**Linux User Management**

User management includes everything from creating a user to deleting a user on your system. User management can be done in three ways on a Linux system.

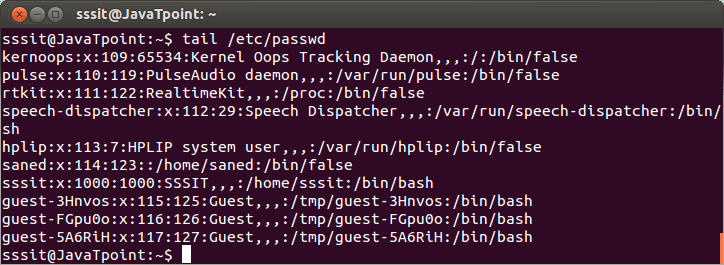
**Graphical tools**are easy and suitable for new users, as it makes sure you'll not run into any trouble.

**Command line tools** includes commands like useradd, userdel, passwd, etc. These are mostly used by the server administrators.

Third and very rare tool is to **edit the local configuration files** directly using vi.

1. /etc/passwd

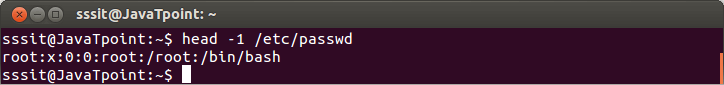
The local user database in Linux is /etc/passwd directory.



Look at the above snapshot, it has seven columns separated by a colon. Starting from the left columns denotes username, an x, user id, primary group id, a description, name of home directory and a login shell.

root

The root user is the superuser and have all the powers for creating a user, deleting a user and can even login with the other user's account. The root user always has userid 0.



useradd

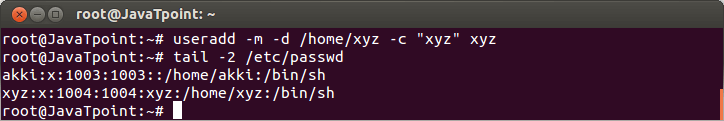
With useradd commands you can add a user.

**Syntax:**

1. useradd -m -d /home/**<userName>** -c "**<userName>**" **<userName>**

**Example:**

1. useradd -m -d /home/xyz -c "xyz" xyz



Look at the above snapshot, we have created a user**xyz** along with creating a home directory (-m), setting the name of home directory (-d), and a description (-c).

The 'xyz' received **userid** as 1004 and**primary group id** as 1004.

/etc/default/useradd

File /etc/default/useradd contains some user default options. The command **useradd -D** can be used to display this file.

**Syntax:**

1. useradd -D

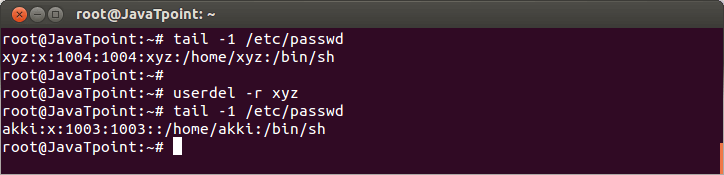


userdel

To delete a user account userdel command is used.

**Syntax:**

1. userdel -r **<userName>**



**Example:**

1. userdel -r xyz

Look at the above snapshot, first we have shown the xyz user account with 'tail' command. To delete it, command **"userdel -r xyz"** is passed.

To recheck, again 'tail' command is passed and as you can see no xyz user account is displayed.

Hence, it is deleted.

usermod

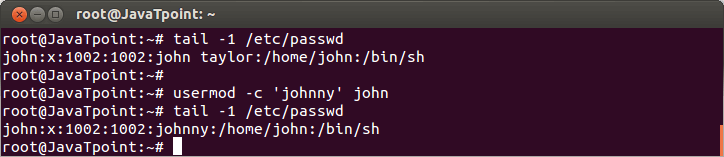
The command usermod is used to modify the properties of an existing user.

**Syntax:**

1. usermod -c **<**'newName'**>** **<oldName>**

**Example:**

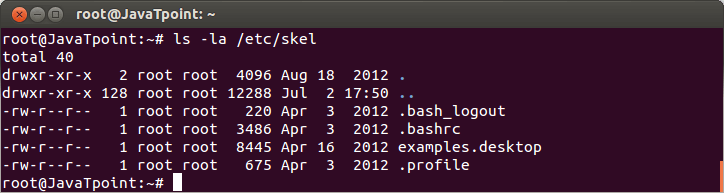
1. usermod -c 'jhonny' john



Look at the above snapshot, user name **john**is replaced by the new user name **jhonny**

/etc/skel/

The /etc/skel/ contains some hidden files which have profile settings and default values for applications. Hence, it serves as a default home directory and user profile. While using useradd -m option, the /etc/skel/ is copied to the newly created directory.



Look at the above snapshot, files of /etc/skel/ is listed.

Deleting Home Directories

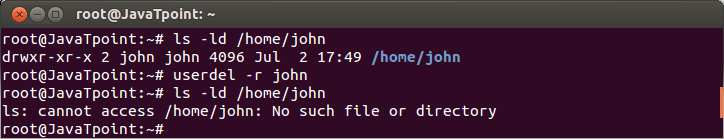
By using**userdel -r** option, you can delete home directory along with user account.

**Syntax:**

1. userdel -r **<userName>**

**Example:**

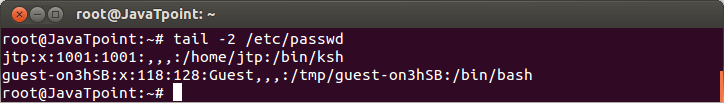
1. userdel -r john



Look at the above snapshot, both home directory as well as user account john is deleted.

Login Shell

The /etc/passwd file also tells about the login shell for the user.



Look at the above snapshot, user guest will log in with**/bin/bash**shell and user jtp will log in with **/bin/ksh shell**.

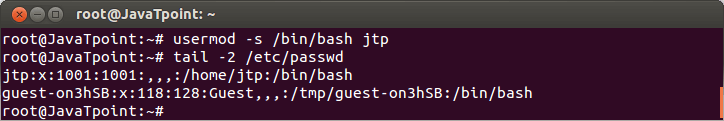
You can change the shell mode with usermod command for a user.

**Syntax:**

1. usermod -s **<newShell>** **<userName>**

**Example:**

1. usermod -s /bin/bash jtp



Look at the above snapshot, shell of jtp is changed to **/bin/bash** from**/bin/ksh.**

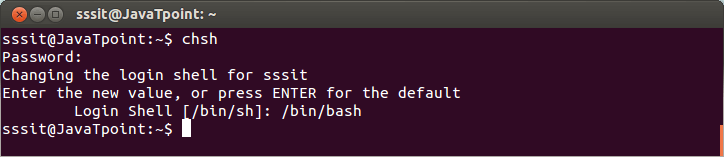
chsh

Users can change their login shell with chsh command.

Both the command**chsh** and**chsh -s** will work to change the shell.

**Syntax:**

1. chsh



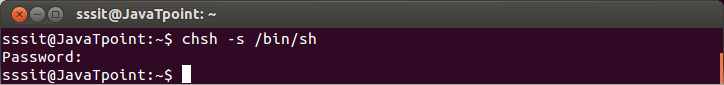
Look at the above snapshot, command chsh has changed the sssit login shell from **/bin/sh** to **/bin/bash**.

**Syntax:**

1. chsh -s **<newShell>**

**Example:**

1. chsh -s /bin/sh



Look at the above snapshot, login shell is changed into /bin/s.

Linux User Password

This chapter tells you about the local users password. You will learn here to change the password, set the password using different methods.

First method is by using **passwd command**.

Second method is with **openssel passwd**command.

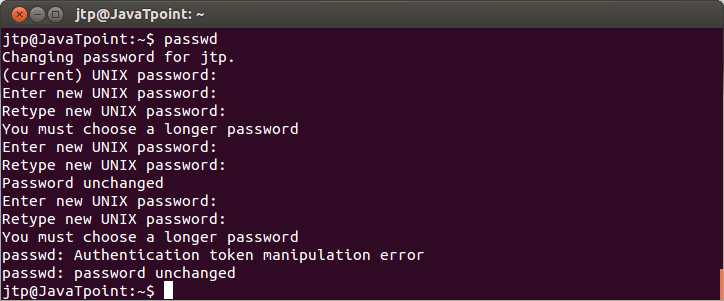
Using passwd command

**passwd**

A user can set the password with the command **passwd**. Old password has to be typed twice before entering the new one.

**Syntax:**

1. passwd



Look at the above snapshot, shell warns the user from creating a simple password. Ultimately, after two or three attempts if password is not changed then the command**passwd fails** and you have to pass the command again.

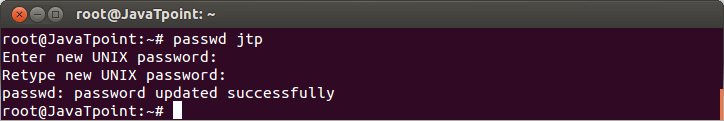
Although, these rules are not applied on the root user neither they need to type the old password. They can change the password directly.

**Syntax:**

1. passwd **<userName>**

**Example:**

1. passwd jtp



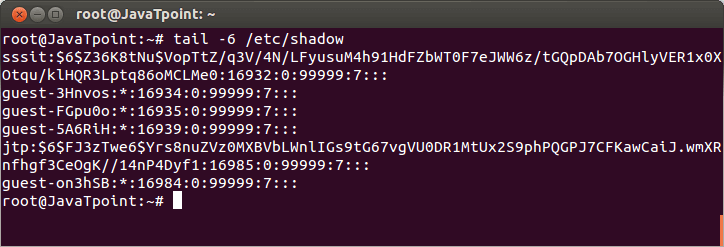
Look at the above snapshot, password is changed successfully without any warning.

Shadow File

Shadow files are the encrypted user passwords which are kept in **/etc/shadow**. This file is read-only directory and can be read only by root.

**Syntax:**

1. /etc/shadow



Look at the above snapshot, the **/etc/shadow** file contains nine columns separeted by colons.

Starting from left to right, these nine columns contain username, encrypted password, last changed password day, number of days password must be left unchanged, password expiry day, warning number of days before password expiry, number of days after expiry before disabling the account, and the day account was disabled. Last column has no meaning yet.

Encryption With passwd

Passwords are always stored in encrypted format. Encryption is done with crypt function. The simplest way to add a user with a password is to add the user with the command **useradd -m** and then set the user's password with command **passwd**.

**Syntax:**

1. useradd -m **<userName>**

**Example:**

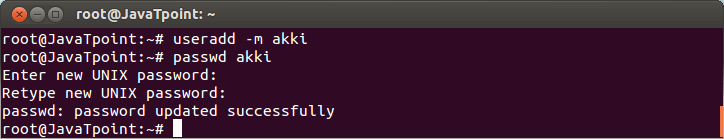
1. useradd -m akki

**Syntax:**

1. passwd **<typePassword>**

**Example:**

1. passwd \*\*\*\*



Look at the above snapshot, user name akki is created with a password successfully.

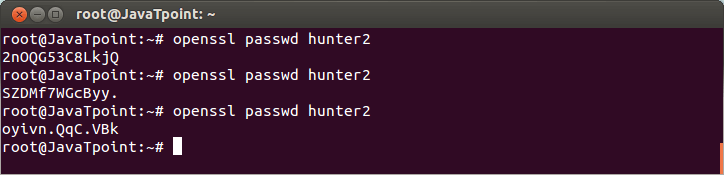
Using openssl passwd

**Encryption With openssl**

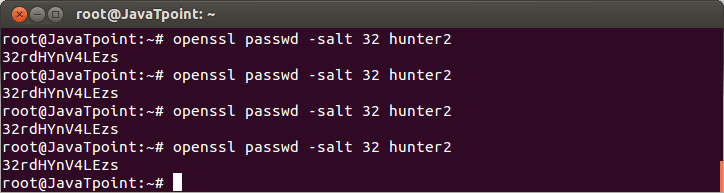
To create a user with a password **-p** option is also used, but that requires an encrypted password.

Third encrypted password can be generated with openssl passwd command.

openssl passwd command can generate several distinct hashes for the same password. To do this, it uses **salt**.



This salt can be chosen and is visible as the first two characters of the hash as shown below.



Look at the above snapshot, the first two characters start from the defined sale **'32'**.

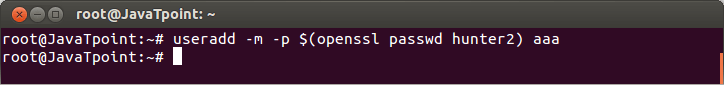
To create a user with password using openssl command, following syntax is used.

**Syntax:**

1. useradd -m -p $(openssl paeewd hunter2) **<userName>**

**Example:**

1. useradd -m -p $(openssl paeewd hunter2) aaa



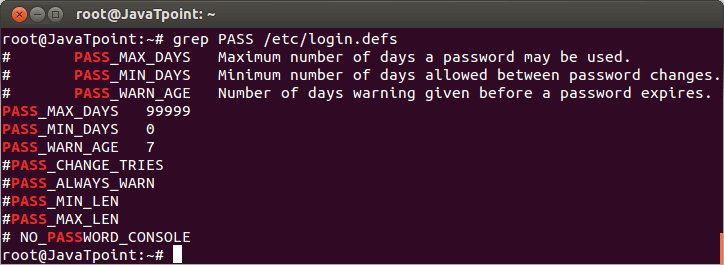
Look at the above snapshot, user aaa is created and its password is kept into command history.

/etc/login.defs

The /etc/login.defs file contains some default settings like password aging and length settings.,

**Syntax:**

1. grep PASS /etc/login.defs



chage

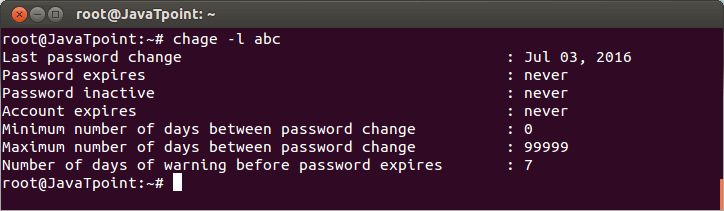
The chage command can be used by a user to know the information about their password. The -l option is used to list the information.

**Syntax:**

1. chage -l **<userName>**

**Example:**

1. chage -l abc



Disabling a Password

Passwords in /etc/shadow are not saved starting with exclamation mark **(!)**. If exclamation mark is present in starting then password can not be used.

This feature can be used to disable a password and the process is called **locking, disabling** and**suspending** a user account. It can be done in **vi** or with **usermod**command.

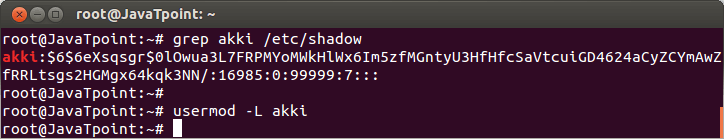
Here, we'll disable the password of akki with usermod command.

**Syntax:**

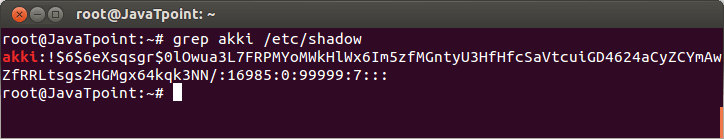
1. usermod -L **<userName>**

**Example:**

1. usermod -L akki



Look at the above snapshot, first command shows hashed password of **akki**, and command **"usermod -L akki"** disables the password of akki. Now user akki can't authenticate using this password.



Look at the above snapshot, hashed password is preceded with **!**, which means it is disabled.

Please note that root user will be able to open the akki account as password is not needed here. And if user akki wouldn't have set password, then akki can also login.

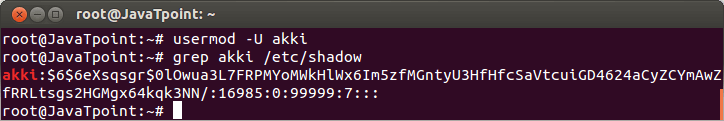
You can**unlock** your account with**usermod -U**.

**Syntax:**

1. usermod -U **<userName>**

**Example:**

1. usermod -U akki



Look at the above snapshot, hashed password of akki is unlocked now as there is no **(!)** mark in starting.

Linux Groups

Users can be listed in different groups. Groups allow us to set permission on the group level instead of setting the permission on individual level.

Every Linux distribution have a graphical tool to manage groups. Groups can be managed by graphical tools, command line tools and by vi or vigr depending upon the user's experience. Only experienced users should use **vi** or **vigr** to manage groups, since it will do proper locking or changes in the file.

**groupadd**

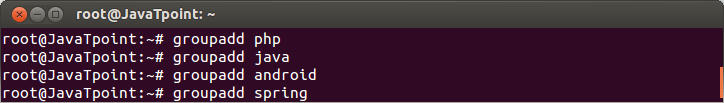
The groupadd command creates or add a group in our system.

**Syntax:**

1. groupadd **<groupName>**

**Example:**

1. groupadd php
2. groupadd java
3. groupadd android
4. groupadd spring



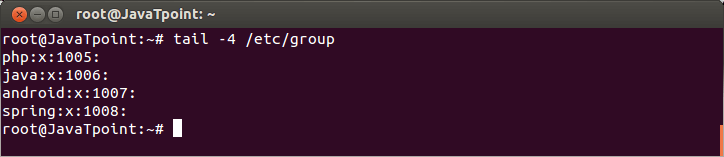
Look at the above snapshot, groups php, java, android and spring are created with groupadd command.

Group File

The /etc/group file defines the group membership. A user can be a member of more than one group.

**Syntax:**

1. /etc/group



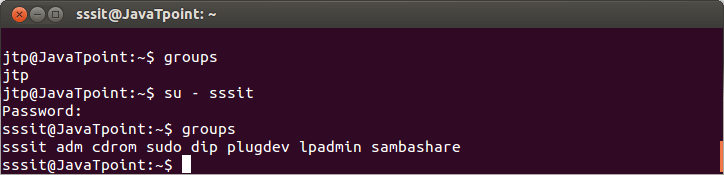
Look at the above snapshot, **first** column indicates group name,**second** is the group's encrypted password which may remain empty also, **third** is group identification (GID) and fourth is the list of members.**Fourth**list is empty as these groups do not have members.

Groups

The group command tells about the group where current user belongs to.

**Syntax:**

1. groups



Look at the above snapshot, user **jtp** and **sssit** belongs to the different groups.

usermod

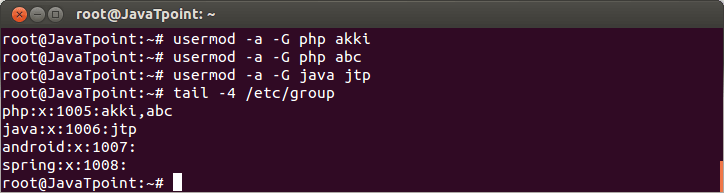
The group members can be edited with **usermod** or **useradd** command. If a group is not listed then by default, usermod command will remove the user from every group of which he is a member. Here, **-a (append)** option is used to prevent this from happening.

**Syntax:**

1. usermod -a -G **<group>** **<userName>**

**Example:**

1. usermod -a -G php akki
2. usermod -a -G php abc
3. usermod -a -G java jtp



Look at the above snapshot, we have displayed the list of /etc/group. User **akki** and **abc** are added into the group**php,** user **jtp**is added into **java**.

groupmod

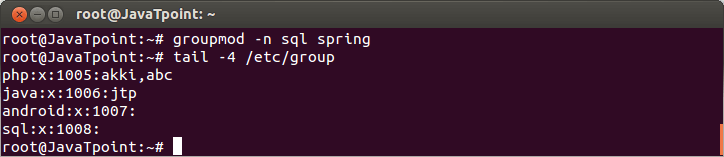
With the help of groupmod command you can change the name of an already existing group.

**Syntax:**

1. groupmod -n **<oldGroup>** **<newGroup>**

**Example:**

1. groupmod -n sql spring



Look at the above snapshot, group **spring** is changed into **sql**.

groupdel

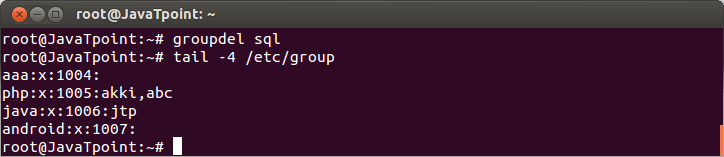
The command groupdel will delete a group permanently from the system.

**Syntax:**

1. groupdel **<group>**

**Example:**

1. groupdel sql



Look at the above snapshot, group **sql** is deleted from the system.

gpasswd

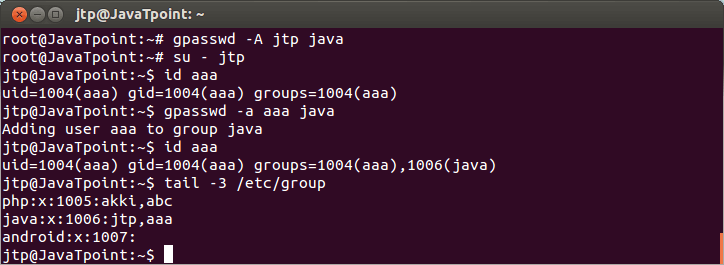
Control of group membership can be passed on to another user with gpasswd command.

**Syntax:**

1. gpsswd -A **<user>** **<group>**

**Example:**

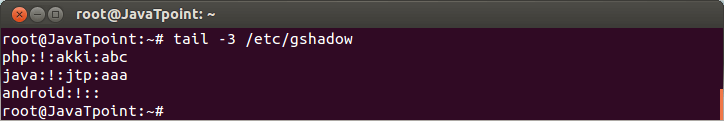
1. gpasswd -A jtp java



Look at the above snapshot, we have passed the membership of**java** to the user with command **"gpasswd -A jtp java"**. Then we**su** to jtp and add **aaa** to java.

Group administrators need not to be a member of the group. They can add or remove a member without being a member of that group.

File **/etc/gshadow**keeps the information about the group administrators as shown in below snapshot.



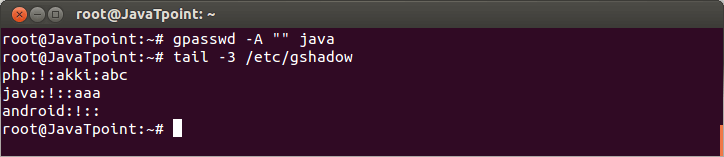
To remove all the administrators from a group, set an empty administrator list.

**Syntax:**

1. gpasswd -A "" **<group>**

**Example:**

1. gpasswd -A "" java



Look at the above snapshot, administrator**jtp** is removed from the group java.

# Linux Add User to Group

Linux allows us to add users to a specific user group. A user account can be added to more than one group. For example, the users that have sudo rights can access the sudo commands to gain related permission. Moreover, we can configure the file permission and other rights to a group.

Adding a user to a group is a straight forward process, all we need to execute the appropriate command. We will walk through all the scenarios related to user and group, such as how to add, delete, remove users to a group. Moreover, we will also learn how to list groups. First, understand the Linux groups:

## Linux Groups

Linux groups are the community of the users that have the right to administrate and organize the users and file permission in a Linux system. The main motive of creating a group is to define the set of permissions such as read and write permission or allowing the shared resources of groups.

### Types of Linux Groups

In Linux-based system, there are two types of groups, which are:

* Primary group
* Secondary or supplementary group

**Primary group:** When we create a file through a specific user account, by default, the filegroup is set to the user's primary group. It will provide the same name as the file user group as the user account name. The primary group stores the user information in /etc/passwd file.

**Secondary or supplementary group:** The main motive to create a secondary group

is to allow the specific permission to limited users. For example, if we want to add any user to the sudo group, the added user will inherit the sudo rights, and be able to run the sudo commands. If we add a user to the docker group, then it will inherit the properties of the docker group and be able to run the docker commands. A **Dockerfile** is a text document that contains all the **commands** a user could call on the **command** line to assemble an image. Using **docker** build users can create an automated build that executes several **command**-line instructions in succession.

A user can be added to precisely one primary group. It is not necessary to add a user to a secondary group, so a user can be added to zero or more secondary groups.

#### Note: Only the root user has the right to add a user to a group.

## Add a Group

To create a new user group on a Linux system, execute the **groupadd command,** followed by the group name. You will need the sudo access to gain the elevated permissions. Execute the command as follows:

1. sudo groupadd jtpGroup

The above command will ask for the system administrative password, type the password. Consider the below output:

## How to add a user to a Group

Adding users to a group is a straight forward process. We can add users to a group by using **usermod** command. To add a user to a group, execute this command, followed by the group name and user name. The group name is the name of a group in which you want to add a particular user.

let's create two users, user1 and user2, execute the command as follows:

1. sudo useradd user1
2. sudo useradd user2

The above command will add the two users called user1 and user2. Consider the below output:

To add the user to a group, execute the following command:

1. sudo usermod -a -G jtpGroup user1

The above command will add the user1 to jtpGroup. Consider the below output:

The -a option used to append the group; it is recommended to use this option when adding a new user to a group. If you do not use the -a option, the user will be removed from any group not specified after -G option.

If the usermod command executes successfully, it will not display any output, but in case of failure, it will post warning messages such as user or group does not exist.

## Add a user to multiple groups

Linux system allows us to add a user to various groups. However, a user only can be added to one primary group, but it can be added to any number of the secondary group.

To assign a user to a secondary group, use the usermod command, followed by group names ( separate them by a comma) and user name.

let's create secondary groups named JtpGroup2 and JtpGroup3:

1. sudo groupadd JtpGroup2
2. sudo groupadd JtpGroup3

Now, add the user1 ( which is already added to JtpGroup, so its primary group is JtpGroup) to the secondary group jtpGroup2, JtpGroup3. Execute the below command.

1. sudo usermod -a -G jtpGroup, JtpGroup2, JtpGroup3 user1

The above command will add the user1 to JtpGroup2 and JtpGroup3 without removing it from JtpGroup. Consider the below output:

we can specify any numbers of groups, specify them by a comma after the primary group.

## List the Groups in Linux

Many groups exist in Linux; these groups can be user-defined as well as pre-defined. We can list all the available groups by using getent command.

To list all the available groups in the Linux system, execute the command as follows:

1. getent group

The above command will display a large list of pre-defined and user-defined groups. Consider the below snap of the output:



The above snap of the output is taken from the last. At the starting of the output, there is an extensive list of pre-defined groups.

## Remove a User from Group

The Linux system allows us to remove a user from a group. To remove a user from a group, execute the **gpasswd command** with -d option followed by group name and user name. Consider the below command:

1. sudo gpasswd -d user1 jtpGroup

The above command will remove the user from the JtpGroup. Consider the below output:

Linux Add User to Group

This group will be assigned a new group configuration at the next login. If the user is logged in, the changes cannot be seen immediately.

## Change a User's Primary Group

A user can be added to more than one group. In such scenarios, there will be only one primary group, and the others will be secondary groups. The access permission of files will be assigned to the primary group. Sometimes we may need to change the primary group; it can be changed by using the usermod command.

To change the user's primary group, execute the usermod command as follows:

1. sudo usermod -g JtpGroup2 user1

Consider the below output:

Linux Add User to Group

The above command will remove the user1 from JtpGroup and add it to JtpGroup2. The -g option is used to assign a user to a primary group, and -G option is used to assign a user to a secondary group.

# Linux id Command

Linux **id** command is used to print the genuine and effective user ID and group ID. A user ID (UID) is a particular user identity, whereas group IDs (GIDs) can contain more than one user's identity. This command is a useful tool for user management.

Tracing the names of users and groups, and their corresponding UIDs and GIDs is a complex task for the new user. This is important for user management. In such cases, the id command will help us.

**Syntax:**

1. id [option]... [user name]

**Options:**

The supported options by the id command are as following:

* **-a:** It is used to ignore the compatibility with other versions.
* **-Z, --context:** It is used to print only the security context of the process.
* **-g, --group:** It is used to print only the effective GID.
* **-G, --groups:** It is used to print all group Ids.
* **-n, --name:** It is used to print a name instead of a number.
* **-r, --real:** It is used to print the real ID instead of the effective ID, with -ugG
* **-u, --user:** It is used to print only the effective UID.
* **-z, --zero:** It is used to delimit entries with NULL characters, except the whitespace;
* **--help:** It used to display the help documentation and exit.

## How to use it?

The id command is a built-in utility; therefore, we do not need to install any additional tools to use it. The 'id' command can be used without arguments. Execute the command as follows:

1. id

From the above command, the output will use the active user. Its default behavior is as follows:

Linux id Command

From the above output, we can see there are several fields. The first two fields are the UID and GID for the user 'javatpoint'. Whereas, rest are the groups with their respective GID that the 'javatpoint' is a part of.

Let's see some other useful examples of the id command.

### Find a specific user's UID

To find a user's specific user ID, execute the command with the '-u' option. For example. Find the user Id for the user 'javatpoint', execute the command as follows:

1. id -u javatpoint

Consider the below output:

Linux id Command

### Find a specific user's GID

To find a specific user's GID, execute the command with the '-g' option. For example, to find the GID for the user 'javatpoint', execute the command as follows:

1. id -g javatpoint

Consider the below output:

Linux id Command

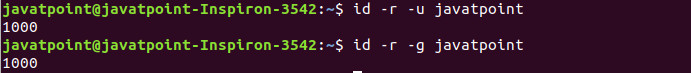
### Display the real GID and UID instead of effective

We can display the real UID and GID instead of effective ids by executing the command with the '-r' option.

To display the real UID and GID for the user 'javatpoint', execute the following commands:

1. id -r -u javatpoint
2. id -r -g javatpoint

Consider the below output:



### Display all the different Group IDs

To display all the different GIDs, execute the command with the '-G' option as follows:

1. id -G

Consider the below output:

Linux id Command

### Display effective GROUP ID

To display only the effective GID, execute the command with the '-g' option as follows:

1. id -g

Consider the below output:

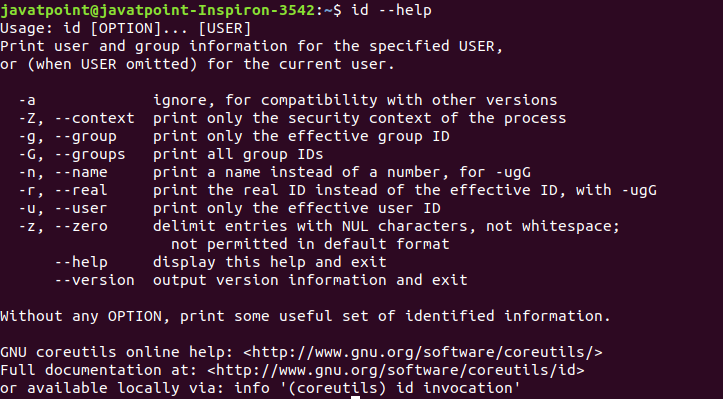
Linux id Command

## Getting Help

If you get stuck during the use of the id command, you can take help from your terminal. You can read the manual page and help documentation. To take help, execute the below command:

1. id --help

The above command will display the help documentation as follows:



To read the manual, execute the command as follows:

1. man id

It will display the manual as given below:

