**OPERATING SYSTEMS LAB**

**Report**



**Bahria University (Karachi Campus)**

Department of Computer Science

**Name: ALI GAUHAR**

**Enrollment #:** 02-134202-006

**COURSE INSTRUCTOR: MR. Shahid Khan**

**LAB INSTRUCTOR: MISS Fatima Zafar**

### **Contents**

[ACKNOWLEDGEMENT](#_bookmark0)

[ABSTRACT](#_bookmark0)

[INTRODUCTION](#_bookmark0)

[EXECUTIVE SUMMARY/ABSTRACT](#_bookmark1)

[OVERVIEW OF THE WORK](#_bookmark1)

[SYSTEM DESIGN](#_bookmark1)

[CODE/IMPLEMENTATION](#_bookmark2)

[CONCLUSION](#_bookmark3)

**Introduction:**

An operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. In addition, users can interact directly with the operating system through a user interface such as a command line. We have created a simple command line mini operating system. A command-line interface is a means of interacting with the computer where the user issues commands to the program in for of successive lines of text called command lines. A program which handles the interface is called common language interpreter or shell. Today there exist very few people who use command line in their daily use.

# ACKNOWLEDGEMENT

“Firstly, we would like to give our great and lovely thanks to Almighty Allah (S. W) for blessing us with strong and comfortable health as well as courage for the accomplishment of my project report. We have taken efforts in this project. However, it would not have been possible without the kind support and help of our teachers especially “SIR Shahid” and “MISS Fatima Zafar”, who taught us to get the knowledge and skills. I would like to extend my sincere thanks to highly indebted to OUR TEAM for their support in completing our project.

**ABSTRACT:**

Here we have attempted to make mini os of our own.we have experimented on many different modules of an operating system. We use zanity which is a free software and a cross platform program that allow the execution of GTK dialog boxes in command line and shell scripts.GTK is a free and open source cross platform widget tool kit for creating graphical user interface.We made this mini operating system with basic modules like system, process, user, network, storage, file etc. This operating system is based on Linux command. As we know that this is a system OS which maintains and manages the system functionality by guiding the hardware. It is also called as set of programs that control the hardware. Kernel provides path to the peripheral device it is also responds to the calls and invoke the devices according to the call.we are deeply indebted to thanks our respected teacher who help us in our project and guide us.Our operating system is a GUI system. However, it would be very easy to use and highly user-friendly unlike other command line systems.Our project is for learning purposes and can be modified and implemented as a fullfledged GUI system operating system.

**EXECUTIVE SUMMARY:**

Operating system means it is a system OS which maintains and manages the system functionality by guiding the hardware. It is also called as set of programs that control the hardware. Some examples of operating systems are UNIX, MS-DOS, MS-Windows, Windows/NT, Chicago, OS/2, MacOS, VMS, MVS, and VM. There are some services like Kernel services, library services, and application-level services these are all part of an operating system. Kernel provides path to the peripheral device it’s also responds to the calls and invoke the devices according to the call. It can terminate a executing process according to the resource requested.

**Advantages Command Line Interface (CLI):**

1. If the user knows the correct commands, then this type of interface can be much faster than any other type of interface. This type of interface needs much less memory (Random Access Memory) to use compared to other types of user interfaces.

2. This type of interface does not use as much CPU processing time as others

3. A low resolution, cheaper monitor can be used with this type of interface.

**Disadvantages Command Line Interface (CLI):**

* For someone who has never used a CLI, it can be very confusing.
* Commands must be typed precisely. If there is a spelling mistake, then the command will not respond or fail.
* If user mistypes an instruction, it is often necessary to start from scratch again.
* There are many commands which need to be learned-in the case of Unix it can be more than hundred.
* User can’t just guess what the instruction might be, and user can’t just ‘have a go’.

**Goals:**

1. User can see storage and cache details.
2. User can add, delete, update, zip, execute, cut, copy, paste files.
3. User can add, delete, update new user.
4. User can add new user.
5. User can see network detail.
6. User can add new user.
7. User can see process details.

**Overview of the Work:**

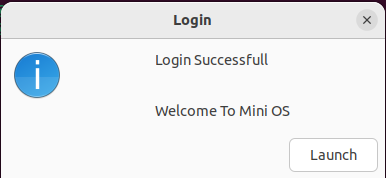
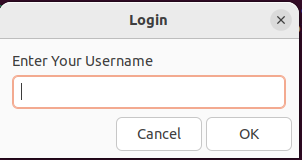
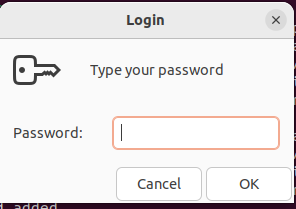
**Problem description:** make GUI interface to communicate user with system easily

**Software used:** vmware station 2017(for coding our OS script)

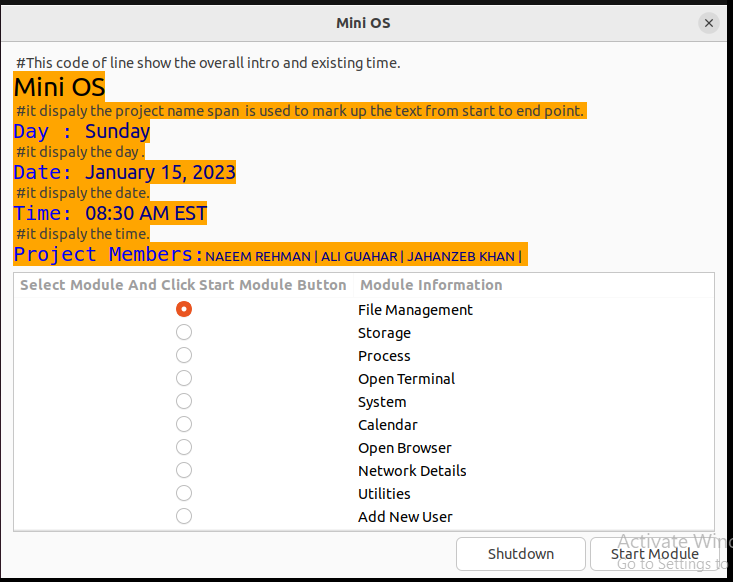
**Hardware used:** Laptop /desktop pc

**System Design:**

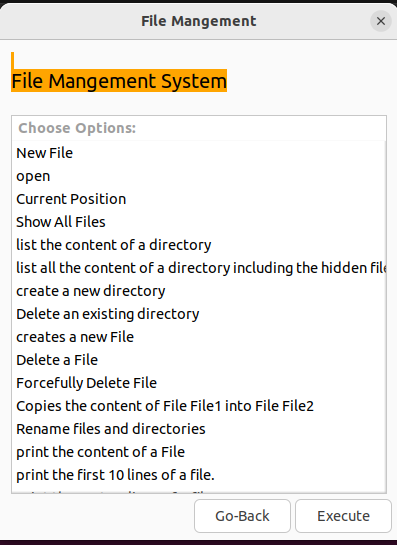
**Login interface:**

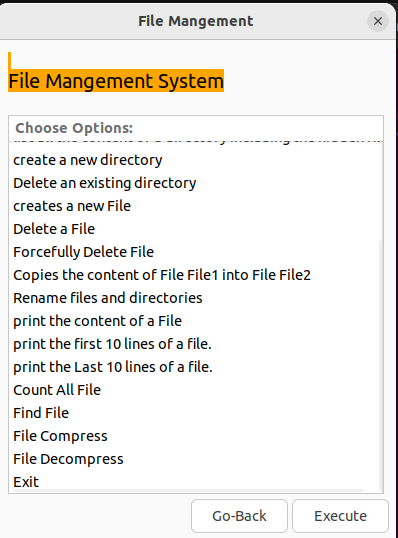


**Main interface view:**

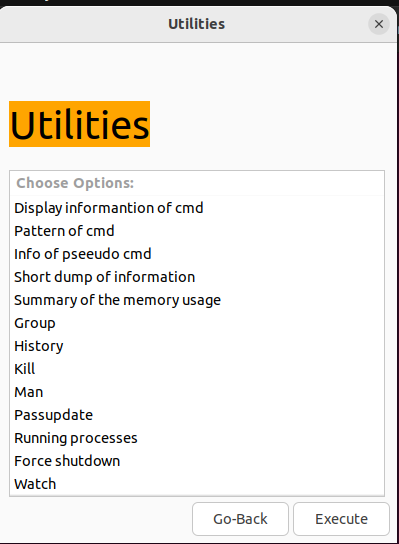


**File management:**

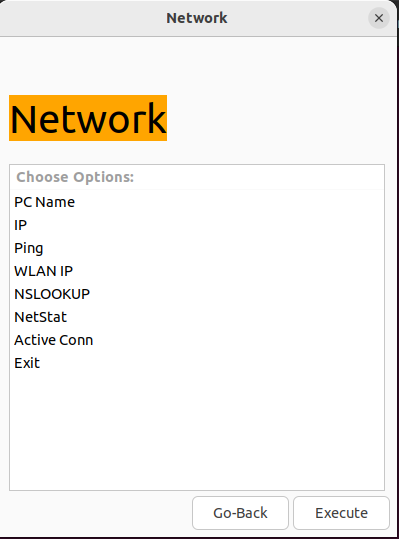
****



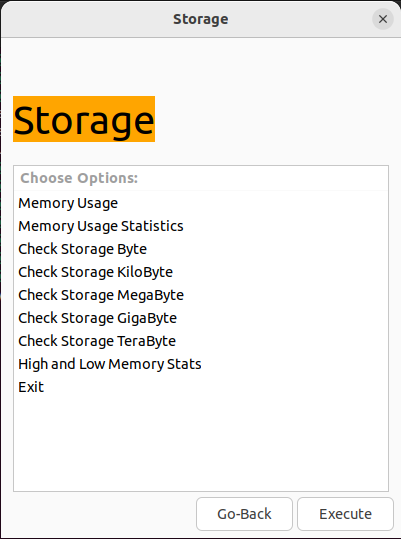
**Utilities:**



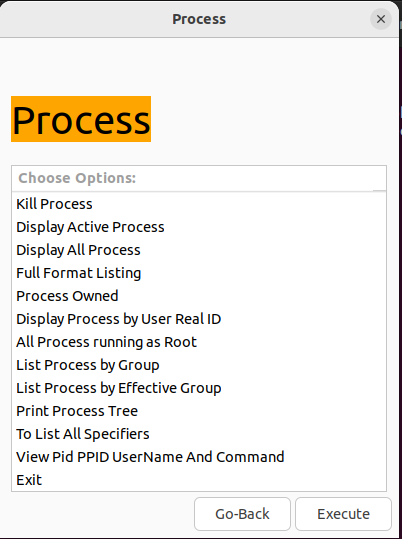
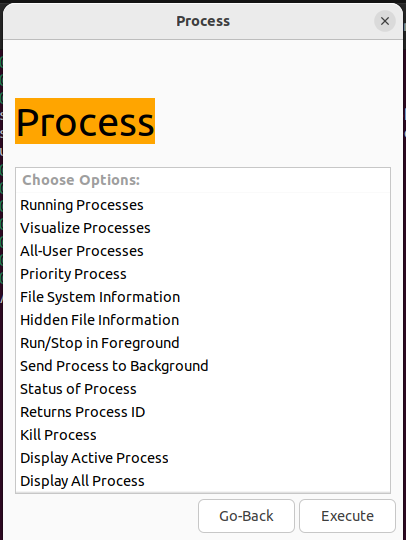
**Network:**



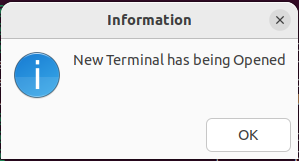
**storage:**



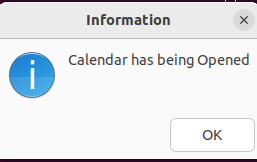
**Process:**



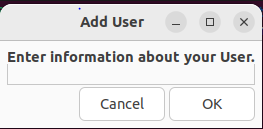
**New terminal:**



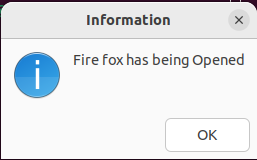
**Calender:**



**Add user:**



**Browser open:**



**CODE/IMPLEMENTATION**

#!/bin/bash

f\_new(){

if zenity --entry \

--title="Add new profile" \

--text="Enter name of new profile:" --width 309 --height 240 \

--entry-text "NewProfile">>USER.txt

then echo $?

else echo "No name entered"

fi

}

f\_f(){

OLDIFS="$IFS"

IFS='-'

FILES=$(zenity --file-selection --multiple --separator='-' --title "Pick a file")

IFS="$OLDIFS"

for file in "${FILES[@]}"

do

if [[ $file == /\* && -f $file ]]

then

echo $file

fi

done

}

f\_open(){ #this function will open the file and and read it sequential through cat command and this function make the read file content show on the window and also then print on the terminal but sometime it not read that file becuase due to lock or permisiion denied chmod

zOpenPath="$(zenity --file-selection)" #this function will open the file and and read it sequential through cat command and this function make the read file content show on the window and also then print on the terminal but sometime it not read that file becuase due to lock or permisiion denied chmod

if [ "$zOpenPath" != "" ] #this function will open the file and and read it sequential through cat command and this function make the read file content show on the window and also then print on the terminal but sometime it not read that file becuase due to lock or permisiion denied chmod

then

zData=$(cat "$zOpenPath") #this function will open the file and and read it sequential through cat command and this function make the read file content show on the window and also then print on the terminal but sometime it not read that file becuase due to lock or permisiion denied chmod

zNewData=$(echo -n "$zData" | zenity --text-info --editable --width 650 --height 400)

zSavePath=$(echo -n "$(zenity --file-selection --filename="$zOpenPath" --save --confirm-overwrite)")

echo -n "$zNewData" &gt; "$zSavePath"

fi

}

f\_file(){ #this function is a file managment it wil perform all the task of file mangent by only a single click there is no need to enter any command on terminal all the comand while automatically will write by click on the task you want to perform.

a="<span background='Orange'> <span font='15' color='black'>\nFile Mangement System\n</span></span>" #it set the color adn setthing of the samll window by using zenity working.

while true; do #first time the condition will be true to check the belwo condition.

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="File Mangement" --text="${a}" --list --column "Choose Options:" "New File" "open" "Current Position" "Show All Files" "list the content of a directory" "list all the content of a directory including the hidden files" "create a new directory" "Delete an existing directory" "creates a new File" "Delete a File" "Forcefully Delete File" "Copies the content of File File1 into File File2" "Rename files and directories" "print the content of a File" "print the first 10 lines of a file." "print the Last 10 lines of a file." "Count All File" "Find File" "File Compress" "File Decompress" "Exit")"

#the above menu\_luch function are the first interface where all the function are wrote from this fucntion we call the function.it is the main function.

if [ "$menu\_launch" = "Exit" ]; then # it will call the mein function if we cancel this window we can move forward and backward trough this main window of i nterface.

f\_menu

echo done

exit

elif [ "$menu\_launch" = "New File" ]; then #this option will crate the new file and it will display the command on terminal when we exit from the program.

echo $(gedit) #gedit function always move you to the text editor where you can save and make new file.if someone did not know how to make new file they can use this commadn nit will directly move user top the text editor.

exit

elif [ "$menu\_launch" = "open" ]; then #This option will open the new file.which the user wanted to open and it will display the command on terminal when we exit from the program.

f\_open #it again call the fucntion for opening any file this function define above where the command are written .

exit

elif [ "$menu\_launch" = "Current Position" ]; then #this option will tell the current working area of the user like home ,desktop etc and it will display the command on terminal when we exit from the program.

a=$(pwd |zenity --text-info --editable --width 650 --height 400) #it store the current position of the directory or the user wrking area in a variable by using pwd command.

echo $a #simple print the store address

elif [ "$menu\_launch" = "Show All Files" ]; then #this option will show all file of current directory from the user list and it will display the command on terminal when we exit from the program.

a=$(file \* |zenity --text-info --editable --width 650 --height 400) #it store the name of all the file in a variable by using file command.

echo $a #it print the store files name.it print all files list on terminal and as wellas on the text box.

elif [ "$menu\_launch" = "list the content of a directory" ]; then #This option only display the content of the current directory and it will display the command on terminal when we exit from the program.

a=$(ls |zenity --text-info --editable --width 650 --height 400) #ls stand for list which is used to show the content of a directory or it is the shell command that lists files and directories within a directory.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "list all the content of a directory including the hidden files" ]; then #this option will list all the content of a directory including the hidden files and it will display the command on terminal when we exit from the program.

a=$(ls -la |zenity --text-info --editable --width 650 --height 400) #this list command with -la

echo $a #it print the all information which stored in the varibale.ls use with combination of -la which means to show the hidden file too.

elif [ "$menu\_launch" = "create a new directory" ]; then #This option create a new directory and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(mkdir $b |zenity --text-info --editable --width 650 --height 400)

echo $a #it print the all information which stored in the varibale and the varable will print that info on terminal.

elif [ "$menu\_launch" = "Delete an existing directory" ]; then #This option will Delete an existing directory and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(rmdir $b |zenity --text-info --editable --width 650 --height 400) #editable means display text on daiolgbox can be change and also text dialog box will appear and the data we write on it will be display on terminal.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "creates a new file" ]; then #This option create a new file and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #touch command create new file and its primary function to modify the timestamp.

a=$(touch $b |zenity --text-info --editable --width 650 --height 400) #perform operation and store in varable of a

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "Delete a File" ]; then #This option Delete a File and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(rm $b |zenity --text-info --editable --width 650 --height 400)

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "Forcefully Delete File" ]; then #This option will Forcefully Delete File and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(rm -f $b |zenity --text-info --editable --width 650 --height 400)

echo $a #it print the all information which stored in the varibale.rm command delete file -rf also do same thing but it delete it permanently.

elif [ "$menu\_launch" = "Copies the content of File File1 into File File2" ]; then #This option will Copies the content of File File1 into File File2 and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

c=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(cp $b $c |zenity --text-info --editable --width 650 --height 400) #cp command copy the content of the file and move the content of the file from one to another.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "Rename files and directories" ]; then #This will option Rename files and directories and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

c=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(mv -r $b $c |zenity --text-info --editable --width 650 --height 400) #it move file from one directory into another dirctory and also rename file and directory.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "print the content of a file" ]; then #This option will print the content of a file and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #taking input in varable by using dialog box.

a=$(cat $b |zenity --text-info --editable --width 650 --height 400)

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "print the first 10 lines of a file." ]; then #This option will print the first 10 lines of a file and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #it take the file name and store in the variable .

a=$(head $b |zenity --text-info --editable --width 650 --height 400) #head command take the first ten lines of the program from the top.

echo $a #it print the all information which stored in the varibale on the termianl by picking the lines of character first ten.

elif [ "$menu\_launch" = "print the Last 10 lines of a file." ]; then #This option will print the Last 10 lines of a file and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello")

a=$(tail $b |zenity --text-info --editable --width 650 --height 400) #tail command take the last ten lines of the program from the bottom or it is the opposite of head command.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "Count All File" ]; then #This option Count All File and it will display the command on terminal when we exit from the program.

b=$(zenity --entry text="Hello") #it store on the variable

a=$(wc \*$b |zenity --text-info --editable --width 650 --height 400) #wc command to count a file word,line ,characteror byte count.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "Find File" ]; then #This option Find File and it will display the command on terminal when we exit from the program.

b=$(zenity --entry --title="Which File You want To Find Enter Extension" text="Hello") #it store the extenion of the file on the variable

a=$(find \*$b |zenity --text-info --editable --width 650 --height 400) #ind command search the file you want to find on the computer.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "File Compress" ]; then #This option will Compress file and it will display the command on terminal when we exit from the program.

b=$(zenity --entry --title="Enter File With Extension" text="Hello") #take the extension of the file you want to compress

a=$(gzip $b |zenity --text-info --editable --width 650 --height 400) #gzip command compress the file.

echo $a #it print the all information which stored in the varibale.

elif [ "$menu\_launch" = "File Decompress" ]; then #This option will File Decompress and it will display the command on terminal when we exit from the program.

b=$(zenity --entry --title="Enter Extension" text="Hello")

a=$(gzip -d $b.gz |zenity --text-info --editable --width 650 --height 400) #-d with gzip is used to decompress the file or you can use gunzip but we used this -d bc we can easily remember.there are aslo -r adn -l and many more with alience gzip used but they have all different purpose.

echo $a #it print the all information which stored in the varibale.

else

f\_menu #closing statement

fi #close the if condition

done

}

f\_user(){ #This function is used to add any user it only add the user and store info in USE.txt file

a=$(zenity --forms --title="Add User" \

--text="Enter information about your User." \ #taking input from user by using zenity dialog

--separator="," \

--add-entry="User Name" \ #taking input from user by using zenity dialog

--add-password="Password" \ #taking input from user by using zenity dialog

--add-password="Confirm Password" >>USE.txt) #the use.txt file store the name password in the file and store iun the dirctory

if [ $a = 0 ]; then

zenity --error

fi

case $? in

0)

echo "Friend added.";; #it print the message when user add simple by dialog

1)

echo "No friend added." #it print the message when user not add simple by dialog

;;

-1)

echo "An unexpected error has occurred." #it print the message when user not add simple by dialog

;;

esac

}

f\_system\_info(){

a=$(lshw -short|zenity --text-info --width 650 --height 400) #lshw is a command -line tool to extract detailed information on the harware configuration of the machine.

echo $a

}

f\_pack(){

a="<span background='Orange'><span font='30' color='black'>\nPackage Management\n</span></span>"

while true; do

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="Packages" --text="${a}" --list --column "Choose Options:" "nmap" "nmap remove" "update" "dpkg l" "dpkg -l | grep apache2" "dpkg -L ufw" "unattended-upgrades" "install apticron" "EXIT")"

if [ "$menu\_launch" = "EXIT" ]; then

f\_menu

echo done

exit

elif [ "$menu\_launch" = "nmap" ]; then #this can install many module and libraries in os

a=$(sudo apt install nmap|zenity --text-info --width 650 --height 400) #nmap is a network mapper is an open source security auditin and network scanning software designed by gordon lyon.it is developed in such a way that it can quickly analyze massive network as well as single hsots.

echo $a #sudo=permit user to execute command as super user.apt get =Get a packae from the advance packing tool install operation to do.

elif [ "$menu\_launch" = "nmap remove" ]; then

a=$(sudo apt remove nmap|zenity --text-info --width 650 --height 400) #this can remove the store module uninstall or opposite of nmap

echo $a

elif [ "$menu\_launch" = "update" ]; then

a=$(sudo apt update|zenity --text-info --editable --width 650 --height 400) #this can only update the existin module or libraries

echo $a

elif [ "$menu\_launch" = "dpkg l" ]; then #the differnce between dpkg and apt ,apt is a front end and dpkg is more user friendly than the earlier select front end while dpkg handle individual package activitiesapt handle package relationship as well as the sourcing and administration of higher level versioning choices .

a=$(dpkg -l|zenity --text-info --width 650 --height 400) #it is the software which can delete modifya and install and provide info .deb package .dpkg itself a low level tool.it is package management system.

echo $a #apt is a front end dpkg.-l is a query action which list all package from dpkg database.the dpkg command detect the foriegn option and runs dpk-query automatically.

elif [ "$menu\_launch" = "dpkg -l | grep apache2" ]; then

a=$(dpkg -l | grep apache2 |zenity --text-info --width 650 --height 400) #apache2 is the commonly used web server on linux system.web server are used to serve web pagerequested by client computer.

echo $a

elif [ "$menu\_launch" = "dpkg -L ufw" ]; then

a=$(dpkg -L ufw|zenity --text-info --width 650 --height 400) #it is a fronted for iptables and is particular well suited for host based firewalls. ufw provide a framework for managing netfilter as well as a command line interface for manipulating the firewell.

echo $a #ufw is frontend for managing firewall rules in arch linux debian or ubuntu.

elif [ "$menu\_launch" = "unattended-upgrades" ]; then

a=$(sudo apt install unattended-upgrades|zenity --text-info --width 650 --height 400) #it uprade all the module which unattend by the user.

echo $a

elif [ "$menu\_launch" = "install apticron" ]; then

a=$(sudo apt install apticron|zenity --text --width 650 --height 400) #apticorn is a shell script which generates a amil with a list of all package a mail with a list of all packages currently pending an upgrade as well as summary of change using apt listchange to a cofigurable email address.

echo $a

else

f\_menu

fi

done

}

f\_utilities(){

a="<span background='Orange'><span font='30' color='black'>\nUtilities\n</span></span>"

while true; do

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="Utilities" --text="${a}" --list --column "Choose Options:" "Display informantion of cmd" "Pattern of cmd" "Info of pseeudo cmd" "Short dump of information" "Summary of the memory usage" "Group" "History" "Kill" "Man" "Passupdate" "Running processes" "Force shutdown" "Watch" "Top -Ss" "Shh cmd" "EXIT")"

if [ "$menu\_launch" = "EXIT" ]; then

f\_menu

echo done

exit

elif [ "$menu\_launch" = "Display informantion of cmd" ]; then

a=$(help -d |zenity --text-info --editable --width 650 --height 400) #help command which as its nae=me say help you to learn any built in command it display info about shell built in commands.

echo $a #help -d used when you just want to get an overview about any shell built in command it only give short description

elif [ "$menu\_launch" = "Pattern of cmd" ]; then

a=$(help -s |zenity --text-info --editable --width 650 --height 400)

echo $a

elif [ "$menu\_launch" = "Info of pseeudo cmd" ]; then

a=$(help -m |zenity --text-info --editable --width 650 --height 400) #this is used to display info about a command in pseudo manpage format.

echo $a

elif [ "$menu\_launch" = "Short dump of information" ]; then

a=$(finger |zenity --text-info --editable --width 650 --height 400) #finger command is a userinfo lookup command which give detail of all the user logged in.

echo $a

elif [ "$menu\_launch" = "Summary of the memory usage" ]; then

a=$(free -h |zenity --text-info --editable --width 650 --height 400) #free command provide us valuable info of available Ram in linuxunix machine.

echo $a

elif [ "$menu\_launch" = "Group" ]; then

a=$(groups |zenity --text-info --editable --width 650 --height 400) #the group command lists the group that a user belongs to.A user can have only one primary group at a time.

echo $a

elif [ "$menu\_launch" = "History" ]; then

a=$(!188 |zenity --text-info --editable --width 650 --height 400)

echo $a

elif [ "$menu\_launch" = "Kill" ]; then

a=$(ps |zenity --text-info --editable --width 650 --height 400) #it display the of the process that can currently running and obtain additional info about these processes .

echo $a

elif [ "$menu\_launch" = "Man" ]; then

a=$(Man chown |zenity --text-info --editable --width 650 --height 400) #it change the user ownershipof each given file.

echo $a

elif [ "$menu\_launch" = "Pass update" ]; then

a=$(sudo passwd |zenity --text-info --editable --width 650 --height 400) #the passwd chane the password for user account.sudo means it will run that command with elevated privilage.

echo $a

elif [ "$menu\_launch" = "Force shutdown" ]; then

a=$(shutdown now |zenity --text-info --editable --width 650 --height 400) #it shutdown the systema nd we can also used systemctl halt fo same purpose.

echo $a

elif [ "$menu\_launch" = "Shh cmd" ]; then

a=$(shh |zenity --text-info --editable --width 650 --height 400) #this commadn is used to start the ssh client program that enable secure connection to ssh server on a remote machine.it provide encrypted cnnection between two host.

echo $a

elif [ "$menu\_launch" = "Top -Ss" ]; then

a=$(top -Ss |zenity --text-info --editable --width 650 --height 400) #it is used to display system performance info.it is also used to monitor memory.

echo $a #-s is used secure mode operation#commulative time toggle.-S commulative time toggle to find.

elif [ "$menu\_launch" = "Watch" ]; then

a=$(watch |zenity --text-info --editable --width 650 --height 400)

echo $a

else

f\_menu

fi

done

}

f\_storage(){

a="<span background='Orange'><span font='30' color='black'>\nStorage\n</span></span>"

while true; do

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="Storage" --text="${a}" --list --column "Choose Options:" "Memory Usage" "Memory Usage Statistics" "Check Storage Byte" "Check Storage KiloByte" "Check Storage MegaByte" "Check Storage GigaByte" "Check Storage TeraByte" "High and Low Memory Stats" "Exit")"

if [ "$menu\_launch" = "Exit" ]; then

f\_menu

echo done

exit

elif [ "$menu\_launch" = "Memory Usage" ]; then

a=$(free |zenity --text-info --editable --width 650 --height 400) #The linux free command output a summary of Ram usage including total used free shared and available memory and swap space.

echo $a

exit

elif [ "$menu\_launch" = "Memory Usage Statistics" ]; then

a=$(vmstat |zenity --text-info --editable --width 650 --height 400) #the vmstat command (short for virtual memory statistics) is a built in monitorin utility in linux .the command is used to obtain information about memory system processes,paging interrupts, block i/o,disk and cpu scheduling.

echo $a

elif [ "$menu\_launch" = "PerProcess Detail" ]; then

a=$(top |zenity --text-info --editable --width 650 --height 400) #the top command is used for memory monitoring.it works only on linux platform.

echo $a

elif [ "$menu\_launch" = "Deep Perprocess Detail" ]; then

a=$(htop |zenity --text-info --editable --width 650 --height 400) #it determine the cause of load by each process.

echo $a

elif [ "$menu\_launch" = "Check Storage Byte" ]; then

a=$(free -b |zenity --text-info --editable --width 650 --height 400) #free command output a summary of ram usage including total used free shared and available memory and swap space.

#-d is used with free -d display the memory amount in bytes.we can also used -k -m -g --tebi --pebi -h these are the different memory unit.

echo $a

elif [ "$menu\_launch" = "Check Storage KiloByte" ]; then

a=$(free -k |zenity --text-info --editable --width 650 --height 400) #-d is used with free -d display the memory amount in bytes.we can also used -k -m -g --tebi --pebi -h these are the different memory unit.

echo $a

elif [ "$menu\_launch" = "Check Storage MegaByte" ]; then

a=$(free -m |zenity --text-info --editable --width 650 --height 400) #-d is used with free -d display the memory amount in bytes.we can also used -k -m -g --tebi --pebi -h these are the different memory unit.

echo $a

elif [ "$menu\_launch" = "Check Storage GigaByte" ]; then

a=$(free -g |zenity --text-info --editable --width 650 --height 400) #-d is used with free -d display the memory amount in bytes.we can also used -k -m -g --tebi --pebi -h these are the different memory unit.

echo $a

elif [ "$menu\_launch" = "Check Storage TeraByte" ]; then

a=$(free --tera |zenity --text-info --editable --width 650 --height 400) #-d is used with free -d display the memory amount in bytes.we can also used -k -m -g --tebi --pebi -h these are the different memory unit.

echo $a

elif [ "$menu\_launch" = "High and Low Memory Stats" ]; then

a=$(free -l |zenity --text-info --editable --width 650 --height 400) #it show the status of high and low

echo $a

else

f\_menu

fi

done

}

f\_browser(){

zenity --info --text="Fire fox has being Opened"

firefox -p #this -p is used to open the file

}

f\_process(){

a="<span background='Orange'><span font='30' color='black'>\nProcess\n</span></span>"

while true; do

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="Process" --text="${a}" --list --column "Choose Options:" "Running Processes" "Visualize Processes" "All-User Processes" "Priority Process" "File System Information" "Hidden File Information" "Run/Stop in Foreground" "Send Process to Background" "Status of Process" "Returns Process ID" "Kill Process" "Display Active Process" "Display All Process" "Full Format Listing" "Process Owned" "Display Process by User Real ID" "All Process running as Root" "List Process by Group" "List Process by Effective Group" "Print Process Tree" "To List All Specifiers" "View Pid PPID UserName And Command" "Exit")"

if [ "$menu\_launch" = "Exit" ]; then

f\_menu

echo done

exit

elif [ "$menu\_launch" = "Running Processes" ]; then

a=$(ps |zenity --text-info --editable --width 650 --height 400) #the ps command linux provide a utility called ps(process status) for viewin information related with the processes on a system.it list the cureently runnin processand thier pids along with some other information which depend on different option.

echo $a

elif [ "$menu\_launch" = "Visualize Processes" ]; then #the pstree command linux that show the running processes as a tree.

a=$(pstree -g |zenity --text-info --editable --width 650 --height 400) #the pstree command linux that show the running processes as a tree.

echo $a

elif [ "$menu\_launch" = "All-User Processes" ]; then

a=$(ps aux |zenity --text-info --editable --width 650 --height 400) #ps aux command is a tool to monitor processes running on your linux system.

echo $a

elif [ "$menu\_launch" = "Priority Process" ]; then

a=$(nice |zenity --text-info --editable --width 650 --height 400) #the nice command lets you run a command at a priority lower than the command normal priority.

echo $a

elif [ "$menu\_launch" = "File System Information" ]; then

a=$(df |zenity --text-info --editable --width 650 --height 400) #The df command display the amount of disk space available on the filesystem with each file names argument.

echo $a

elif [ "$menu\_launch" = "Hidden File Information" ]; then

a=$(df -h |zenity --text-info --editable --width 650 --height 400) #the -h option is used to display the disk space in a human readable form.

echo $a

elif [ "$menu\_launch" = "Run/Stop in Foreground" ]; then

a=$(fg |zenity --text-info --editable --width 650 --height 400) #the fg command switches a job running in the background into the foreground.

echo $a

elif [ "$menu\_launch" = "Send Process to Background" ]; then

a=$(bg |zenity --text-info --editable --width 650 --height 400) #the bg command in linux better known as the background command is a job control command in linux /unix like system.

echo $a

elif [ "$menu\_launch" = "Status of Process" ]; then

a=$(ps PID |zenity --text-info --editable --width 650 --height 400) #the command it provide utility called ps ps comamdn is used to list the currently running processes and their pid along with some of other information which depends on different option.

echo $a

elif [ "$menu\_launch" = "Returns Process ID" ]; then

a=$(pidof |zenity --text-info --editable --width 650 --height 400) #it find the process id of the named program.

echo $a

elif [ "$menu\_launch" = "Kill Process" ]; then

a=$(kill PID |zenity --text-info --editable --width 650 --height 400) #kill command kill the process simply.

echo $a

elif [ "$menu\_launch" = "Display Active Process" ]; then

a=$(ps -A |zenity --text-info --editable --width 650 --height 400) #ps with \_A is used to show the active process.

echo $a

elif [ "$menu\_launch" = "Display All Process" ]; then

a=$(ps au |zenity --text-info --editable --width 650 --height 400) #ps with au is used to show all processes.

echo $a

elif [ "$menu\_launch" = "Full Format Listing" ]; then

a=$(ps -ef |zenity --text-info --editable --width 650 --height 400) #linux command ps -ef is used to print all the processes running on the system in the standard format.

echo $a

elif [ "$menu\_launch" = "Process Owned" ]; then

a=$(ps -x |zenity --text-info --editable --width 650 --height 400) #the x option instruction ps to list the processes without a controling terminal.

echo $a

elif [ "$menu\_launch" = "Display Process by User Real ID" ]; then

a=$(ps -fu |zenity --text-info --editable --width 650 --height 400) #to select a user processes by effective user id.

echo $a

elif [ "$menu\_launch" = "All Process running as Root" ]; then

a=$(ps -U root -u root |zenity --text-info --editable --width 650 --height 400) #this command enable you to view every process running with root user privilage real effective id in user format.

echo $a

elif [ "$menu\_launch" = "List Process by Group" ]; then

a=$(ps -fG apache |zenity --text-info --editable --width 650 --height 400) #if we want to list all processes owned by a certain group realgroup id or name type.

echo $a

elif [ "$menu\_launch" = "List Process by Effective Group" ]; then

a=$(ps -fg apache |zenity --text-info --editable --width 650 --height 400) #if we want to list all processes owned by a certain group realgroup id or name type.

echo $a

elif [ "$menu\_launch" = "Print Process Tree" ]; then

a=$(ps -e --forest |zenity --text-info --editable --width 650 --height 400) #a process tree shows how processes on the system are linked to each other,processes whose parents have been killed are adopted by the init .

echo $a

elif [ "$menu\_launch" = "To List All Specifiers" ]; then

a=$(ps L |zenity --text-info --editable --width 650 --height 400) #to list all format specifers,include the l flag.

echo $a

elif [ "$menu\_launch" = "View Pid PPID UserName And Command" ]; then

a=$(ps -eo pid,ppid,user,cmd |zenity --text-info --editable --width 650 --height 400) #the command below aloow you to view the pid ,ppid user name and command of a process.

echo $a

else

f\_menu

fi

done

}

f\_calendar(){

zenity --info --text="Calendar has being Opened"

zenity --calendar #we have to use zenity --calender to read data from user and it show the calender box it is the zenity command to show the calender box.

}

f\_NewTerminal(){

zenity --info --text="New Terminal has being Opened"

gnome-terminal #gnome terminal is a terminal emulator for the Gnme desktop envirmoment written by Havooc pennington and other.terminal emulator allow user to access a unix shell while remaining on their graphical desktop.

}

f\_network(){

a="<span background='Orange'><span font='30' color='black'>\nNetwork\n</span></span>"

while true; do

menu\_launch="$(zenity --width 400 --height 500 --cancel-label="Go-Back" --ok-label="Execute" --title="Network" --text="${a}" --list --column "Choose Options:" "PC Name" "IP" "Ping" "WLAN IP" "NSLOOKUP" "NetStat" "Active Conn" "Exit")"

if [ "$menu\_launch" = "Exit" ]; then

f\_menu

echo done

exit

elif [ "$menu\_launch" = "PC Name" ]; then

a=$(hostname |zenity --text-info --editable --width 650 --height 100) #hostname commadn show the owner name of the pc

echo $a

elif [ "$menu\_launch" = "IP" ]; then

a=$(hostname -I |zenity --text-info --editable --width 650 --height 100) #hostname with combination of -i shhow ip.

echo $a

elif [ "$menu\_launch" = "Ping" ]; then

a=$(zenity --entry --title="IP Address" --text="Enter the IP Address") #it take ip fro user

echo $a

b=$(ping $a |zenity --text-info --editable --width 650 --height 400) #it used pip structure to used the input a for b and ping command is used to check this network is present or not.

echo $b

elif [ "$menu\_launch" = "WLAN IP" ]; then

a=$(pwd |zenity --text-info --editable --width 650 --height 400) #pwd prints the current working directoery pathstartin from root.

echo $a

elif [ "$menu\_launch" = "NSLOOKUP" ]; then

a=$(zenity --entry --title="IP Address" --text="Enter the IP Address")

echo $a

b=$(nslookup $a |zenity --text-info --editable --width 650 --height 400) #nslookup command name server lookup is a useful command for getting info from Dsn server.

echo $b

elif [ "$menu\_launch" = "NetStat" ]; then

a=$(netstat -r |zenity --text-info --editable --width 650 --height 400) #netstat is a networking tool used for troubleshooting and config that can also server as a monitoring tool for connection over the network.

echo $a

elif [ "$menu\_launch" = "Active Conn" ]; then

a=$(ss |zenity --text-info --editable --width 650 --height 400) #it rovide the extensive detailof the socket that provide comunication between systems.this is used to dump socket statistics on linux system.it is also replacement of netstat commadn and is often use for trouble shooting problem.

echo $a

else

f\_menu

fi

done

}

f\_datetime(){

#!/bin/bash

# Time, Date and Timezone configuration utility for Linux Lite

# Fork of Date and Time Setting Tool Copyright 2009,2011 by Tony Brijeski under the GPL V2

# TODO: add timezone settings, test on a live dvd, use image-list for the main dialog,

DIALOG="`which zenity` --width 400"

TITLE="--window-icon=/usr/share/icons/hicolor/24x24/apps/gnome-panel-clock.png --print-partial --title="

TEXT="--text="

ENTRY="--entry "

ENTRYTEXT="--entry-text "

MENU="--list --print-column=1 --column=Pick --column=Info"

YESNO="--question "

MSGBOX="--info "

SCALE="--scale "

PASSWORD="--entry --hide-text "

TITLETEXT="Linux Lite Date and Time Setting Tool"

if [ $EUID -ne 0 ]; then

gksu -m '

<b>Lite Time requires Administrative privileges</b>

Please enter your password to continue.' bash $0

exit

else :

fi

while [ "$SETCHOICE" != "Exit" ]; do

DAY="`date +%d`"

MONTH="`date +%m`"

YEAR="`date +%Y`"

MINUTE="`date +%M`"

HOUR="`date +%H`"

SETCHOICE=`$DIALOG --height 300 $TITLE"$TITLETEXT" $MENU $TEXT"Set Year Date and Time Tool\n\nTime=$HOUR:$MINUTE\nDate=$MONTH-$DAY-$YEAR\n\n" Exit "Quit" SETTIME "Set Current Time" SETDATE "Set Current Date"`

SETCHOICE=`echo $SETCHOICE | cut -d "|" -f 1`

if [ "$SETCHOICE" = "SETTIME" ]; then

HOUR="`date +%H`"

HOUR=`echo $HOUR | sed -e 's/^0//g'`

SETHOUR=`$DIALOG $TITLE"$TITLETEXT" $SCALE --value=$HOUR --min-value=0 --max-value=23 $TEXT"Move the slider to the correct Hour"`

if [ "$?" = "0" ]; then

if [ "${#SETHOUR}" = "1" ]; then

SETHOUR="0$SETHOUR"

fi

MINUTE="`date +%M`"

MINUTE=`echo $MINUTE | sed -e 's/^0//g'`

fi

SETMINUTE=`$DIALOG $TITLE"$TITLETEXT" $SCALE --value=$MINUTE --min-value=0 --max-value=59 $TEXT"Move the slider to the correct Minute"`

if [ "$?" = "0" ]; then

if [ "${#SETMINUTE}" = "1" ]; then

SETMINUTE="0$SETMINUTE"

fi

date $MONTH$DAY$SETHOUR$SETMINUTE$YEAR

hwclock --systohc

fi

fi

if [ "$SETCHOICE" = "SETDATE" ]; then

DAY="`date +%d`"

DAY=`echo $DAY | sed -e 's/^0//g'`

MONTH="`date +%m`"

MONTH=`echo $MONTH | sed -e 's/^0//g'`

YEAR="`date +%Y`"

SETYEAR=`$DIALOG $TITLE"$TITLETEXT" $SCALE --value=$YEAR --min-value=2000 --max-value=2020 $TEXT"Move the slider to the correct Year"`

if [ "$?" = "0" ]; then

SETMONTH=`$DIALOG $TITLE"$TITLETEXT" $SCALE --value=$MONTH --min-value=1 --max-value=12 $TEXT"Move the slider to the correct Month"`

if [ "$?" = "0" ]; then

if [ "${#SETMONTH}" = "1" ]; then

SETMONTH="0$SETMONTH"

fi

SETDAY=`$DIALOG $TITLE"$TITLETEXT" $SCALE --value=$DAY --min-value=1 --max-value=31 $TEXT"Move the slider to the correct Day"`

if [ "$?" = "0" ]; then

if [ "${#SETDAY}" = "1" ]; then

SETDAY="0$SETDAY"

fi

MINUTE="`date +%M`"

HOUR="`date +%H`"

date $SETMONTH$SETDAY$HOUR$MINUTE$SETYEAR

hwclock --systohc

fi

fi

fi

fi

done

exit 0

}

#the main function will be call from anohter place indirectly by giving correct user and password it will call the main function where all the working of the function are stored.

f\_menu(){

#it declare different varibal to store info.

WEEKDAY=`date +"%A"`

MONTH=`date +"%B"`

DAYOFMONTH=`date +"%d"`

YEAR=`date +"%Y"`

DATE=`date +"%F"`

TIME=`date +"%I:%M %p %Z"`

SHOWDATE="${MONTH} ${DAYOFMONTH}, ${YEAR}"

DATE\_TIME\_TO\_SHOW=" #This code of line show the overall intro and existing time.

<span background='Orange'><span color='black'font='20'>Mini OS\n</span> #it dispaly the project name span is used to mark up the text from start to end point.

<span font='15' color='blue'><tt>Day : </tt></span><span font='15' color='darkblue'>$WEEKDAY\n</span> #it dispaly the day .

<span font='15' color='blue'><tt>Date: </tt></span><span font='15' color='darkblue'>$SHOWDATE\n</span> #it dispaly the date.

<span font='15' color='blue'><tt>Time: </tt></span><span font='15' color='darkblue'>$TIME\n</span> #it dispaly the time.

<span font='15' color='blue'><tt>Project Members:</tt></span><span font='10' color='darkblue'>NAEEM REHMAN | ALI GUAHAR | JAHANZEB KHAN | </span></span> " #it dipaly the group member name.

#The below function will be execute when we press the launch button on the gui box .it will direclty open the new window on which differnt function will lsit down .

while true; do

menu\_launch="$(zenity --width 600 --height 1200 --title="Mini OS" --text="${DATE\_TIME\_TO\_SHOW}" --cancel-label="Shutdown" --ok-label="Start Module" --list --radiolist --column "Select Module And Click Start Module Button" --column "Module Information" TRUE "File Management" FALSE "Storage" FALSE "Process" FALSE "Open Terminal" FALSE "System" False "Calendar" FALSE "Open Browser" FALSE "Network Details" FALSE "Utilities" FALSE "Add New User" FALSE "Package Management")"

#list command is used to show the full list or content of the direcory.

#start module will call the function which will selecty by the user and list function display the content of the selected function.

#Date time to show stored different information of intro project.so it will call by within the menu\_lunch function to dispaly all the information.

if [ "$menu\_launch" = "Exit" ]; then #if we press shutdown button then it will not move to the next stage and program will be finsihed and we will comback to the terminal.

echo done

exit

elif [ "$menu\_launch" = "File Management" ]; then #if the user select file management option then it will call the file mangemnt function .

f\_file

elif [ "$menu\_launch" = "Storage" ]; then #if the user select Storage option then it will call the Storage function .

f\_storage

elif [ "$menu\_launch" = "Process" ]; then #if the user select process option then it will call the process function

f\_process

elif [ "$menu\_launch" = "Open Terminal" ]; then #if the user select Open Terminal option then it will call the Open Terminal function

f\_NewTerminal

elif [ "$menu\_launch" = "System" ]; then #if the user select System option then it will call the System function

f\_system\_info

elif [ "$menu\_launch" = "Calendar" ]; then #if the user select Calendar option then it will call the Calendar function

f\_calendar

elif [ "$menu\_launch" = "Open Browser" ]; then #if the user select Open Browser option then it will call the Open Browser function

f\_browser

elif [ "$menu\_launch" = "Network Details" ]; then #if the user select Network Details option then it will call the Network Details function

f\_network

elif [ "$menu\_launch" = "Utilities" ]; then #if the user select Utilities option then it will call the Utilities function

f\_utilities

elif [ "$menu\_launch" = "Add New User" ]; then #if the user select Add New User then it will call the Add New User function

f\_user

elif [ "$menu\_launch" = "Package Management" ]; then #if the user select Package Management option then it will call the Package Management function

f\_pack

else

exit

fi

done

}

# This function will be used first at the start of program.

usr=$(zenity --width 300 --height 100 --entry --text="Enter Your Username" --title="Login") # it ask the user name with zenity command tell the inforamtion of box on which it take the input.

#zenity is a free software and a cross platform program that allows the execution of GTK dialog boxes in commandline and shell scripts.GTK is a free and open source crossplatform widget toolkit for creating graphical user interfaces.it is one of the most popular tool kit.it is stand for Gimp toolkit.it is a object oriented toolkit written in c.

pass=$(zenity --width 300 --height 100 --password --text="Enter Your Password" --title="Login") #we provide the correct information of user name and password and this if statment is used to compare with the correct user name and password .

if test "$usr" = 'AliGauhar' && test "$pass" = 'NaeemGroup' #if the password and user name is correct it will process next step.

then

zenity --width 300 --height 100 --info --text="Login Successfull\n\n\nWelcome To Mini OS" --ok-label="Launch" --title="Login" #if provided usr name and password is correct it will tell the user through sms.

f\_menu #This is the main function of program where all the function are stored.we call this function when if condition is true.

else

zenity --width 300 --height 100 --error --text="Acess Denied" #if the above condition is false then it will be display error mesage on screen .

fi #this is the closing statement of if condition.

**CONCLUSION**

This project will help the new user who are beginner and want to learn linux command.we use zenity because it is a free software and a cross platform program that allows the execution of GTK dialog boxes in command line and shell scripts.GTK is a free and open source cross platform widget toolkit for creating graphical user interfaces.it is one of the most popular tool kit.it is stand for Gimp toolkit.it is a object oriented toolkit written in c and it make the communication of system and user are friendly in the form of GUI. We made this mini operating system with basic modules like system,process, user, network, storage, file etc. This operating system is based on Linux command. As we know that this is a system OS which maintains and manages the system functionality by guiding the hardware. It is also called as set of programs that control the hardware. Kernel provides path to the peripheral device it is also responds to the calls and invoke the devices according to the call.