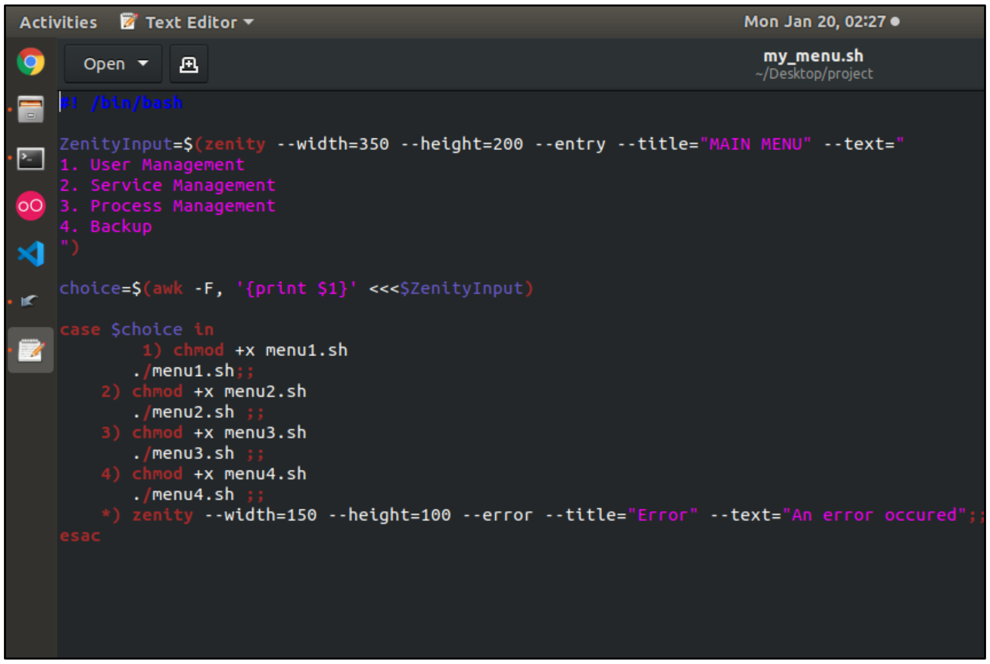
3.1 CODE

Some sample of code that is used to create this GUI based operating system.



```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>

int main ()
{
        int pid , fd[2] , i;
        char buffer[20];
        char *message;
        message=malloc(9);
                scanf("%s",message);

                char *con="friends";
        pipe(fd);
        pid=fork();
        if(pid==0)
        {
                close(fd[0]);
                write(fd[1],message,strlen(message)+1);
                                close(fd[1]);
        }
        else
        {
                        close(fd[0]);
                        printf("Iam parent process\n");
                        read(fd[0],buffer,20);
                        int k = strlen(buffer);
                        for(i=0;i<strlen(con);i++)
                        {
                                buffer[k++]=con[i];
                        }
                        printf("%s",buffer);
                        close(fd[0]);
        }
        return 0;
}
```

This code is written in C which is generating process synchronization, the main purpose of synchronization is the sharing of resources without interference using mutual exclusion.

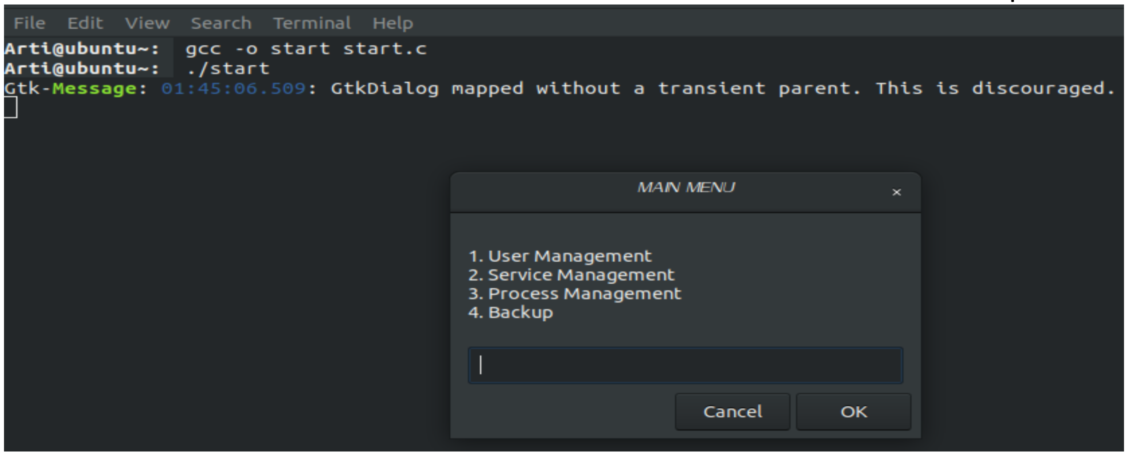


```
#! /bin/bash

ZenityInput=$(zenity --width=350 --height=200 --entry --title="MAIN MENU" --text="
1. User Management
2. Service Management
3. Process Management
4. Backup
")

choice=$(awk -F, '{print $1}' <<<$ZenityInput)

case $choice in
        1) chmod +x menu1.sh
       ./menu1.sh;;
    2) chmod +x menu2.sh
       ./menu2.sh ;;
    3) chmod +x menu3.sh
       ./menu3.sh ;;
    4) chmod +x menu4.sh
       ./menu4.sh ;;
    *) zenity --width=150 --height=100 --error --title="Error" --text="An error occured";;
esac
```

This is the bash script creating menu i.e. GUI using zenity.

3.2 GUI

The GUI has main menu which includes 4 main options.

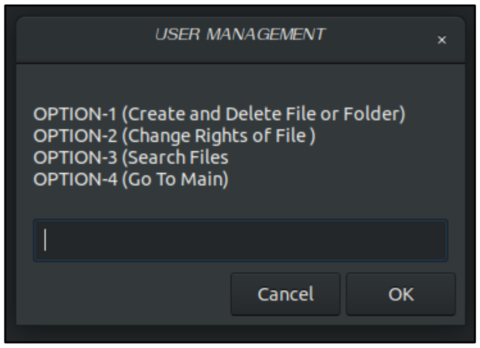


```
Arti@ubuntu~:  gcc -o start start.c
Arti@ubuntu~:  ./start
Gtk-Message: 01:45:06.509: GtkDialog mapped without a transient parent. This is discouraged.
```

```
1. User Management
2. Service Management
3. Process Management
4. Backup
```

1. User Management

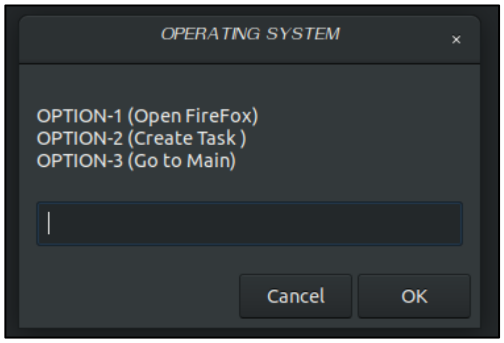
User Management further has 3 main options which can create and delete file or folders, change rights of the file or folder and can search files.



OPTION-1 (Create and Delete File or Folder)
OPTION-2 (Change Rights of File)
OPTION-3 (Search Files
OPTION-4 (Go To Main)

2. Service Management

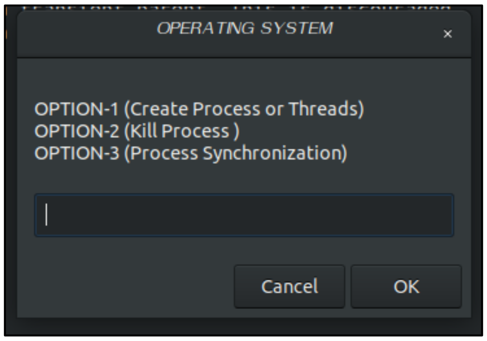
Service management has options which can open application like firefox.



OPTION-1 (Open FireFox)
OPTION-2 (Create Task)
OPTION-3 (Go to Main)

3. Process management

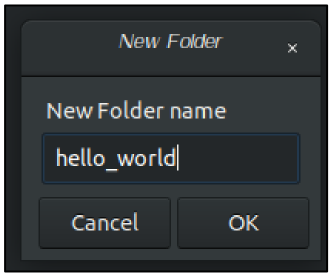
It has options which can create processes and threads and can also kill them and can perform process synchronization. The need for synchronization originates when processes need to execute concurrently. The main purpose of synchronization is the sharing of resources without interference using mutual exclusion.



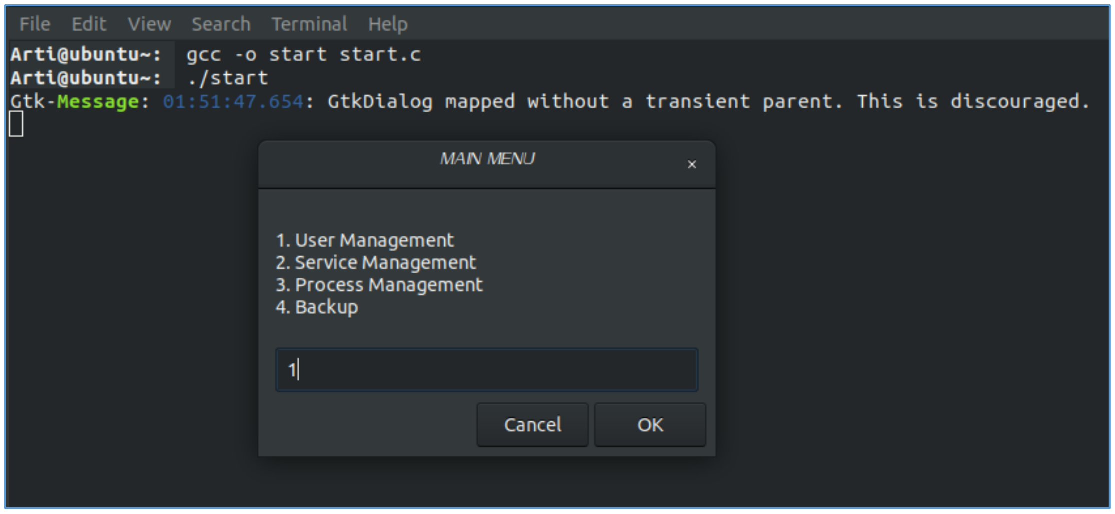
OPTION-1 (Create Process or Threads)
OPTION-2 (Kill Process)
OPTION-3 (Process Synchronization)

4. Backup

This can create backup of files and folders.



3.3 ACTIONS PERFORMED



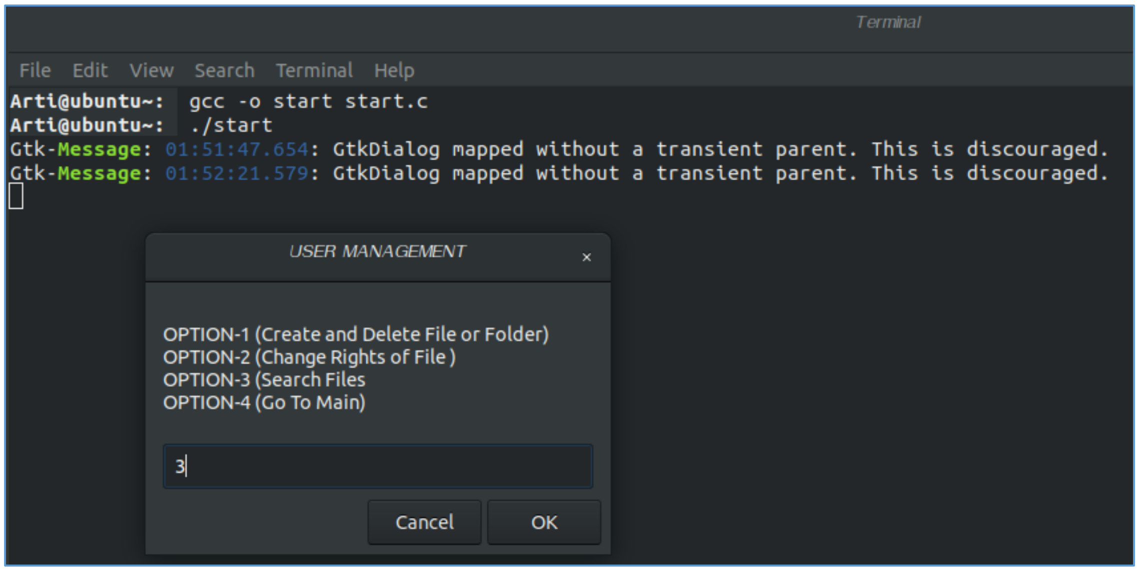
```
Arti@ubuntu~:  gcc -o start start.c
Arti@ubuntu~:  ./start
Gtk-Message: 01:51:47.654: GtkDialog mapped without a transient parent. This is discouraged.
```

MAIN MENU

1. User Management
2. Service Management
3. Process Management
4. Backup

1

Selecting option 1 i.e. user management.



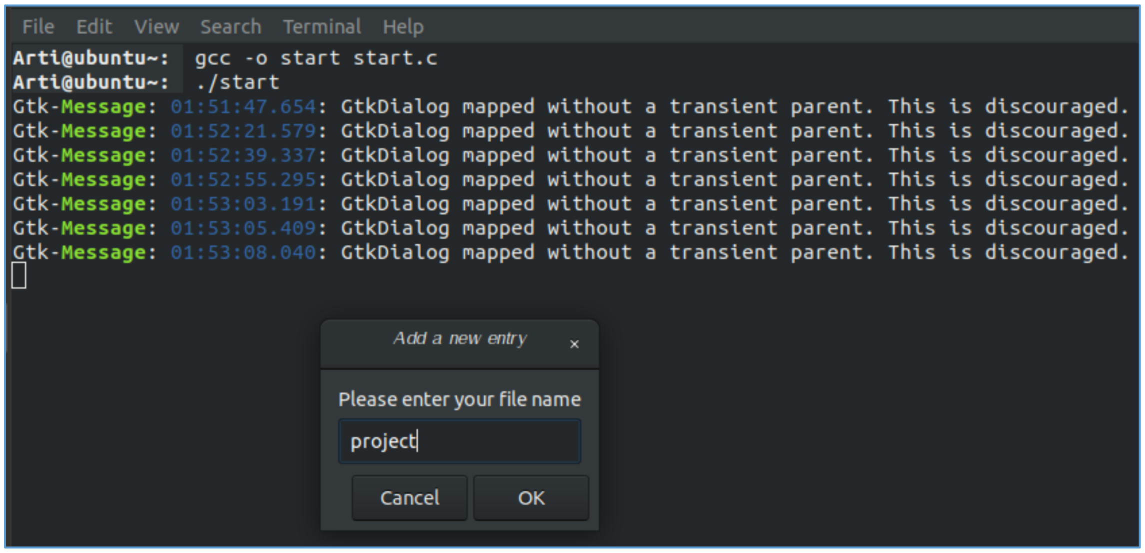
```
Arti@ubuntu~:  gcc -o start start.c
Arti@ubuntu~:  ./start
Gtk-Message: 01:51:47.654: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:21.579: GtkDialog mapped without a transient parent. This is discouraged.
```

USER MANAGEMENT

OPTION-1 (Create and Delete File or Folder)
OPTION-2 (Change Rights of File)
OPTION-3 (Search Files
OPTION-4 (Go To Main)

3

Selecting option 3 for searching any file.



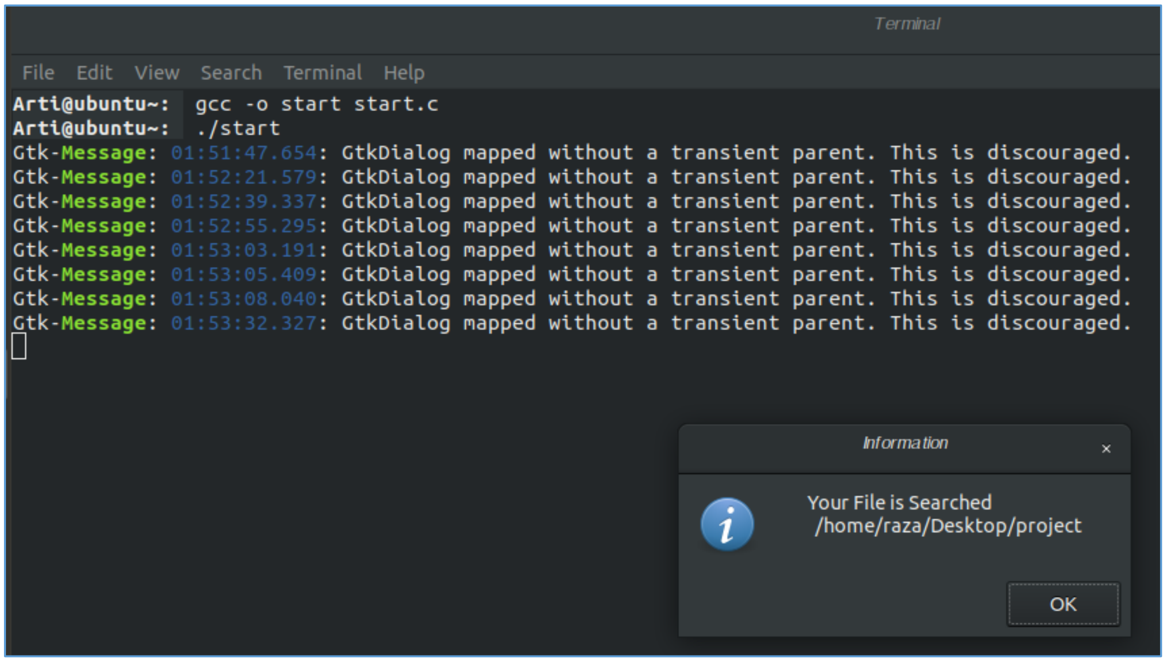
```
Arti@ubuntu~:  gcc -o start start.c
Arti@ubuntu~:  ./start
Gtk-Message: 01:51:47.654: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:21.579: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:39.337: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:55.295: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:03.191: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:05.409: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:08.040: GtkDialog mapped without a transient parent. This is discouraged.
```

Add a new entry

Please enter your file name

project

Entering the file name that needs to be searched i.e. project.



```
Arti@ubuntu~:  gcc -o start start.c
Arti@ubuntu~:  ./start
Gtk-Message: 01:51:47.654: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:21.579: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:39.337: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:52:55.295: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:03.191: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:05.409: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:08.040: GtkDialog mapped without a transient parent. This is discouraged.
Gtk-Message: 01:53:32.327: GtkDialog mapped without a transient parent. This is discouraged.
```

Information

Your File is Searched
/home/raza/Desktop/project

The file has been searched and it is telling the path of the file where it is saved.