Users and Groups

Types of Users:

- In a multiuser OS environment, there are multiple users who log into the system to carry out their tasks. The users in multiuser OS environment can be categorized as-

End users-

- These work on various utilities or applications installed on the system. These utilities can be single line utilities, shell scripts and C/C++ programs.
- The End users of the system are not involved in system maintenance task.

Power users-

- The power users of the system carry out all the activities of an end user and are also responsible for maintaining printer queues, backing up files on regular basis and performing house keeping tasks like cleaning up unwanted files.

System Administrators-

- System administrators have complete control over the manner in which the OS is set up on the machine and all the applications installed on the system. They are also responsible for maintaining the hardware devices attached to the system.

The Function of the Unix Administrator

- The task of the Unix Administrator is difficult to define. Broadly it involves ensuring that the Unix system provides the services it was intended for.
- These tasks vary from situation to situation, but will usually include the following.

- Adding and removing users.
- Adding and removing hardware.
- Adding and removing software.
- Performing backups.
- Monitoring the system to ensure correct operation.
- Troubleshooting.
- Documentation.
- Auditing security.
- Helping users.

The control of users and groups exists at the core of Red Hat Linux system administration. Users can be either people or logical users (accounts that exist for applications so that they can perform specific tasks). Both types of users have a Unique User ID and Group ID.

Groups are logical expressions of organization. Groups form the foundation of tying users together and giving them permissions to read, write, or execute a given file.

Proper management of users and groups, as well as assigning and revoking permissions, is one of the most important tasks of a system administrator.

The User Database Files

In order to understand the concept of users, you need to know the format of the user database files, /etc/passwd and /etc/shadow. Each line in both files consists of colon-separated fields, one line per user. The format of the password file /etc/passwd is given below.

Username: password: uid: gid: comment: home directory: shell

- **Username: -** Username is the unique name of the user.
- **Password: -** This field contains the encrypted password of the user account. But if the shadowing is enabled this file contains only a X which indicates that the user's password is stored in /etc/shadow file.
- **Uid: -** The uid field contains the unique user identification number of the user.
- **Gid: -** This field contains the user's primary group identification number.
- **Comment:-** This is an optional field which contains some comments about the user – normally his full name.
- **Directory:-** This field contains the path to the user's home directory.
- **Shell:-** This field contain the default shell which has to be activated when the user is logged in.

Following is a sample entry in the /etc/passwd file

ameya:g$12fy:225:150: CWIT, PUNE : /home/ ameya: /bin/bash

The above entry shows that the user name is **ameya**, password is **g$12fy**, uid is **225**, gid is **150**, comment is **CWIT,PUNE**, user's home directory is at **/home/user1** and the default shell of the user is **/bin/bash**.

The Shadow Password System

Shadow passwords offer a number of distinct advantages over the traditional password system, including:

- Improved system security by moving the encrypted passwords normally found in /etc/passwd to

/etc/shadow, which is readable only by root

- Information concerning password aging, how long it has been since a password was last changed.
- Control over how long a password can remain unchanged before the user is required to change it.

The /etc/shadow file contains the following fields:

1. The account name
2. The account's encrypted password
3. The number of days elapsed since 1 January 1970 to the day last password was changed
4. The minimum number of days required between password changes
5. The maximum number of days after which password must be changed
6. The number of days before password expires the user is to be warned
7. The number of days after the password expires before the account is disabled
8. The number of days since 1 January 1970 after which the account is disabled

Tools for User and Group Administration:

Managing users and groups is tedious, but Red Hat Linux provides a few tools and conventions to make users and groups easier to manage.

While you can use useradd to create a new user from the shell prompt, a popular way to manage users and groups is through

```
# redhat-config-users
```

User and Group Configuration:

useradd command

The useradd command can be used to create user accounts. The command format is given below:

useradd [-u UID][-g GID] [-G group1,group2…] [-c comment]

[-s shell] [-d home] [-e expire_date] [-f n] –m <username>

-u UID User identification no. If not specified the next available user ID will be taken as the UID

-g GID To specify the primary group ID for the user

-G group1 To specify the supplementary groups where the user should have the membership

-c To specify the comment.

-s To specify the default shell of the user. If not specified /bin/bash will be taken as the default shell.

-d To specify the path for the home directory. If not specified home directory will be created as /home/<username>

-e To specify the date of expiry of the user account in YYYY-MM-DD format.

-f n Disable the account n days after the account password expires.

-m To create the home directory if it does not exists.

<username> Name of the useraccount.

```
#useradd –u 210 –g 300 –s /bin/ksh -d /home/manoj -m manoj
```

The above command will create a user account named manoj with UID 210, GID 300, Korn Shell as the default shell and home directory as /home/manoj.

The passwd command:

After creating a user account you must set a password for that user account using the passwd command as follows:

```
  # passwd manoj
Enter the password:
Confirm the password:
```

Only root can specify a username along with the passwd command. The other users can use the passwd command without any argument to change their own passwd.

usermod command

The usermod command is used to modify the user parameters as follows

usermod [-u UID] [-g GID] [-G grop1,group2…] [-c comment] [-s shell] [-d home] [-e expire_date] -l -L -U –m <username>.

The options are almost same as useradd command.

-l This option is used to change the user’s login name alone

-L This option is used to lock a user account.

-U used to unlock the useraccount.

The chage command:

The chage command is used to set password aging policies.

```
# chage [options]  username
```

The options are given below

-m Minimum days required between password changes.

-M Maximum number of days a user can use the same password without changing it.

-E <date> Password expiry date in YYYY-MM-DD format.

-W Set warning days as the number of days before password expiration during which the user is warned that his password is due to expire.

The userdel Command:

The userdel command can be used to delete a user account.

```
userdel  [-r]  username
```

The –r option is used to remove the users home directory along with the user account.

Group Administration:

A group contain similar type of users as its members. In traditional Unix environment security and permissions are assigned with the help of group accounts. Linux provides a set of commands for administering the group accounts.

As like /etc/passwd file for user accounts, group account information will be stored in /etc/group file. In this file one line will be available for each group with colon-separated fields as follows

groupname:passwd:gid:userlist

- Groupname is the name of the group
- Password is an optional field containing the encrypted group password.
- Gid is the numeric group ID number.
- Uselist is a comma-separated list of the user account names that comprise the group.
- A typical entry in the group file might resemble the following:
- **dcomp3 : x : 510 : manoj, babu, ram**

The groupadd command

The groupadd command can be used to create new group accounts.

```
groupadd  [ -g GID ]  group name
```

-g is the option used to specify the group identification number. If it is not specified in the command line next available group ID will be taken as the new GID

```
# groupadd  comp3
```

The above command will create a group called comp3

The groupmod command

The groupmod command is used to modify the group parameters, normally the gid or the group name as follows.

groupmod[-g new gid][-n new name] groupname

-g   option is used to change the group ID

–n   option is used to change the group name alone.

The groupdel command

The groupdel command is used to delete the user accounts as follows

groupdel groupname

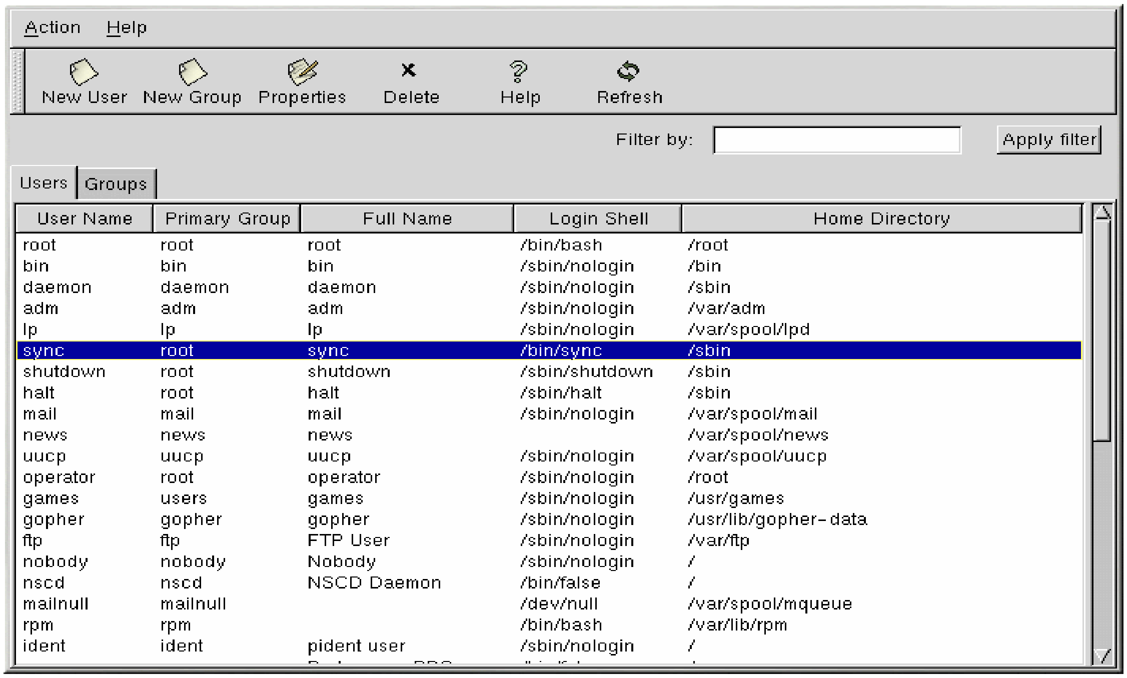
Using Graphical tools

On the GNOME desktop, go to the Main Menu Button   (on the Panel) --> Programs --> System -->

User Manager.

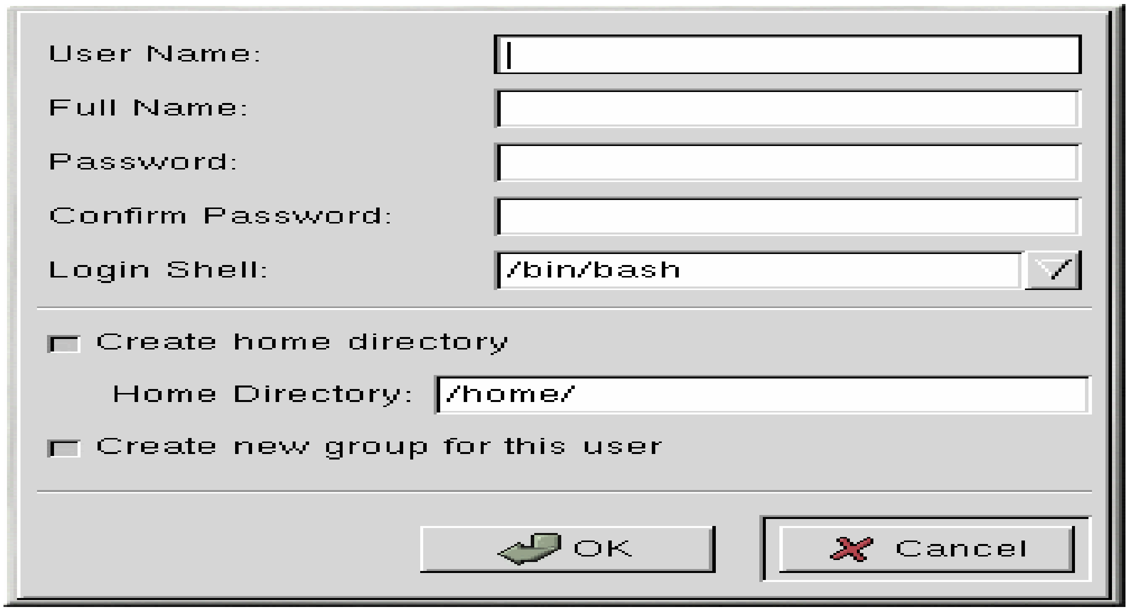
On the KDE desktop, go to the Main Menu Button   (on the Panel) --> Red Hat --> System -->

User Manager.



User Name	Primary Group	Full Name	Login Shell	Home Directory
root	root	root	/bin/bash	/root
bin	bin	bin	/sbin/nologin	/bin
daemon	daemon	daemon	/sbin/nologin	/sbin
adm	adm	adm	/sbin/nologin	/var/adm
lp	lp	lp	/sbin/nologin	/var/spool/lpd
sync	root	sync	/bin/sync	/sbin
shutdown	root	shutdown	/sbin/shutdown	/sbin
halt	root	halt	/sbin/halt	/sbin
mail	mail	mail	/sbin/nologin	/var/spool/mail
news	news	news		/var/spool/news
uucp	uucp	uucp	/sbin/nologin	/var/spool/uucp
operator	root	operator	/sbin/nologin	/root
games	users	games	/sbin/nologin	/usr/games
gopher	gopher	gopher	/sbin/nologin	/usr/lib/gopher-data
ftp	ftp	FTP User	/sbin/nologin	/var/ftp
nobody	nobody	Nobody	/sbin/nologin	/
nscd	nscd	NSCD Daemon	/bin/false	/
mailnull	mailnull		/dev/null	/var/spool/mqueue
rpm	rpm		/bin/bash	/var/lib/rpm
ident	ident	pident user	/sbin/nologin	/

Adding a New User



User Name:

Full Name:

Password:

Confirm Password:

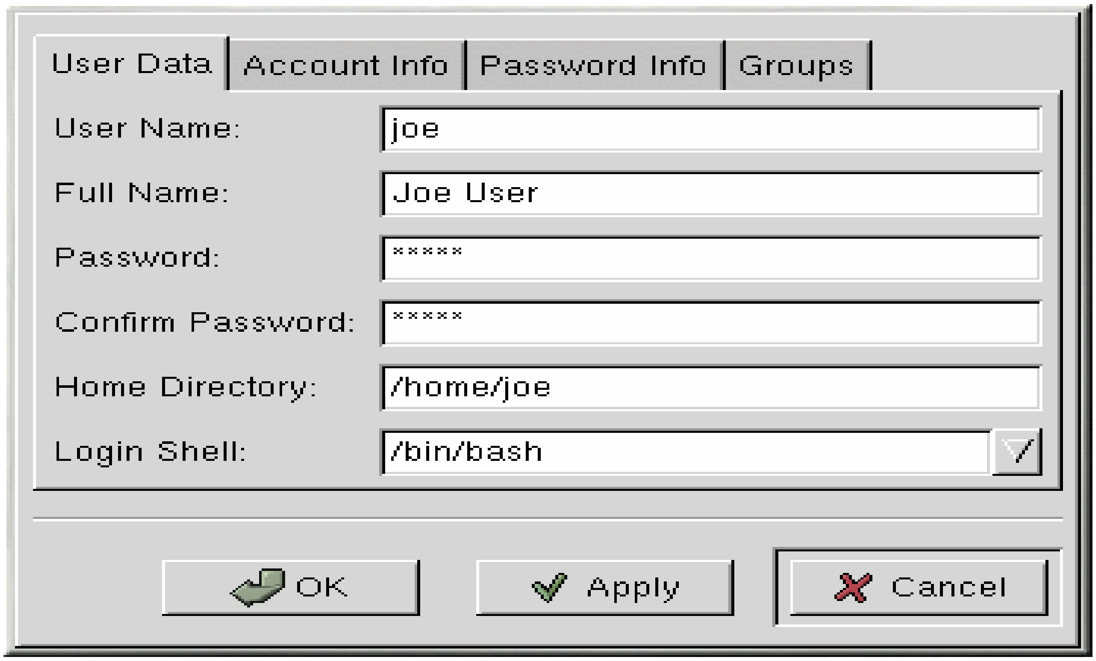
Login Shell: /bin/bash

□ Create home directory

Home Directory: /home/

□ Create new group for this user

Modifying User Properties



User Name: joe

Full Name: Joe User

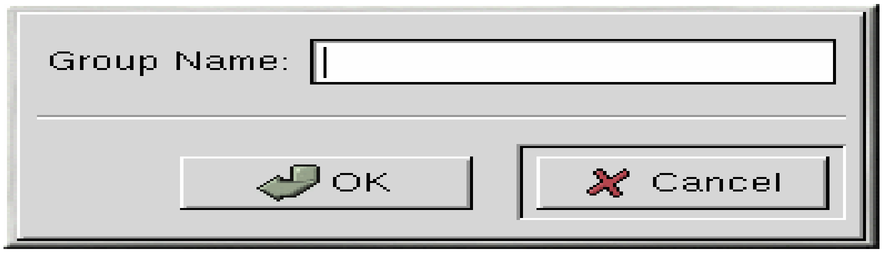
Password: *****

Confirm Password: *****

Home Directory: /home/joe

Login Shell: /bin/bash

Adding a New Group



Group Name:

The su Command

The su – Switch user command is used to get the privilege of a different user without logging to the system using that user name. The command format is as follows.

```
su  [ - ]  username
```

The - is used as a option to get the exact environments like the home directory, path etc of the changed user account. If you have not specified a user name su will try to acquire the root’s privilege.

```
$ su – manoj
password : ***********
$
```

Now you got the privilege of the user manoj.

- **$ exit**
- **$**
- Now you have returned back to the privilege of the original user
- # **su – manoj**
- **$**
- Since you are using su as root it will not ask for the password of the new user.

The sudo - Super User Do

- Another way to switch users or execute commands as others is to use the sudo command. The syntax for sudo is:

  sudo *command*
- The sudo allows you to run programs with the security privileges of another user.
- Like su, if no username is specified, it assumes that you were trying to run commands as the superuser. This why sudo is referred to as superuser do.
- It is commonly used to install, start and stop applications that require root priviledges.
- One of the advantages of using sudo over the su command is that you don't need to know the password of the other user.
- To run a command as the root user, use

```
sudo command
```

- You can specify a user with -u, for example sudo -u root *command* is the same as sudo *command*.
- However, if you want to run a command as another user, you need to specify that with -u. So, for example

```
sudo -u ameya command.
```

The groups Command

The groups command will display the names of the groups where a user is having membership.

```
# groups
     root  bin admin daemon sys ….. etc
 # groups user1
```

The above command will display the names of the groups where the user user1 is having membership.

The id command

This command can be used to get the information about the uid, gid and supplementary group Ids and group names of a user account. If you invoke this command without any argument it will   give details about the current user.

```
# id
   uid=0(root) gid=0(root) groups=0(root), 1(bin), 2(daemon), 3(sys), ……
 # id meera
```

For getting the uid and gid information of a different user you can give the user name as an argument to id command as above.