Unit-3
"User Management"

- User, Local User, Super user, Root account, Managing user
- Group, User Private Group (UPG), Group directories
- Managing group by groupadd, groupmod, groupdel
- GUI user management tools: User admin and KUser
- Password file, Managing user environment
- Adding and removing users with useradd, usermod and userdel
- Managing groups, Controlling access to directories and file using chmod

USER : Local Users & Super users

- A user is a person who enters into the computer system for specific task is called user.
- When we create a user in Linux, system creates a user configuration and directory.
- This configuration file and directory store information about users.
- This configuration file includes home directory, login shell, password, group, encrypted password, encrypted password for group, default login.
- There are two types of users: (1) super user (2) local user.
- Super user is also called system administration; he/she is head of the computer department. He/she is the root user. He/she has all root access.
- All other users than super user are called Local user. They have limited computer access. Basically local users are logged into computer system for their work.

Management of Users:

For user management the following functions are performed.

- ✓ Create a new user
- ✓ Update a user
- ✓ Delete a user
- ✓ Create a user directory
- ✓ Allocate group

Root Account:

- The "root" account is the most privileged account on a Unix system. This account gives you the ability to carry out all side of system administration, including adding accounts, changing user passwords, examining log files, installing software, etc.
- The "root" account has no security restrictions imposed upon it. This means it is easy to perform administrative duties without disturbance. However, the system assumes you know what you are doing, and will do exactly what you request.

Create a New Account:

- In this example, Linux use shadow password suite.
- The Shadow suite is fairly easy to install and will automatically convert your non-shadow password file format over to the new shadow format.

- There are two steps to creating a new user account.
- The first is to actually create the account itself, the second is to provide an alias to their e-mail address (at my place of employment, we follow the convention of "Firstname.Lastname@our.domain.name".)
- To create the account, decide on the username you are going to assign to the user.
- The username is at most 8 characters long, and wherever possible you should choose their last name, or last name and first initial if a user account already exists (the adduser script will detect and prevent you from adding duplicate account names).
- You will then be prompted to enter other information: full name of user, user group (usually the default value), a user id # (automatically assigned), home directory (automatically assigned), a user shell, some password expiration values, and finally the desired password (which won't echo to the screen; you should have the user choose a password between 6 to 8 characters in length for security reasons).
- Please note that everything should be entered in lowercase, except for the full name of the user which can be entered in a "pleasing format" (e.g. Joe Smith) and the password.

➡ *GROUP*

- A group is a set of users. Group information is stored in the /etc/group directory.
- It store information about a group in one line with different field separated by comma.
- These fields are group name, group ID, password, users.
- When user created that time it takes default group, the method of assigning a default group is called <u>user private group scheme</u>.

Managing Groups

You can manage groups using either shell commands or GUI utilities. Groups are an effective way to manage access and permissions, letting you control several users with just their group name.

/etc/group and /etc/gshadow

- The system file that holds group entries is called /etc/group. The file consists of group records, with one record per line and its fields separated by colons.

- A group record has four fields: a group name, a password, its ID, and the users who belongs to this group. The Password field can be left blank. The fields for a group record are as follows:
  - **Group name:** The name of the group, which must be unique.
  - **Password** With security implemented, this field is an x, with the password indicated in the /etc/gshadow file
  - **Group ID** The number assigned by the system to identify this group.
  - **Users :** The list of users that belong to the group, separated by commas.
  - Here is an example of an entry in an /etc/group file. The group is called “root”, the password is managed by shadow security, the group ID is 0, and the users who are part of this group are chris, robert, valerie, and aleina:

```
root:x:0:chris,robert,valerie,aleina
```

(Examples of standard groups are : root, bin, mail, news, games, users, ftp, sys, adm..)

User Private Groups (UPG)

- A UPG is created whenever a new user is added to the system. A UPG has the same name as the user for which it was created and that user is the only member of the UPG. UPGs make it safe to set default permissions for a newly created file or directory which allow both user and that user’s group to make modifications to the file or directory. (Traditionally, users were all assigned to one group named users that subjected all users to the group permission controls for the users group. With UPG, each user has its own group, with its own group permissions.)

Group Directories

- As with users, you can create a home directory for a group.
- To do so, you simply create a directory for the group in the /home directory and change its home group to that group and allow access by any member of the group. The following example creates a directory called engines and changes its group to the engines group:

```
mkdir /home/engines

chgrp engines /home/engines
```

Managing Groups Using groupadd, groupmod, and groupdel

- You can also manage groups with the groupadd, groupmod, and groupdel commands. These command line operations let you quickly manage a group from a terminal window.

groupadd and groupdel

- With the groupadd command, you can create new groups. When you add a group to the system, the system places the group's name in the /etc/group file and gives it a group ID number. If shadow security is in place, changes are made to the /etc/gshadow file. The
- groupadd command only creates the group category. You need to add users to the group individually. In the following example, the groupadd command creates the engines group:

```
# groupadd engines
```

- You can delete a group with the groupdel command. In the next example, the engines group is deleted:

```
# groupdel engines
```

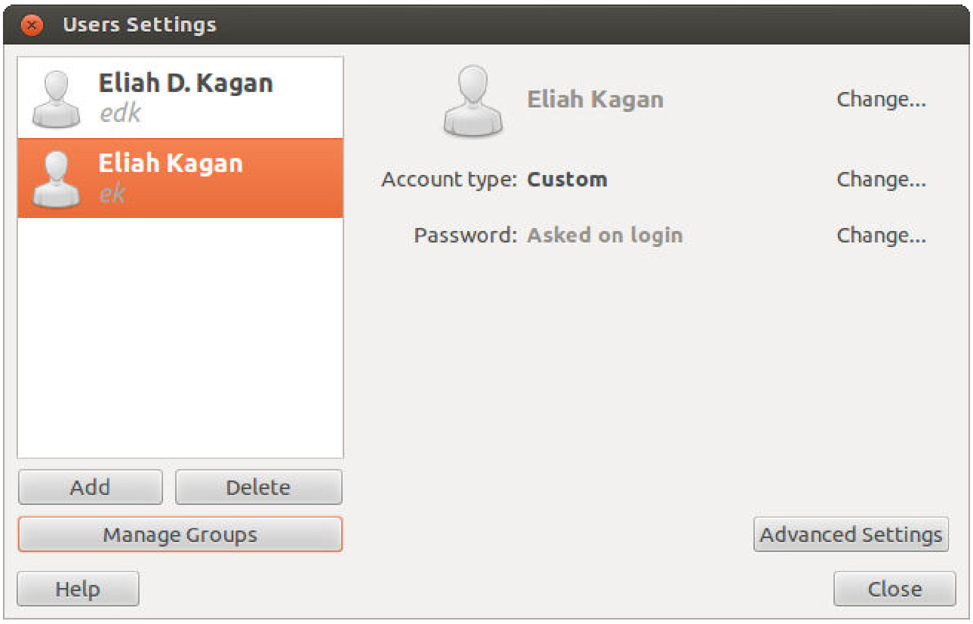
groupmod

- You can change the name of a group or its ID using the groupmod command. Enter groupmod -g with the new ID number and the group name.
- To change the name of a group, you use the -n option. Enter groupmod -n with the new name of the group, followed by the current name. In the next example, the engines group has its name changed to trains:

```
# groupmod -n trains engines
```

GUI User Management Tools : User-Admin & KUser

- User's can be more easily managed using a GUI User management tool like GNOME's user admin and KDE's KUser.
- GNOME's user-admin tool is part of GNOME's system tools package.
- Though some distributions like Red Hat still use their own custom designed tools, other distributions like Ubuntu are making use of GNOME's user-admin tool.
- The Kuser tool has always been available on all distributions with the KDE desktop.
- GNOME's users-admin tool provides a simple interface for adding, modifying, and removing users and groups.
- It opens up with a Users Settings window listing users with their full name, login name, and home directory.
- And Add User button to the side will open a New User Accounts window with Account, User Privileges, and Advanced panels.
- On the Account panel you can enter the login name and password. A profile pop-up menu lets you specify whether to make the user a normal user or an administrator. You can also add in contact information.
- The User Privileges menu lets you control what a user can do, most importantly whether to grant administrative access.
- The Advanced panel lets you specify the user account settings like the home directory, the login shell to use, and what group to belong to.
- Default entries are already set up for you. To later change settings, select the User in the User Settings window and click the Properties button.
- A three panel Account Properties window opens with the same Account, Privileges, and Advanced panels. To delete a user, select it and click the Delete button.
- To manage groups, click on the Groups button. This opens a Group Settings window that lists all groups.
- To add users to a group, select it and click Properties. In the Properties window, users will be listed and you can select the ones you want to add. To add a new group, click on the Add Group button to open a New Group window where you can specify the group name, its id, and select users to add to the group.



Users Settings

Eliah D. Kagan
edk

Eliah Kagan
ek

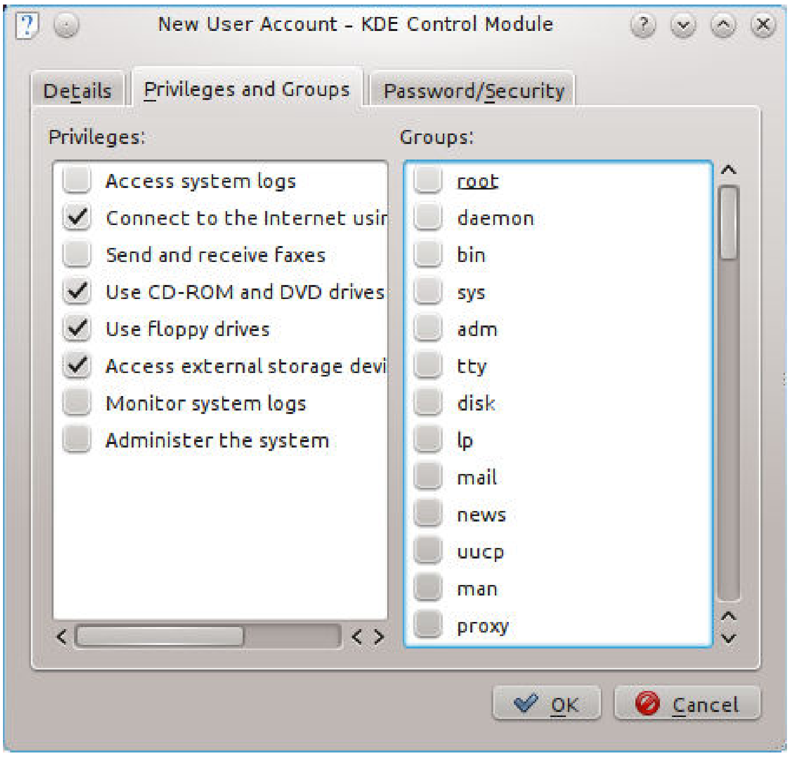
Eliah Kagan

Account type: Custom

Password: Asked on login

GNOME user admin panel

- On KDE, the KDE User Manager (KUser) lets you manage both users and groups.
- The KDE User Manger window displays two panels, one for users and the other for groups.



New User Account – KDE Control Module

Privileges:

☐ Access system logs
☑ Connect to the Internet usir
☐ Send and receive faxes
☑ Use CD-ROM and DVD drives
☑ Use Floppy drives
☑ Access external storage devi
☐ Monitor system logs
☐ Administer the system

Groups:

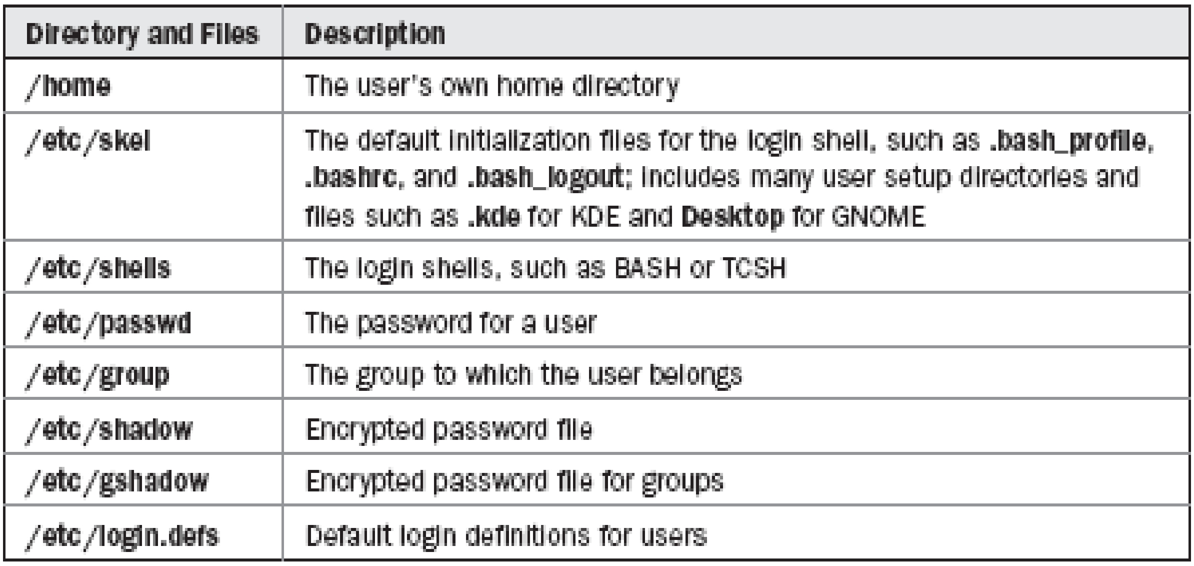
☐ root
☐ daemon
☐ bin
☐ sys
☐ adm
☐ tty
☐ disk
☐ lp
☐ mail
☐ news
☐ uucp
☐ man
☐ proxy

The Users panel lists user's user login, full name, home directory, and login shell, along with either user ID. On the toolbar there are Add, Edit, and Delete buttons for users and groups, as well as Users and Groups menus with corresponding entries.

Initially all system users and groups will also be displayed. Select Hide system users/groups from the Settings menu to display just normal users and groups.

User Configuration Files

- Any utility to manage a user, such as GNOME's users-admin or KDE's KUser, makes use of certain default files, called configuration files, and directories to set up the new account.
- A set of pathnames is used to locate these default files or to indicate where to create certain user directories.
- For example, /etc/skel holds initialization files for a new user. A new user's home directory is created in the /home directory. The following table has a list of the pathnames.



Directory and Files	Description
/home	The user's own home directory
/etc/skel	The default initialization files for the login shell, such as **.bash_profile**, **.bashrc**, and **.bash_logout**; includes many user setup directories and files such as **.kde** for KDE and **Desktop** for GNOME
/etc/shells	The login shells, such as BASH or TCSH
/etc/passwd	The password for a user
/etc/group	The group to which the user belongs
/etc/shadow	Encrypted password file
/etc/gshadow	Encrypted password file for groups
/etc/login.defs	Default login definitions for users

- You can find out which users are currently logged in with the w or who command. The w command displays detailed information about each connected user, such as from where they logged in and how long they have been inactive, and the date and time of login. The who command provides less detailed data.

Managing User Environments

- Each time a user logs in, two profile scripts are executed, a system profile script that is the same for every user, and a user login profile script that can be customized to each user's needs.
- When the user logs out, a user logout script is run. In addition, each time a shell is generated, including the login shell, a user shell script is run. There are different kinds of scripts used for different shells.
- The default shell commonly used is the BASH shell. As an alternative, users can use different shells such as TCSH.

Adding & Removing Users with useradd, usermod, userdel

- Linux also provides the useradd, usermod, and userdel commands to manage user accounts. All these commands take in their information as options on the command line.
- If an option is not specified, they use predetermined default values. These are command line operations.
- To use them on your desktop you first need to open a terminal window (right-click the desktop and select Open Terminal), and then enter the commands at the shell prompt.

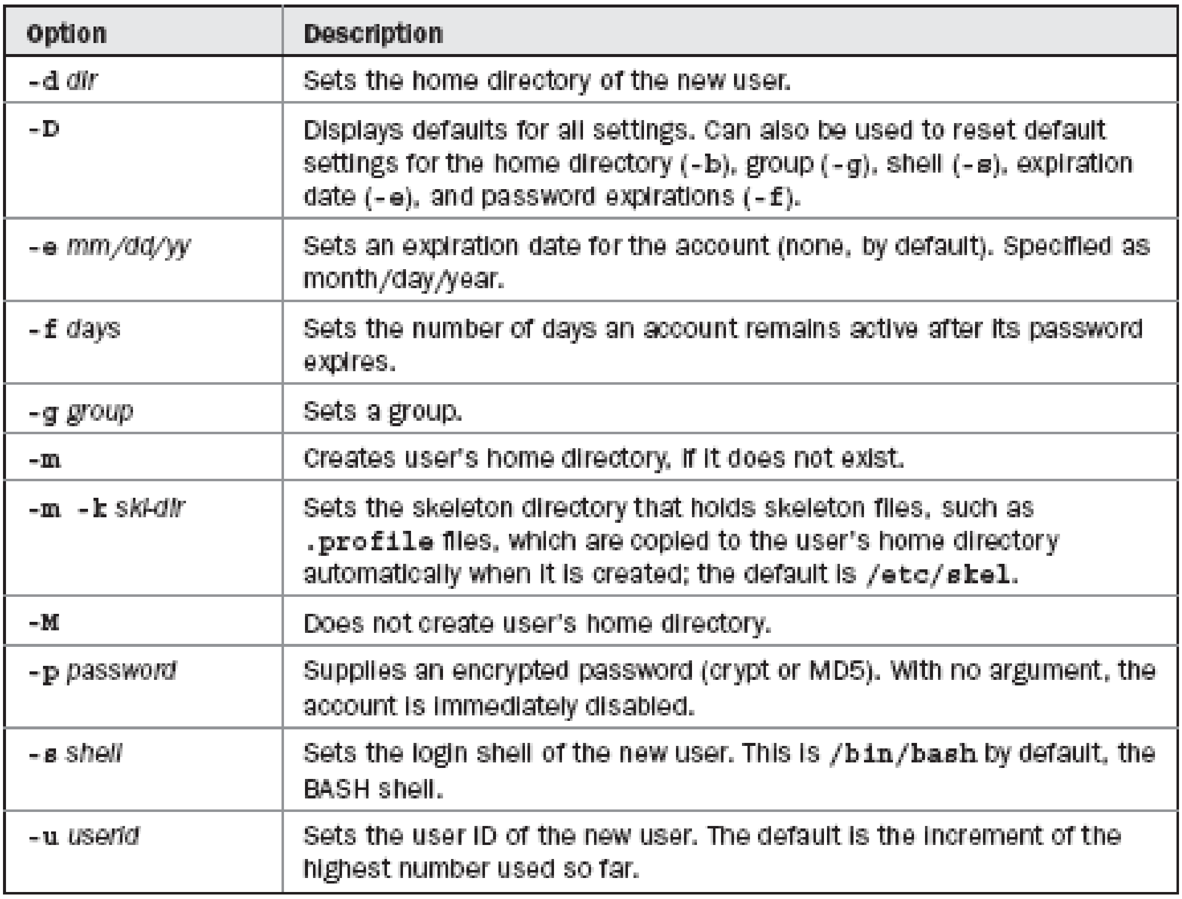
❖ useradd

- With the useradd command, you enter values as options on the command line, such as the name of a user, to create a user account. Then it creates a new login and directory for that name using all the default features for a new account.

```
# useradd chinkal
```

- The useradd utility first checks the /etc/login.defs file for default values for creating a new account.
- For those defaults not defined in the /etc/login.defs file, useradd supplies its own.
- You can display these defaults using the useradd command with the -D option. The default values include the group name, the user ID, the home directory, the skel directory, and the login shell.
- Values the user enters on the command line will override corresponding defaults. The group name is the name of the group in which the new account is placed. By default, this is other, which means the new account belongs to no group.
- The user ID is a number identifying the user account. The skel directory is the system directory that holds copies of initialization files. These initialization files are copied into the user's new home directory when it is created.
- The login shell is the pathname for the particular shell the user plans to use.

The /etc/login.defs file defines the configurations for the shadow password suite. This file is required as absence of this file will not prevent system oprations, but will probably result in undesirable oprations. It is a readable text file.



Option	Description
-d *dir*	Sets the home directory of the new user.
-D	Displays defaults for all settings. Can also be used to reset default settings for the home directory (-b), group (-g), shell (-s), expiration date (-e), and password expirations (-f).
-e *mm/dd/yy*	Sets an expiration date for the account (none, by default). Specified as month/day/year.
-f *days*	Sets the number of days an account remains active after its password expires.
-g *group*	Sets a group.
-m	Creates user's home directory, if it does not exist.
-m -k *skl-dir*	Sets the skeleton directory that holds skeleton files, such as .profile files, which are copied to the user's home directory automatically when it is created; the default is /etc/skel.
-M	Does not create user's home directory.
-p *password*	Supplies an encrypted password (crypt or MD5). With no argument, the account is immediately disabled.
-s *shell*	Sets the login shell of the new user. This is /bin/bash by default, the BASH shell.
-u *userid*	Sets the user ID of the new user. The default is the increment of the highest number used so far.

❖ ***usermod***

- The usermod command enables you to change the values for any of these features. You can change the home directory or the user ID. You can even change the username for the account. The usermod command takes the same options as useradd.

❖ ***userdel***

- When you want to remove a user from the system, you can use the userdel command to delete the user's login. With the -r option, the user's home directory will also be removed. In the next example, the user chinkal is removed from the system:

```
# userdel -r chinkal
```

Administering (Management of) Password

- A user gains access to an account by providing a correct login and password.
- The system maintains passwords in password files, along with login information like the username and ID.
- Tools like the passwd command let users change their passwords by modifying these files; /etc/passwd is the file that traditionally held user passwords, though in encrypted form.
- However, all users are allowed to read the /etc/passwd file, which allows access by users to the encrypted passwords. For better security, password entries are now kept in the /etc/shadow file, which is restricted to the root user.

/etc/passwd

- When you add a user, an entry for that user is made in the /etc/passwd file, commonly known as the password file. Each entry takes up one line that has several fields separated by colons. The fields are as follows:
  - Username Login name of the user
  - Password Encrypted password for the user's account
  - User ID Unique number assigned by the system
  - Group ID Number used to identify the group to which the user belongs
  - Comment Any user information, such as the user's full name
  - Home directory The user's home directory
  - Login shell Shell to run when the user logs in; this is the default shell, usually /bin/bash
- Depending on whether or not you are using shadow passwords, the password field (the second field) will be either an x or an encrypted form of the user's password.
- Linux implements shadow passwords by default, so these entries should have an x for their passwords.
- The following is an example of an /etc/passwd entry.
- For such entries, you must use the passwd command to create a password.
- Notice also that user IDs in this particular system start at 500 and increment by one. The group given is not the generic User, but a group consisting uniquely of that user. For example, the dylan user belongs to a group named Dylan, not to the generic User group.

```
dylan:x:500:500:Dylan:/home/dylan:/bin/bash
chris:x:501:501:Chinkal:/home/Chinkal:/bin/bash
```

/etc/shadow and /etc/gshadow

- The /etc/passwd file is a simple text file and is vulnerable to security breaches.
- Anyone who gains access to the /etc/password file might be able to interpret or crack the encrypted passwords through a brute-force crack. The shadow suite of applications implements a greater level of security.
- These include versions of useradd, groupadd, and their corresponding update and delete programs. Most other user configuration tools support shadow security measures. With shadow security, passwords are no longer kept in the /etc/ password file. Instead, passwords are kept in a separate file called /etc/shadow.
- Access is restricted to the root user. The following example shows the /etc/passwd entry for a user.

  ```
  Chinkal:x:501:501:Chinkal:/home/Chinkal:/bin/bash
  ```

- A corresponding password file, called /etc/gshadow, is also maintained for groups that require passwords.

Password tool :

- To change any particular field for a given user, you should use the user management tools provided, such as the passwd command, adduser, usermod, useradd, and change.
- The passwd command lets you change the password only. Other tools not only make entries in the /etc/passwd file, but also create the home directory for the user and install initialization files in the user's home directory.
- These tools also let you control users' access to their accounts. You can set expiration dates for users or lock them out of their accounts.
- Users locked out of their accounts will have their password in the /etc/shadow file prefixed by the invalid string. Unlocking the account removes this prefix.

Controlling access to directories & file using chmod

✓ Permissions:

- A file or directory may have read, write, and execute permissions.
- When a file is created, it is automatically given read and write permissions for the owner, enabling you to display and modify the file. You may change these permissions to any combination you want. A file can also have read-only permission, preventing any modifications.

✓ Permission Categories

- Three different categories of users can have access to a file or directory: the owner, the group, and all others not belonging to that group.
- The owner is the user who created the file. Any file you create, you own. You can also permit a group to have access to a file.
- Often, users are collected into groups. For example, all the users for a given class or project can be formed into a group by the system administrator. A user can grant access to a file to the members of a designated group.
- Finally, you can also open up access to a file to all other users on the system. In this case, every user not part of the file's group can have access to that file. In this sense, every other user on the system makes up the "others" category.
- If you want to give the same access to all users on your system, you set the same permissions for both the group and the others. That way, you include both members of the group (group permission) and all those users who are not members (others permission).

Read, Write, Execute Permissions:

- Each category has its own set of read, write, and execute permissions.
- The first set controls the user's own access to his or her files—the owner access.
- The second set controls the access of the group to a user's files. The third set controls the access of all other users to the user's files.
- The three sets of read, write, and execute permissions for the three categories—owner, group, and other—make a total of nine types of permissions.
- The ls command with the -l option displays detailed information about the file, including the permissions. e.g $ls –l myfile

✓ Access through Chmod

- chmod stands for "change mode" and it is used to define the way a file can be accessed. You use the chmod command to change different permission configurations. chmod takes two lists as its arguments: permission changes and filenames.
- You can specify the list of permissions in two different ways. One way uses permission symbols (alphanumeric characters) and is referred to as the symbolic method. The other to use octal numbers (The digits 0 to 7).

Suppose, you are the owner of a file named "myfile" and you want to set its permissions.

1. The user can read, write & execute the file.
2. Members of your group can read and execute it.
3. Others may only read it.

```
chmod u=rwx,g=re,o=r myfile
```

This is the symbolic permissions notations. The letter "u" stands for User, "g" stands for Group, "o" stands for others. Similarly, "r" stands for "read", "w" stands for "write" and "x" stands for "execute". The equals sign(=) is is used to assign the permissions separated with commas. You can also do the same with octal permission notations.

```
chmod 754 myfile
```

Here, the digits 7, 5 and 4 each individually represent the permissions for the user, group and others in order. Each digit is a combination of the numbers 4, 2, 1, and 0.

1. "4" stands for "read".
2. "2" stands for "write".
3. "1" stands for "execute".
4. "0" stands for "no permission"

So, 7 is the combination of 4+2+1 (read, write & execute). 5 is 4+0+1 (read, no write, & execute) and 4 is 4+0+0 (read, no write, no execute).