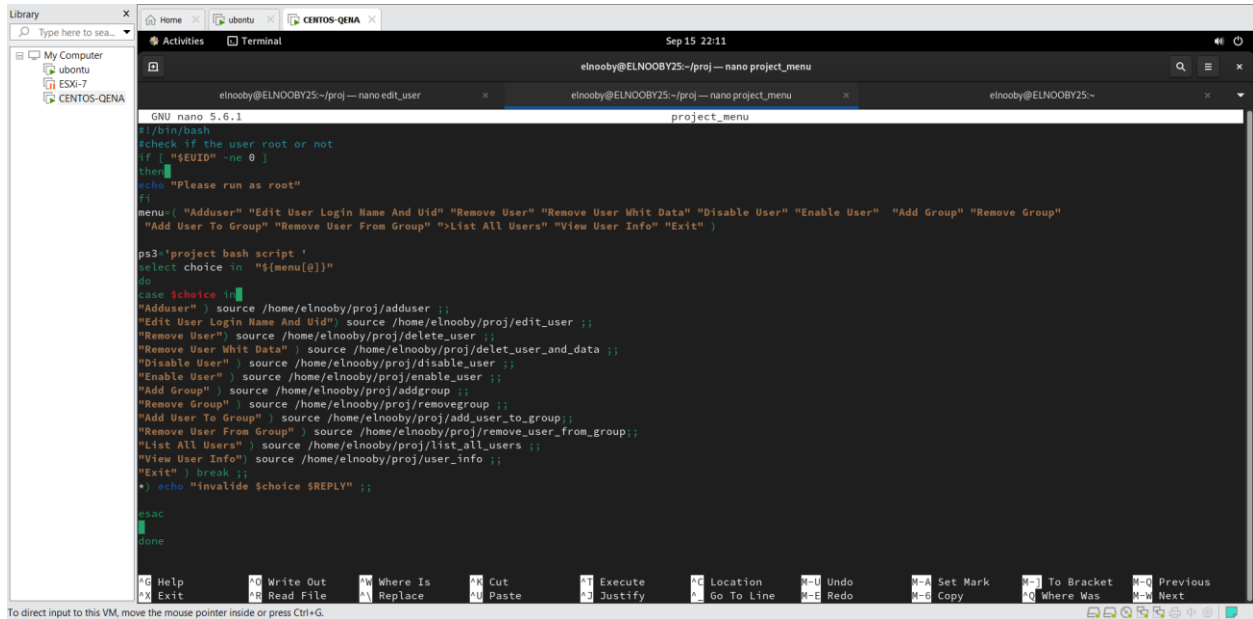


➤ Script menu



```
GNU nano 5.6.1 project_menu
#!/bin/bash
#check if the user root or not
if [ "$SEUID" -ne 0 ]
then
echo "Please run as root"
fi
menu=( "Adduser" "Edit User Login Name And Uid" "Remove User" "Remove User Whit Data" "Disable User" "Enable User" "Add Group" "Remove Group"
"Add User To Group" "Remove User From Group" ">List All Users" "View User Info" "Exit" )

ps3="project bash script "
select choice in "${menu[@]}"
do
case $choice in
"Adduser" ) source /home/elnooby/proj/adduser ;;
"Edit User Login Name And Uid" ) source /home/elnooby/proj/edit_user ;;
"Remove User" ) source /home/elnooby/proj/delete_user ;;
"Remove User Whit Data" ) source /home/elnooby/proj/delet_user_and_data ;;
"Disable User" ) source /home/elnooby/proj/disable_user ;;
"Enable User" ) source /home/elnooby/proj/enable_user ;;
"Add Group" ) source /home/elnooby/proj/addgroup ;;
"Remove Group" ) source /home/elnooby/proj/removegroup ;;
"Add User To Group" ) source /home/elnooby/proj/add_user_to_group ;;
"Remove User From Group" ) source /home/elnooby/proj/remove_user_from_group ;;
">List All Users" ) source /home/elnooby/proj/list_all_users ;;
"View User Info" ) source /home/elnooby/proj/user_info ;;
"Exit" ) break ;;
* ) echo "invalide $choice $REPLY" ;;
esac
done
```

➤ Script adduser

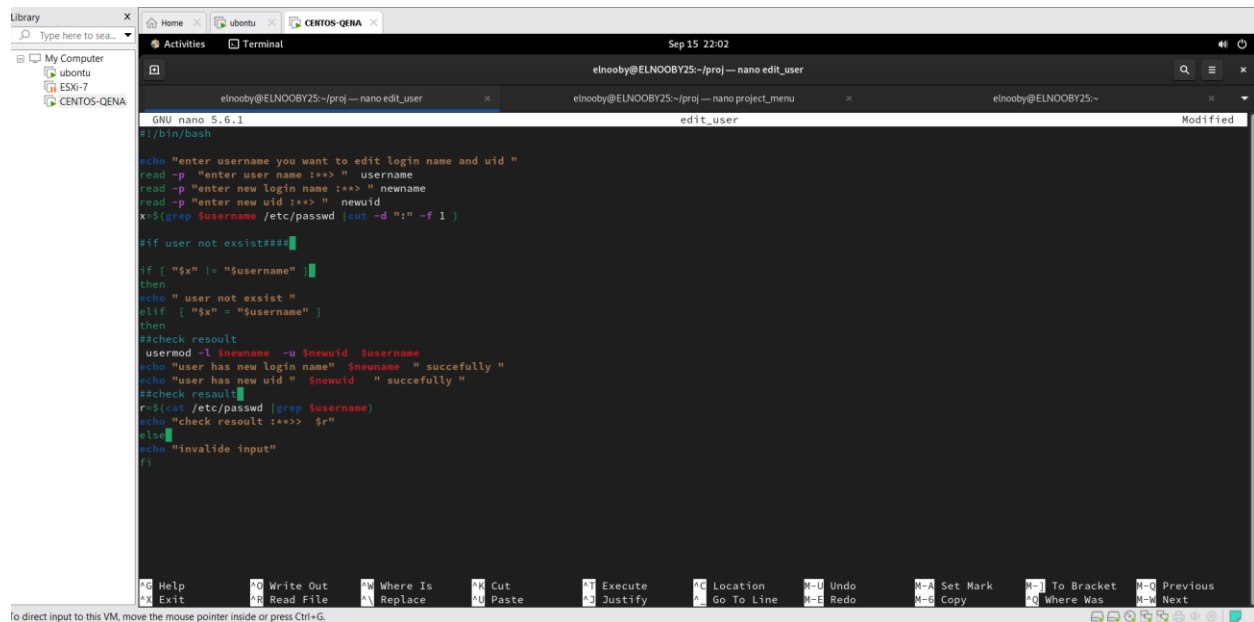


```
GNU nano 5.6.1 adduser
#!/bin/bash
echo "enter username you want to added"
read -p "enter user name :>>> " username

x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

#if user not exist###
if [ "$x" != "$username" ]
then
useradd $username
echo "user added succesfully"
r=$(cat /etc/passwd |grep $username)
echo "user info is equal :>>> $r "
elif [ "$x" = "$username" ]
then
echo "user exist "
else
echo "invalide input "
fi
```

➤ Script edit user



```
GNU nano 5.6.1
#!/bin/bash

echo "enter username you want to edit login name and uid "
read -p "enter user name :>>> " username
read -p "enter new login name :>>> " newname
read -p "enter new uid :>>> " newuid
x=$(grep $username /etc/passwd |cut -d ":" -f 1 )

#if user not exist###
if [ "$x" != "$username" ]
then
echo " user not exist "
elif [ "$x" = "$username" ]
then
#check result
usermod -l $newname -u $newuid $username
echo "user has new login name" $newname " successfully "
echo "user has new uid " $newuid " successfully "
#check result
r=$(cat /etc/passwd |grep $username)
echo "check result :>>> $r"
else
echo "invalide input"
fi

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-] To Bracket
Exit Read File Replace Paste Justify Go To Line M-E Redo M-B Copy M-? Where Was Next
```

➤ Script remove user



```
GNU nano 5.6.1
#!/bin/bash

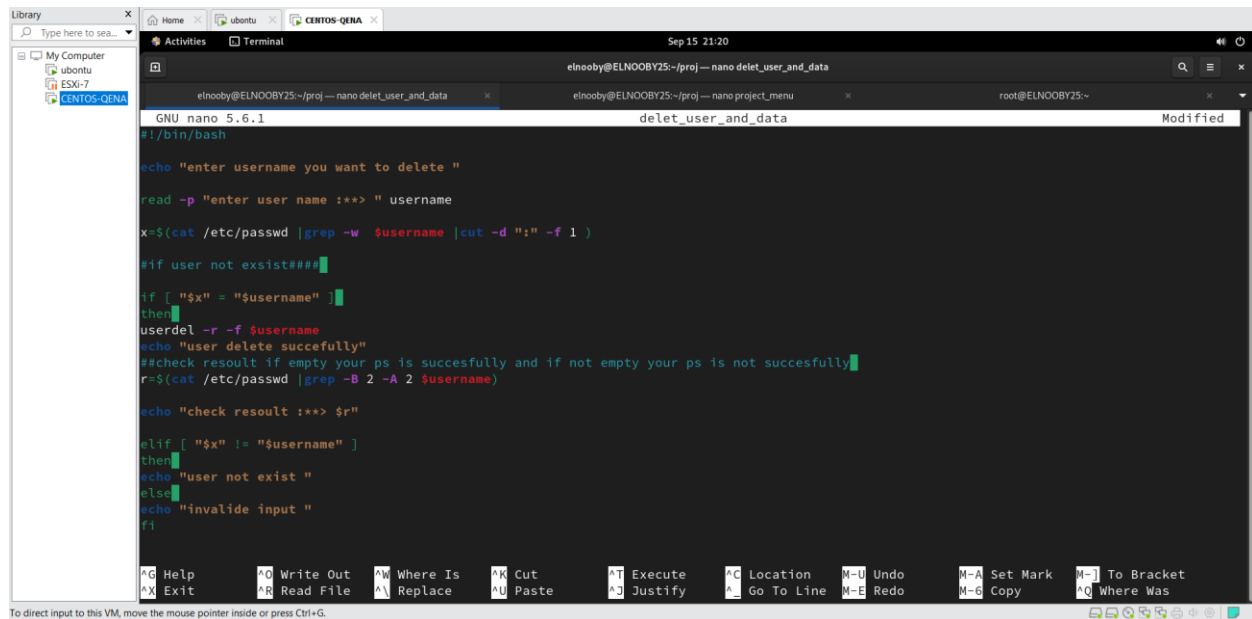
echo "enter username you want to delete "
read -p "enter user name :>>> " username
x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

#if user not exist###
if [ "$x" = "$username" ]
then
userdel $username
echo "user delete succefully"
#check result if empty your ps is succesfully and if not empty your ps is not succesfully
r=$(cat /etc/passwd |grep -B 2 -A 2 $username)

echo "check result :>>> $r"
elif [ "$x" != "$username" ]
then
echo "user not exist "
else
echo "invalide input "
fi

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-] To Bracket
Exit Read File Replace Paste Justify Go To Line M-E Redo M-B Copy M-? Where Was Next
```

➤ Script remove user whit data



The screenshot shows a terminal window with a nano editor open, editing a script named `delet_user_and_data`. The script prompts the user for a username, checks if it exists in the `/etc/passwd` file, and if found, uses `userdel -r` to remove the user and their data. It then checks the result by running `ps` and `grep` on `/etc/passwd`. If the user is not found, it prints "user not exist". If the input is invalid, it prints "invalid input". The terminal window has a title bar with "Sep 15 21:20" and a status bar with various keyboard shortcuts.

```
GNU nano 5.6.1                                delet_user_and_data Modified
#!/bin/bash

echo "enter username you want to delete "

read -p "enter user name :>>> " username

x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

#if user not exist###

if [ "$x" = "$username" ]
then
userdel -r -f $username
echo "user delete succefully"
##check resoult if empty your ps is succesfully and if not empty your ps is not succesfully
r=$(cat /etc/passwd |grep -B 2 -A 2 $username)

echo "check resoult :>>> $r"

elif [ "$x" != "$username" ]
then
echo "user not exist "
else
echo "invalid input "
fi
```

➤ Script disable user



The screenshot shows a terminal window with a nano editor open, editing a script named `disable_user`. The script prompts the user for a username, checks if it exists in the `/etc/passwd` file, and if found, uses `usermod -L` to lock the user. It then prints "user locked succefully". If the user is not found, it prints "user not exist". If the input is invalid, it prints "invalid input". The terminal window has a title bar with "Sep 15 21:44" and a status bar with various keyboard shortcuts.

```
GNU nano 5.6.1                                disable_user Modified
#!/bin/bash

echo "enter username you want to lock "

read -p "enter user name :>>> " username

x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

#if user not exist###

if [ "$x" = "$username" ]
then
usermod -L $username
echo "user locked succefully"

elif [ "$x" != "$username" ]
then
echo "user not exist "
else
echo "invalid input "
fi
```

➤ Script enable user



The screenshot shows a terminal window with the nano 5.6.1 editor open. The file being edited is named 'enable_user'. The script content is as follows:

```
#!/bin/bash

echo "enter username you want to enable "

read -p "enter user name :>>> " username

x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

#if user exist###

if [ "$x" = "$username" ]
then
#if you want take passwd outo enable
#1#pass=$(openssl rand -base64 8 )
echo "enter password "
passwd $username --stdin #2#echo $pass --stdin
echo "user enabled succefully"
#if user not exist
elif [ "$x" != "$username" ]
then
echo "user not exisx "
else
echo "invalide input "
fi
```

The terminal window title bar shows 'Sep 15 21:46' and the current directory is 'elnooby@ELNOOBY25:~/proj — nano enable_user'. The bottom status bar of the nano editor shows various keyboard shortcuts like 'H Help', 'X Exit', 'O Write Out', etc.

➤ Script Add group



The screenshot shows a terminal window with the nano 5.6.1 editor open. The file being edited is named 'addgroup'. The script content is as follows:

```
#!/bin/bash

echo "enter groupname you want to add "

read -p "enter group name :>>> " groupname

x=$(cat /etc/group |grep -w $groupname |cut -d ":" -f 1 )

#if group exist###

if [ "$x" != "$groupname" ]
then
groupadd $groupname
echo "group add succefully"
#check resoult print group name and group id
r=$(cat /etc/group |grep -w $groupname)
echo "check resoult :>>> $r"
elif [ "$x" = "$groupname" ]
then
echo "group is exist "
else
echo "invalide input "
fi
```

The terminal window title bar shows 'Sep 15 21:49' and the current directory is 'elnooby@ELNOOBY25:~/proj — nano addgroup'. The bottom status bar of the nano editor shows various keyboard shortcuts like 'H Help', 'X Exit', 'O Write Out', etc.

➤ Script remove group



The screenshot shows a terminal window with a nano editor editing a script named `removegroup`. The script prompts the user to enter a group name, checks if it exists in `/etc/group`, and then uses `groupdel` to remove it. It also includes a check to ensure the group was successfully removed.

```
GNU nano 5.6.1 removegroup
#!/bin/bash


echo "enter groupname you want to delete "

read -p "enter group name :>>> " groupname

x=$(cat /etc/group |grep -w $groupname |cut -d ":" -f 1 )

if group exist###
then
if [ "$x" = "$groupname" ]
then
groupdel $groupname
echo "group removed succesfully"
#check resoult if check is empty your ps is succesfully
r=$(cat /etc/group |grep -w $groupname)
echo "check resoult :>>> $r"
elif [ "$x" != "$groupname" ]
then
echo "group is exist "
else
echo "invalide input "
fi
fi
```

➤ Script add user to group



The screenshot shows a terminal window with a nano editor editing a script named `add_user_to_group`. The script prompts the user to enter a username and a group name, checks if both exist, and then uses `usermod` to add the user to the group. It also includes a check to ensure the user was successfully added.

```
GNU nano 5.6.1 add_user_to_group Modified
#!/bin/bash

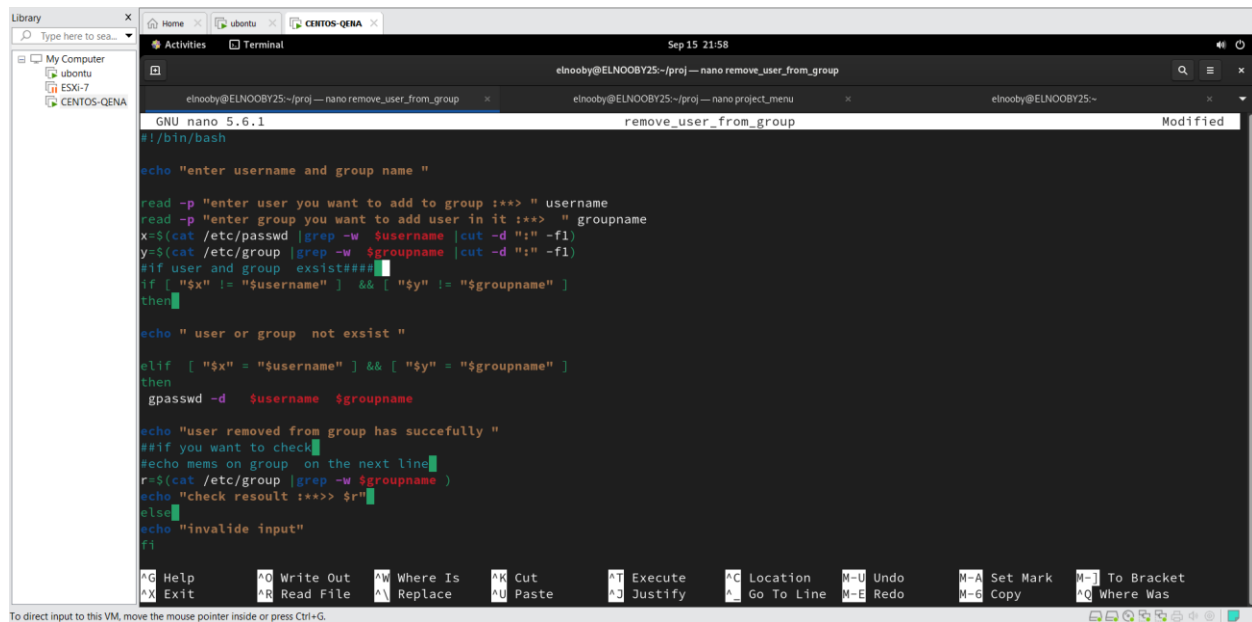
echo "enter username and group name "

read -p "enter user you want to add to group :>>> " username
read -p "enter group you want to add user in it :>>> " groupname
x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1)
y=$(cat /etc/group |grep -w $groupname |cut -d ":" -f 1)

if user and group exist###
if [ "$x" != "$username" ] && [ "$y" != "$groupname" ]
then
echo " user or group not exist "
else
if [ "$x" = "$username" ] && [ "$y" = "$groupname" ]
then
usermod -a -G $groupname $username

echo "user add in group has succesfully "
##if you want to check
#echo mems on group on the next line
r=$(cat /etc/group |grep -w $groupname )
echo "check resoult :>>> $r"
else
echo "invalide input"
fi
fi
```

➤ Script remove user from group



The screenshot shows a terminal window with the nano text editor open. The editor is editing a file named 'remove_user_from_group'. The script content is as follows:

```
GNU nano 5.6.1 remove_user_from_group
#!/bin/bash

echo "enter username and group name "

read -p "enter user you want to add to group :>>> " username
read -p "enter group you want to add user in it :>>> " groupname
x=$(cat /etc/passwd | grep -w $username | cut -d ":" -f1)
y=$(cat /etc/group | grep -w $groupname | cut -d ":" -f1)
if user and group exist###
if [ "$x" != "$username" ] && [ "$y" != "$groupname" ]
then
echo " user or group  not exist "
elif [ "$x" = "$username" ] && [ "$y" = "$groupname" ]
then
gpasswd -d $username $groupname

echo "user removed from group has succesfully "
##if you want to check
echo mems on group  on the next line
r=$(cat /etc/group | grep -w $groupname )
echo "check resoult :>>> $r"
else
echo "invalide input"
fi

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo       ^_ Set Mark   ^] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line ^E Redo       ^G Copy      ^_ Where Was
```

At the bottom of the terminal window, there is a status bar that reads: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

➤ Script list all user



The screenshot shows a terminal window with the nano text editor open. The editor is editing a file named 'list_all_users'. The script content is as follows:

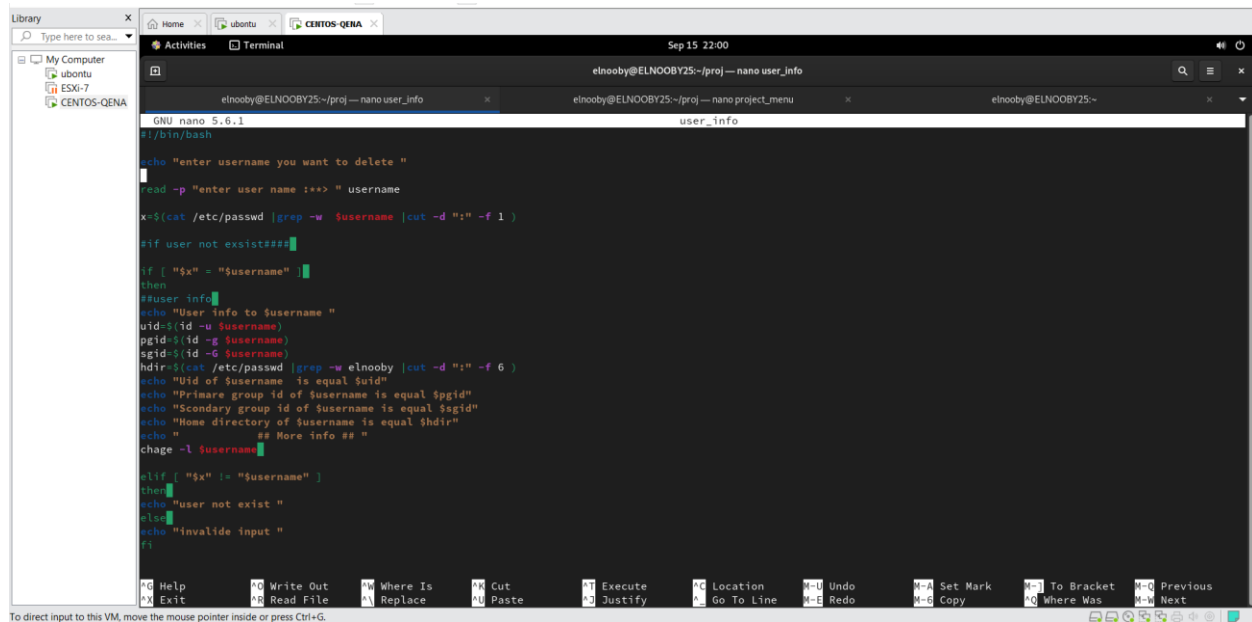
```
GNU nano 5.6.1 list_all_users
#!/bin/bash

##this script to list all users##
list=$(cat /etc/passwd | cut -d ":" -f 1)

echo "list of all users :>>> $list"
```

At the bottom of the terminal window, there is a status bar that reads: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

➤ Script user info



```
GNU nano 5.6.1 user_info
#!/bin/bash

echo "enter username you want to delete "
read -p "enter user name :>>> " username

x=$(cat /etc/passwd |grep -w $username |cut -d ":" -f 1 )

if user not exists==:
then
##user info
echo "User info to $username "
uid=$(id -u $username)
gid=$(id -g $username)
sgid=$(id -G $username)
hdir=$(cat /etc/passwd |grep -w elnooby |cut -d ":" -f 6 )
echo "Uid of $username is equal $uid"
echo "Primare group id of $username is equal $gid"
echo "Secondary group id of $username is equal $sgid"
echo "Home directory of $username is equal $hdir"
echo " ## More info ## "
chage -l $username
elif [ "$x" != "$username" ]
then
echo "user not exist "
else
echo "invalide input "
fi
```

Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was Next

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.