**ABSTRACT**

In order to login in a Linux system we need a username and password. When we go through the directory structure of the Linux, we find different directories *like bin, sbin, etc, lost+found, var, home…etc.* Bin contains the binary files and for any user comment whereas sbin contains super binary files for super user comment. All the configuration files reside in the etc/ including user related configuration, password related configuration. Home/ indicates the home for each user thus we can find the number of user here. When we supply password, it compares with the password stored in */etc/passwd*. If it matches we are able to login. So keep track of all user, Linux uses user identity management because it leads to simplified user and group management and security management becomes easy. For each user there is a unique positive value assigned to it known as user id or UID. Similarly, we have group id to refer group names. Single user can be a part of multiple group which provides the flexibility of sharing files. In this project we are going to implement user identity management on Linux platform including user addition, user information, deletion of user, adding of user to a group and revoking a user.

Keywords

User management, group management, Linux.

**TABLE OF CONTENTS**

**S.No. Contents**

1. **Introduction** 
   1. History
   2. Requirement Analysis
   3. Objective
2. **System Analysis** 
   1. Routing
   2. Routing Techniques
   3. Path Selection
   4. Routing Algorithm
3. **Design** 
   1. UML Diagrams for CPU Scheduling

3.1.1. Activity Diagram

3.1.2. Use Case Diagram

3.1.3. Flow Diagram Diagram

1. **Implementation** 
   1. Configuring a Router

4.1.1. Steps for configuring a router

4.1.2. Screenshot

4.2 Static Routing

4.2.1 Steps for static routing

4.2.2 Screenshot

4.3 RIP (Routing Information Protocol)

4.3.1 Steps for RIP implementation

4.3.2 Screenshot

4.4 OSPF (Open Shortest Path First)

4.3.1 Steps for OSPF implementation

4.3.2 Screenshot

4.5 Migration from IPv4 to IPv6 using tunneling

1. **Limitations and Future Enhancements**
2. **Conclusion**
3. **References**

**LIST OF FIGURES**

**S.No. Figure Page No**

1. **Chapter 3**

Fig. 3.1 Activity Diagram

Fig. 3.2 Use Case Diagram

Fig. 3.3 Data Flow Diagram

1. **Chapter 4**

Fig. 4.1 Screenshot for configuring a Router

Fig. 4.2 Static Routing

Fig. 4.3 Screenshot Static Routing

Fig. 4.4 RIP

Fig. 4.5 Screenshot RIP

Fig. 4.6 RIP Ping Screenshot

Fig. 4.7 OSPF

Fig. 4.8 Screenshot OSPF

Fig. 4.9 Migration Cross over

Fig. 4.10 IP Address Table

Fig. 4.11 Screenshot Cross over

Fig. 4.12 IP Codes

Fig. 4.13 Router Ping Codes

Fig. 4.14 Ping Output Failure

Fig. 4.15 Ping Output Success

1. **INTRODUCTION (Chapter 1)**

**1.1 History**

Linux started around 1991 all because the Finnish student Linus Torvalds wasn’t too happy with Minix, the educational version of the UNIX operating system that he had to work with at the University of Helsinki. In particular, the ability of the kernel (which is the heart of the operating system) of this Minix distribution didn’t please him much. He decided to create a better kernel and gave it the name Linux. Possibly the smartest thing that Torvalds did when starting his initiative was deciding not to do it alone. To find other people who wanted to work with him, he posted a message on Usenet, a major social media platform in those days that could be used to exchange information with other people and get help from other people.

The years between 1993 and 1998 marked the rise of the Linux operating system. One of the most important reasons that it took off so rapidly, is that it provided a very affordable alternative for the expensive UNIX operating system that was used on many mission-critical server systems. In the early days of Linux, no support was available, but that didn’t prevent scientists at different institutes around the globe to start working with it. Linux also acquired huge popularity rapidly in educational environments. Due to this popularity, during this period the most important Linux distributions were created.

**1.2. Requirement analysis**

Before starting to work on your Linux computer, you need to tell it who you are. The person that installed your Linux version has created a user account, and you must use this account and the associated password to make yourself known. To do this, Linux offers you a login prompt. This can be either a graphical or a non-graphical prompt. If you are working on a Linux desktop, you are likely to see a graphical environment.

To log in to a Linux machine, you need a user account name and a password. You should already know what username to use if you installed the machine yourself. If someone else installed the machine for you, ask him or her what username you should use. This username will also have a password. At the login prompt, you need to provide the username and password to make yourself known to the machine. This procedure is also known as *authentication*. When authenticating for the first time, you have to decide what user account to use. There are mainly 2 types of user accounts- Normal account and Root account. Normal account does not have all the permissions for system administration whereas Root account have that permissions.

**1.3.Objective**

1. To simplified user and group management.
2. security management becomes easy.

.

# 2. Line Tasks (Chapter 2)

## password

Every user need password to login. So minimum requirement for a good password are: -

* At least 6 characters
* Use complexity- combination of upper and lower case letters, numbers and special symbols.

**Syntax:-**

Just use ‘***passwd***’ command to change your own password.

Note:-

If you are logged in as root. You can change the password for other users too without knowing their old password.

## Becoming Another user:

As we read earlier, there are 2 types of users- normal user and root user. Root user is used for system administration. you can accidently remove everything using root account. So linux offers a way to substitute any user any time by ***‘su’*** command.

Syntax:-

**su user\_name**

## user information

various commands to obtain user information are:-

* users- shortlist all users.
* who- detailed information about users are doing.
* w- what a user is doing
* finger- more detailed information about the user.
* lastlog- information about last login time for all users.

## User communication

## There are 2 options available to communicate among users.

## write- individual user communication.

## wall- all user communication.

## Note:-

## At the end of the session, both parties that are involved have to use the *Ctrl+C* key sequence to terminate the session. This will bring them back to their prompts, where they can continue their normal work.

1. **Linux file system**

In this section, you’ll learn what default directories exist and what kinds of files you’ll find in these directories. A simple description of the UNIX system, also applicable to Linux, is this:

***"On a UNIX system, everything is a file; if something is not a file, it is a process."***

This statement is true because there are special files that are more than just files (named pipes and sockets, for instance), but to keep things simple, saying that everything is a file is an acceptable generalization. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. Input and output devices, and generally all devices, are considered to be files, according to the system. Various directories available for the root account are: -

| **Directory** | **Content** |
| --- | --- |
| /bin | Common programs, shared by the system, the system administrator and the users. |
| /boot | The startup files and the kernel. |
| /dev | Contains references to all the CPU peripheral hardware, which are represented as files with special properties. |
| /etc | Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows |
| /home | Home directories of the common users. |
| /initrd | (on some distributions) Information for booting. Do not remove! |
| /lib | Library files, includes files for all kinds of programs needed by the system and the users. |
| /lost+found | Every partition has a lost+found in its upper directory. Files that were saved during failures are here. |
| /misc | For miscellaneous purposes. |
| /mnt | Standard mount point for external file systems, e.g. a CD-ROM or a digital camera. |
| /net | Standard mount point for entire remote file systems |
| /opt | Typically contains extra and third party software. |
| /proc | A virtual file system containing information about system resources. More information about the meaning of the files in proc is obtained by entering the command **man *proc*** in a terminal window. The file proc.txt discusses the virtual file system in detail. |
| /root | The administrative user's home directory. Mind the difference between /, the root directory and /root, the home directory of the root user. |
| /sbin | Programs for use by the system and the system administrator. |
| /tmp | Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work! |
| /usr | Programs, libraries, documentation etc. for all user-related programs. |
| /var | Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet, or to keep an image of a CD before burning it. |

# User management

Before starting to create users, it makes sense to know about the different properties that Linux userstypically have. These properties are stored in the /etc/passwd and /etc/shadow files. Based on this knowledge, you’ll be better able to create the user according to your specific needs. When creating a user, you need to provide a value for the following properties.

* Username
* Password
* User ID (UID)
* ID of the primary group of the user
* Comment field
* Home directory
* Default shell

## Username: -

To login in a Linux system, every user needs username to authenticate their identity. Username is unique in nature and it may be combination of letters and numbers.

Basically there are 2 type of user account: -

* Root user
* Non root user

Root user have special permissions and by using this account you can accidently destroy everything on your system. So it is a good idea not to use root user much.

## Password: -

## This is the key used to authenticate user. Every user should have strong password. Minimum requirement for a strong password is:

* At least 6 character
* Use complexity- combination of uppercase-lowercase letters, numbers and special characters.

Password and related setting are stored in */etc/shadow* file.

## UID:-

## It acts as the identifier of the user for your system. Username is just a convenient way for humans. Every user has unique ID known as UID. In default Linux configuration, 16 bits are available for creating UID.i.e. maximum number of UID available is approx. 60000. UID 0 is reserved for Root user and UID below than 1000 are reserved for system accounts. UID and GID related information are stored in */etc/log.defs*.

## Group Membership: -

## There are 2 types of groups-

* Primary group
* Secondary group

On Linux, all user must be a member of at least one group. Files of primary group are stored in */etc/passwd* and files from secondary group are stored in */etc/group.*

## Gecos field:-

It stands for “General Electric Comprehensive Operating System”. It is used to include some comment to make it easier to identify user.

**“finger”** command is used to display the content of Gecos field.

## Home Directive: -

Most user have home directive where users store their files. It resides in */home*. When a user account is created, the content of */etc/skel* directory is copied to the users’ home directory.

## Shell:-

It is the environment where the user types the command which is needed to execute. Any user who needs to login in Linux system needs a shell.

# User management commands

There are mainly 4 user management commands in linux namely,

* useradd
* passwd
* usermod
* userdel

## **useradd**

## This command is used to create a new user account.

## *useradd -D*

## 

Group: -

So there is a GROUP. Every user must be member of at least one group.

Mainly there are 2 types of group: -

* primary group and
* secondary group.

Every group has its own id called gid. We will read about it later.

Note: All user gets their own private groups as primary groups and this primary group has the same name as the user. A user can be a member of more than just the primary groups and they automatically inherit the permissions granted to these other groups.

**Home: -**

When a user is created, a home directory is created too. The contents of /etc/skel directory are copied to the home directory. Here the users store files.

**Inactive: -**

When a user is created, it does not mean it is active. It remains inactive until a password is not set for that user.

Note: -

**inactive=1** the number of days after a password expires until the account is permanently disabled.

**Expire:-**

It is mainly done for expiration period- the date on which the user account will be disabled.

**Shell:-**

It is the environment where the user type commands that needs to be execute on the computer. Any users who needs to login need a shell.

***Skel:-***

It signifies the directory containing “skeletal” user files. In other word, files such as sample.

Note:-

When a user account is created, the contents of the /etc/skel directory are copied to the user’s home directory.

**Create\_mail\_spool:-**

It is the directory where the mails of the user accounts are stored.

## Syntax

## useradd –[options] login\_name

## Options in useradd:-

So now we will look for the different options available in useradd: -

* -b base directory if home directory is not specified
* -c comment or description about user account
* -d to define a new home directory
* -D changing the default value
* -e expiration time of the user account
* -f number of days after which password will be expired
* -g change the gid or group name of primary group
* -G list of supplementary groups
* -h help
* -k change the /etc/skel directory
* -K overrides the default values
* -l do not add user to last log database
* -m create home directory
* -M do not create user’s home directory
* -N do not create the same name as the user.
* -o create user account with duplicate UID
* -p password of the user account
* -r create a system account
* -R apply changes in the CHROOT\_DIR directory
* -s name of user’s login shell
* -u numerical value of the user’s id
* -U create a group with same name as the user
* -z SELinux user for the user’s login.

## Create user account

Suppose I have to create a user account with :-

## *Conditions:-*

* Login name: ibm
* Full name: internship project
* Group user\_management with gid 1223
* Secondary group: group\_managemnet, linux;
* User id: 2132
* Home : /home/intern/user
* Password: ibm@123
* Expiry date: 24/09/2017
* Inactivity condition: 28days
* Shell: /bin/bash

## Do not add user to last log database and make it a system account.

***Basic Requirements:-***

1. Create 3 groups:-

* User\_management
* Group\_management
* Linux



1. New directory in /home with name of /intern/user



**Commands**

****

**Verification**

****

****

**To find the group information: -**

****

**Note:-** Encrypted password resides in shadow file. /etc/shadow.

## passwd:-

## This command is used for password modification of any existing user.

## If you create a new user

## 

## *It is not active until you provide a password to it.*

## Verification

## To check to the status of passwd, type:-

## 

## And its output will be:

## 

* Nitin: login name.
* L: locked.
* 06/25/2016: last password change date.
* 0: minimum days required to change password.
* 99999: maximum days required to change password.
* 7: no of days before the password expiration warning (warn).
* -1: no of days to wait after password expiration to disable account (inactive).

**Attributes of passwd :-**

Password files are stored in shadow file i.e. in /etc/shadow

****

And its output will be a list of users and their info. Let us take one of them

****

* Nitin:- login name
* !:- encrypted password
* 16977:- no of days of last password change after unix generation (January 1, 1970)
* 0: minimum days required to change the password
* 99999: maximum days required to change the password
* 7: number of days to warn user about password expiration
* And the last field is empty i.e. reserved for the future use.

**So Major attributes are:-**

* Password status
* Minimum days required to change password
* Maximum days required to change password
* Number of days to warn user about password expiration
* Inactivity of user account

**Syntax:-**

**passwd –[options] login**

****

Again check the status of password for user nitin



It is changed from L to P.

**Options in passwd:-**

* -a: report password status of all user
* -d: remove password
* -e: force expiration of password
* -h: help
* -k: inactive
* -l: lock password
* -n: minimum number of days before password can be changed
* -q: quiet mode
* -r: change password in REPOSITORY repository
* -S: status of account
* -u: unlock password
* -w: set expiration warning days
* -x: maximum number of days before password can be changed.
* **passwd commands: -**

let us consider a situation: -

* New password: - negi@123
* Lock nitin
* Minimum days to change password= 20
* Maximum days to change password= 300
* Warning date of password expiration=10
* Inactivity condition= 28

****

****

**Verification: -**

****

Now unlock this account

****

## Usermod

## This command is used to modify user details

## Syntax: -

## usermod –[options] login

**Options available in usermod:-**

* -c full name
* -e expiry date of account
* -f inactive condition
* -g user’s default group
* -G specifies additional group
* -I change login name
* -L lock account
* -p password
* -m home directory
* -s default shell
* -u UID
* -U unlock user account

**usermod commands:-**

Let us modify the user nitin:-

* Full name: Nitin negi
* Password: nitin@123
* Expiry date: 2016-08-24
* Inactive condition: 28 days
* New login name: negi
* Home: /home/negi
* Default shell: /bin/sh
* UID : 1245

## 

Default group:-project

## Additional groups: user\_management, group\_management

## 

## Verification:-

## 

## 

## 

## Now lock the user and then unlock it: -

## 

## userdel

## This command is used to delete any user account.

## Syntax:-

## userdel –[options] login

**Options in userdel:-**

* -f: force removal of files
* -h: help
* -r: remove home directory and mail spool
* -R: directory to chroot into
* -Z: remove any SELinux user mapping for user.

## Commands of userdel

## Delete user negi and their all files

## 

# Group Management

In Linux, every user must be a member of a group so now we will discuss group management on Linux.

**Properties of Group:-**

* Every user must be a member of a group.
* All user gets their own private group as primary group and this primary group has the same name as the user.
* A user can be member of more than just the primary group and will automatically inherit the permission granted to these other supplementary

**Commands for Group Management: -**

Like user management, Group management also have some commands like-

* *groupadd*
* *groupmod*
* *groupdel*

**groupadd**

**syntax:-**

**groupadd group\_name**

****

and to view the status of group

**cat /etc/group**

****

Output will be like this

****

here the first field.i.e. intern signifies groupname. Second field contains the encrypted password stored in /etc/gshadow. Third field is the unique group id and the last part contains the names of the members of the group.i.e. User list. In this there is no member in the group.

So lets start with adding users to a group. This can be done using usermod command. but before that we will look for the options available in groupadd.

* -f : force
* -g: GID
* -h: help
* -k: override default values
* -o: non unique
* -p: password
* -r: system account
* -R: root

e.g.



**output**



now we will add some user to this group:-

we can do this using ***usermod*** command by modifying the user account.



Now check whether user1 is added to group4 or not by cat /etc/group



we may assign a user to different groups too.



Output will be like: -



in it you may see that ravi is in group1, group2, group3 and group4.

**Groupmod:**

it is used to modify groups. The various options available are:-

• -p: password

• -n: new name

• -o: non unique

• -R: root

• -g: gid

• -h: help

e.g.



to check its effect type cat /etc/group



and its output will be:-



**groupdel:-**

This command delete the group.

# Shell Script Implementation:-

Source Code:-

#! /bin/bash

# #! is called SHE-BANG which denotes the path of the interpretor or shell

while true

do

printf "Welcome to user management control panel\n"

#printing the selection menu for user management control panel

printf "1. User Addition \n"

printf "2. Password Management\n"

printf "3. User Modification\n"

printf "4. User Deletion\n"

printf "5. Exit \n"

printf "Enter your option: \t"

#taking the selection option.

read choice

#switch case is used for the applied options

case $choice in

'1')

#user Addition section

printf " ----Welcome to User Addition Section---- \n"

#taking the information for the user addition

printf " Enter your Login name :\t"

read p

printf " Enter full name : \t"

read c

#shell is used for execution of command in UNIX

printf " ---Enter your shell option---: \n"

# printing the various shell options

printf "1. Bourne shell \n" # it is sh shell which is available in every UNIX

printf "2. C shell \n" # syntax are based on C

printf "3. Bash shell \n" # it is bourne again shell.

printf "4. tcsh\n" # it is also a cshell but with advanced features

while true #it is the loop for shell option

do

read choice

#taking the options for the available shell

case $choice in

'1')

s="/bin/sh"

break #if any of these 4 value is matched,it will come out of of the shell loop

;;

'2')

s="/bin/csh"

break

;;

'3')

s="/bin/bash"

break

;;

'4')

s= "/bin/tcsh"

break

;;

\*)

# it is the deafault option. if user press other than 1,2,3, or 4 it will execute

printf "Invalid choice entered \n"

printf "Enter 1 2 3 or 4 \n"

continue

#continue is a loop control statement which is responsible for the current iteration of loop. exit from shell switch loop option and execute next commands

;;

esac #switch case for shell is closed here

done #exit from shell while loop

printf "Enter your password: \t" $pass

read pass

printf "Enter your primary group \t"

read g

# now check group id for that user

if id -g $g >/dev/null 2>&1; then

# /dev/null is a special file which is used to discard the output.

# /dev/null 2>&1 is used to discard the output as well as error message.

# the command will execute but it will not shown

# 1 is STDOUT (standard output) 2 is STDERR (standard error).

# this concept is in IO redirection of UNIX.

printf "Group exists- Do not need to add group \n"

else

#if that id for the group exist, there will no need for creation of new group.

# if that id for the group does not exist, we have to create a new group.

groupadd $g

fi

# as we did for the primary group we have to do same for secondary group as well

printf "Enter secondary group \t"

read g1

if id -g $g1 >/dev/null 2<&1; then

printf "Group exists- Do not need to add group \n"

else

#groupadd is the command used to create a group

groupadd $g1

fi

printf "Enter the inactivity time \t"

read d

printf "Enter the expiry date for the user(Date-Month-year) \t"

read e

# After taking all the information, ask for confirmation

printf "Please confirm [y/n] \t" $x

read x

if [ "$x" = "y" ]; then

#if the user is confirmed he/she has to press y to proceed

if id -u $p >/dev/null 2>&1; then

#it is the same concept of IO reirection of UNIX. but here the user id for the new user is checked. if that id exists

printf "User already exists"

$p=""

#if that id for new user do not exist , create a new user with applied information

else

useradd -c "$c" -s "$s" -m -d "/home/$p" -p "$pass" -g "$g" -G $g1 -f "$d" -e "$e" $p

printf "new user created with \nlogin name $p \nfull name $c \nshell \t $s \n home directory /home/$p \n password $pass \n Primary Group $g \n Secondary Group $g1 \n Inactivity time $d \n Expiry date $e \n"

fi

# if the user is not confirmed and type anything except y , it will exit from the loop and print the main menu again and print failed

else

printf "Failed to create user \n"

continue #loop to print the main menu again

fi

continue #exit from the case 1 but to execute the next commands .i.e. again want to user management option. thats why this continue is here.

break; # it will responsible to exit from the case 1.

;;

'2')

# it is the password management section

# display the selection menu

while true

do

printf "Welcome to Password Management Section \n"

printf " 1. to check the status of user\n"

printf " 2. reset password \n"

printf " 3. Modify password attributes\n"

printf " 4. Exit \n"

# input the selected option

read choice

# use the switch case to execute

case $choice in

'1')

printf "Enter the username \t"

read p

# status of user passwd -S login\_name

passwd -S $p

continue #loop the password managent loop

;;

'2')

printf "Enter the username \t"

read p

# Reset password .i.e. new password set

passwd $p

continue #loop the password management loop

;;

'3') while true

do

printf "welcome to password attributes modification \n"

# display the options available .i.e. menu of password modification

printf " 1. set minimum days to change password\n"

printf " 2. set maximum days to change password\n"

printf " 3. warning date of password expiration\n"

printf " 4. Inactivity condition for user\n"

printf " 5. Exit \n"

# take the options for password modification

read choice

# use of nested switch case

case $choice in

'1')

printf "Enter the username \t"

read p

printf "Enter the number of minimum days\t"

read n

passwd -n "$n" $p

continue

;;

'2') printf "Enter the username \t"

read p

printf "Enter the number of maximum days\t"

read x

passwd -x "$x" $p

continue

;;

'3') printf "Enter the username \t"

read p

printf "Enter number of warning date\t"

read w

passwd -w "$w" $p

continue

;;

'4') printf "Enter the username \t"

read p

printf "Inactivity condition\t"

read i

passwd -i "$i" $p

continue

;;

'5')

# ----Exit option----

printf "Are you sure? \n"

printf "Please confirm [y/n]" $x

read x

if [ "$x" = "y" ]; then

break #it will exit from the password menu and comes at main menu

else

printf "Enter your values again\n"

fi

continue #start the loop again

;;

\*) #default option for password modification

printf "Invalid choice entered\n"

printf "Enter 1 2 3 or 4\n"

continue # exit from password modification and execute next commands

;;

esac # close inner switch case.i.e. loop for passwor modication

done

;;

'4')

# ----Exit option----

printf "Are you sure? \n"

printf "Please confirm [y/n]" $x

read x

if [ "$x" = "y" ]; then

break #it will exit from the password menu and comes at main menu

else

printf "Enter your values again\n"

fi

continue # exit from case 4

;;

\*) #default option for password management

printf "Invalid choice entered\n"

printf "Enter 1 2 or 3\n"

continue # exit from password managent and execute next commands

;;

esac # close outer switch case .i.e. for password management

done # close outer while loop

continue

break #exit from the case 2

;;

'3')

# ----- user modification section ----

# read the information which needs to be modified

printf "Welcome to user Modification Section\n"

printf " Enter Login name which you want to modify :\t"

read p

printf " Enter your new Login name :\t"

read p1

printf " Enter new full name : \t"

read c

printf " ---Enter new your shell---: \n"

printf "1. Bourne shell \n"

printf "2. C shell \n"

printf "3. Bash shell \n"

printf "4. tcsh\n"

while true

do

read choice

#taking the options for the available shell

case $choice in

'1')

s="/bin/sh"

break

;;

'2')

s="/bin/csh"

break

;;

'3')

s="/bin/bash"

break

;;

'4')

s= "/bin/tcsh"

break

;;

\*)

# it is the deafault option. if user press other than 1,2,3, or 4 it will execute

printf "Invalid choice entered \n"

printf "Enter 1 2 3 or 4 \n"

continue

;;

esac #switch case for shell is closed here

done #end of shell while loop

printf "Enter your password: \t" $pass

read pass

printf "Enter your primary group \t"

read g

# see user additon section for more details- io redirection

if id -g $g >/dev/null 2>&1; then

printf "Group exists- Do not need to add group"

else

groupadd $g

fi

printf "Enter secondary group \t"

read g1

if id -g $g1 >/dev/null 2<&1; then

printf "Group exists- Do not need to add group \n"

else

#groupadd is the command used to create a group

groupadd $g1

fi

printf "Enter new inactivity time \t"

read f

printf "Enter new expiry date for the user(Date-Month-year) \t"

read e

# after reading all the information,ask for confirmation

printf "Please confirm [y/n] \t" $x

read x

if [ "$x" = "y" ]; then #starting of confirmation condition

#if user type y then modifaction will be done

usermod -l "$p1" -c "$c" -s "$s" -p "$pass" -g "$g" -G "$g1" -f "$f" -e "$e" $p

#display the modified information

printf " user information modified with new\n login name\t $p1 \n full name\t $c \n shell\t $s \n primary group\t $g \n secondary group $g1 \n password\t $pass \n inactivity time\t $f \n expiry date \t $e \n "

else

#if user is not confirmed then modification will be aborted

printf " modification can not be done \n"

fi #end of confirmation condition

continue

break # exit from case 3

;;

'4')

while true

do

#---- user deletion section ----

printf "welcome to user deletion section\n"

# display selection options available in user deletion section

printf "1. Force removal of file \n"

printf "2. Removal of home directory \n"

printf "3. Exit \n"

# read the selection option

read choice

#use switch case to execute

case $choice in

'1')

printf "Enter login name\t"

read p

#ask for confirmation about the force removal

printf "Are you sure? \n"

printf "Please confirm [y/n]" $x

read x

if [ "$x" = "y" ]; then #confirmation condition started

userdel -f $p

printf "user successfully deleted\n"

else

continue #if user type y then user will be deleted else it will exit

fi #end of confirmation condition

continue

;;

'2')

printf "Enter login name\t"

read p

printf "Are you sure? \n"

printf "Please confirm [y/n]" $x

read x

if [ $x == "y" ]; then

userdel -r $p

printf "user successfully deleted\n"

else

continue

fi

continue

;;

'3')

printf "Are you sure? \n"

printf "Please confirm [y/n]" $x

read x

if [ "$x" = "y" ]; then

break #it will exit from the password menu and comes at main menu

else

printf "Enter your values again\n"

fi

continue # exit from user deletion menu

;;

\*) #default option for user deletion section

printf "Invalid choice entered\n"

printf "Enter 1 or 2\n"

continue #exit from user deletion section and execute left commands

;;

esac # close switch case for user deletion section

done

continue

break #exit from case 4

;;

'5')

# ----Exit option----

printf "Are you sure? \n"

printf "Please confirm [y/n] \t" $x

read x

if [ "$x" = "y" ]; then

exit 0

else

printf "Enter your values again\n"

continue

fi

break # exit from case 5

;;

\*) # default option for user management section

printf "Invalid choice entered\n"

printf "Enter 1 2 3 or 4\n"

continue #exit from this switch case and start while loop again

;;

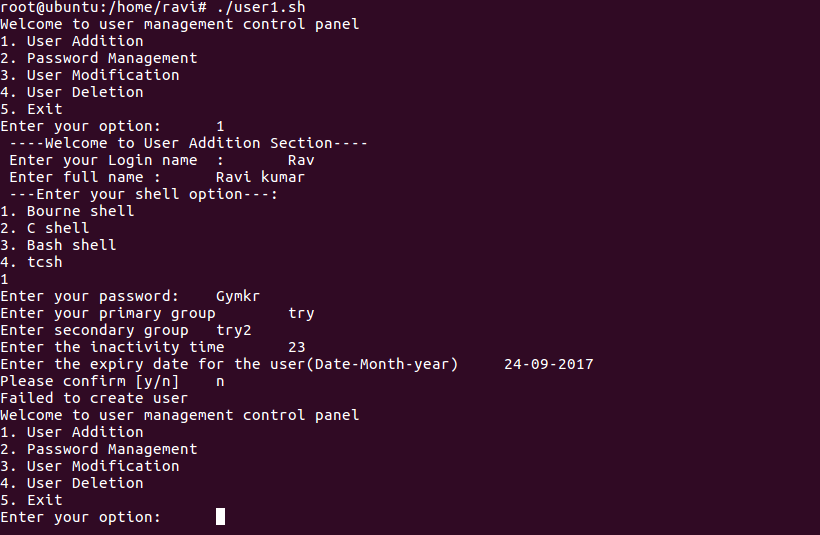
esac # end of outer switch case .i.e. for the main menu

done #end of while loop

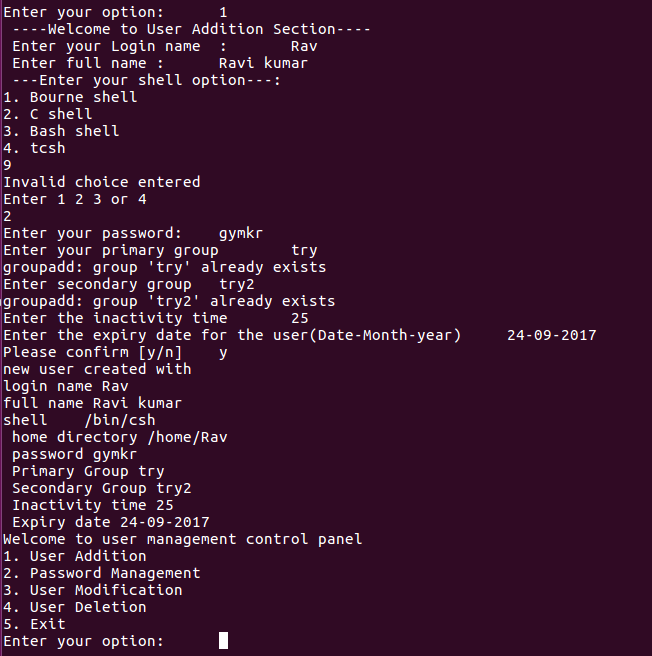
**output screens: - shell script**

1. **Useradd**

First we will add a new user. So choose option 1. Fill the required information. If you are not confirmed then type n, then user will not be added as program will exit and come back on the main menu of user management.



Again go to option 1 to add user, fill the required info and if you are confirmed about the details, type y. A new user will be added and it will come back to main menu of user management.



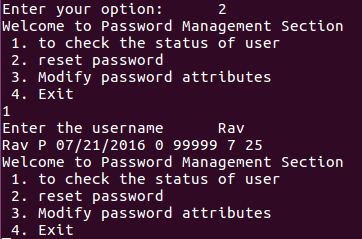
1. **Passwd**

when you will type 2 , you will enter in the password management section. Here there will be different options.

### **Status :-**

To check the status of user, type 1

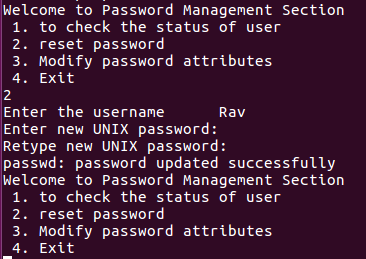
It will show you the status of the user and come back in the password management section.



### **Reset password**

To reset password, type 2

Again it will come back to password section after executing the command.

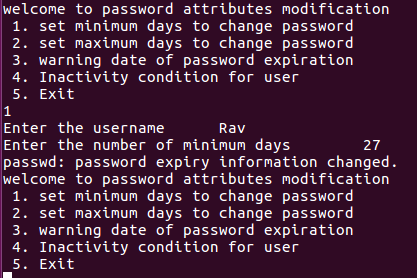


### **Password attributes management**

This comes under option 3 for password section. when you will type 3, you will get different options like change minimum, maximum, warning, and inactivity days.

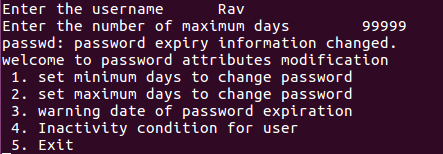
#### **Minimum days :-**

In this section we can modify the minimum days by going through option 1. After execution it will return back in password attribute modification section for modification of other attributes.

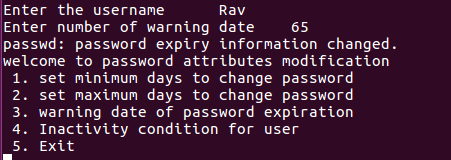


#### **Maximum days**

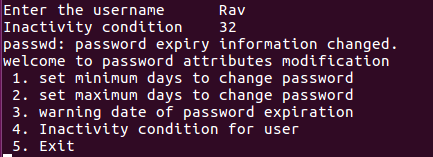
Similary the maximum days for password change is modified.



#### **Warning date**

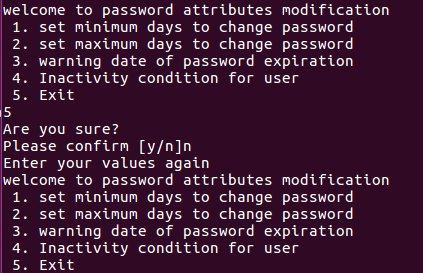


#### **Inactivity condition**

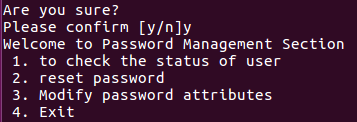


#### Exit- password attribute

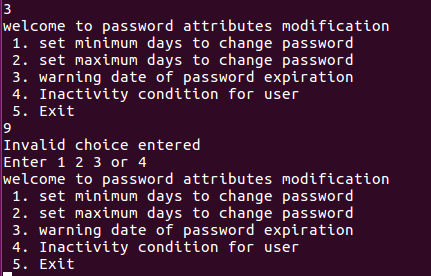
If you are not sure to exit, it will remain in password attribute modification section



If you are sure to exit, it will come to password management section.

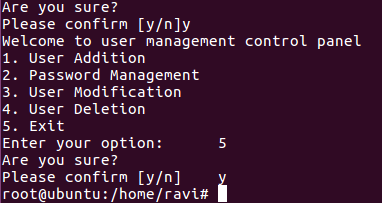


If any other option is entered except 1,2,3,4 and 5. Then it will remain in the password attribute section.

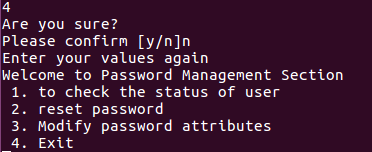


### **Exit- Password management section**

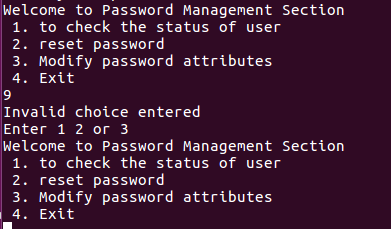
If you are sure to exit, then it will exit and go to main menu



And if you are not sure to exit, it will remain in the password management section.

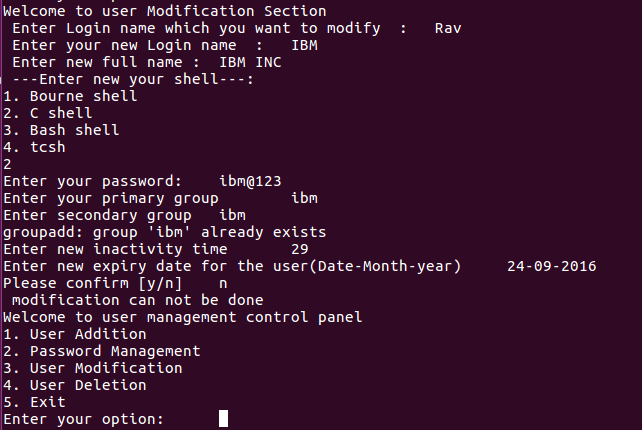


If any other option is entered except 1,2,3 or 4 . then it will remain in the password management section.

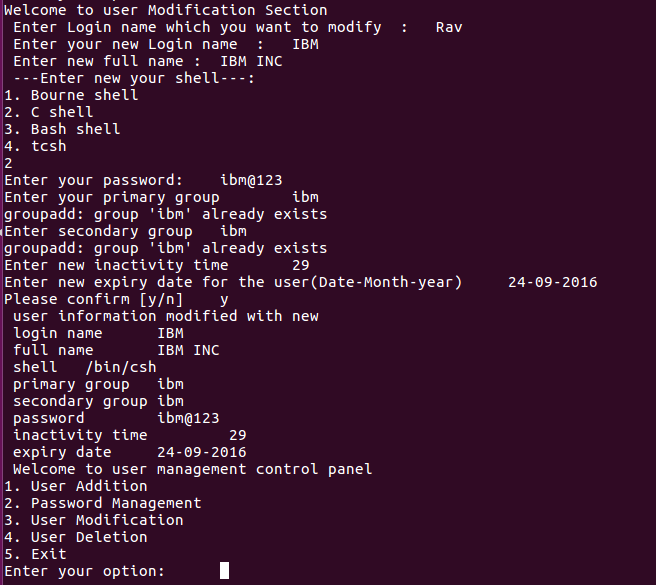


1. **usermod**

It comes under option 3. fill the required information and if you are not sure about the given details then modification will not be done and it will return main menu.



And if you sure about the detail, type y when it asks you for confirmation. Medication will be done and it will return the main menu.

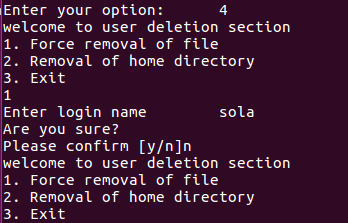


1. **userdel**

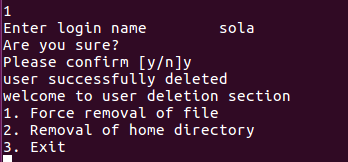
This option is under 4 and comes with mainly 2 options.

### Force removal of files:-

### Enter 1 then it will ask you for login name, Enter it , then program will ask for confirmation. If you are not confirmed about it, type n. it will not delete that user but return to the user deletion section.

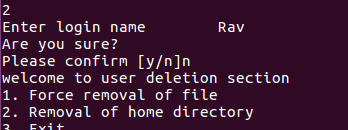


And if you are confirmed about it, type y, user will be deleted and it will return the user deletion section.



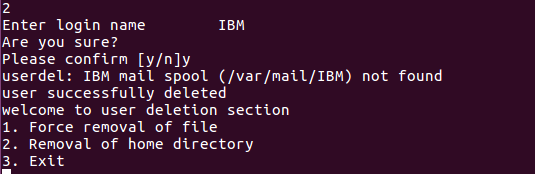
### **Removal of home directory: --**

This is under option number 2 for user deletion section. If you are not sure to delete that given user. It will return back the user deletion menu without deleting that user.



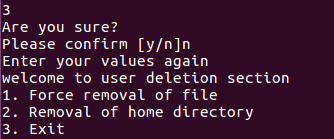
And if you sure to delete that user, type y. it will delete the home

And if you sure to delete that user, type y. it will delete the home directory and return back to user deletion menu.

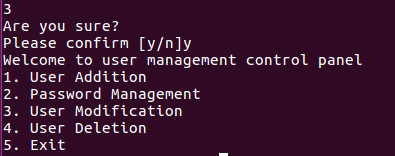


### **Exit- user deletion section**

If you are not sure to exit then type n , it will return back the user deletion section.

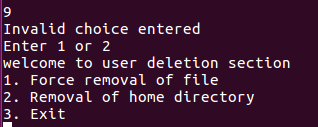


And if you will type y means you are sure to exit then it will return the main menu.



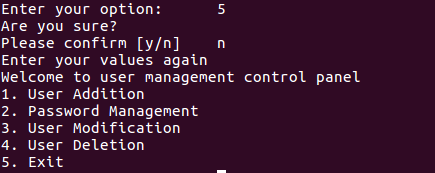
### **Any other option except 1,2,3**

It will return the user deletion menu

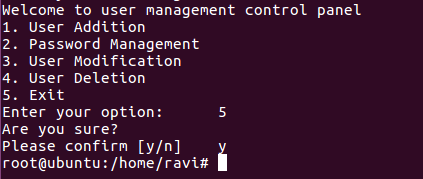


## Exit from main menu

If not sure, main menu is returned.

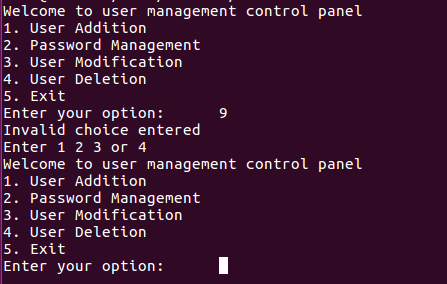


If sure, program will terminate.



## Any other option except 1,2,3,4,5 in main menu

It will return the main menu by stating that- invalid option.



**6. References (Chapter 6)**

* [**www.wikipedia.com**](http://www.wikipedia.com)
* ABRAHAM, JOSHUA ISD, JOSHUA BLOCH, JOSHUA FINKELSTEIN, JOSHUA KYLE, AND JOSHUA PONDEXTER., 2009. “How TCP/IP works”. New Books Title Author Pub Edition Accounting. [CD-Rom]: a lecture-based presentation on the inadequacy of the ecumenical perspectives.

Website, “http://www.docstoc.com/docs/72134291/Investment-Banking-Joshua-Pearl”.

* AMNON BARAK, ILIA GILDERMAN, AND IGOR METRIK., 1999. Performance of the communication layers of TCP/IP with the Myrinet gigabit LAN. Journal: Computer Communications - COMCOM , vol. 22, no. 11, pp. 989-997.ANTHONY MCAULEY, SOLIMAN, AND RAMJEE., 2002. Title “IDMP-Based Fast Handoffs and Paging in IP-Based 4G Mobile Networks”, Magazine A sub- IEEE Communications •140. Archan Misra, IBM T. J. Watson Research Center, University of Texas at Arlington.
* JIAN-MIN, AND INDRA, 1998. Title “Congestion detection in ATM networks”. Published in *Volume 34*, Issue *3*.