

UNIX / LINUX - USER ADMINISTRATION

<https://www.tutorialspoint.com/unix/unix-user-administration.htm>

Copyright © tutorialspoint.com

Advertisements

In this chapter, we will discuss in detail about user administration in Unix.

There are three types of accounts on a Unix system –

Root account

This is also called **superuser** and would have complete and unfettered control of the system. A superuser can run any commands without any restriction. This user should be assumed as a system administrator.

System accounts

System accounts are those needed for the operation of system-specific components for example mail accounts and the **sshd** accounts. These accounts are usually needed for some specific function on your system, and any modifications to them could adversely affect the system.

User accounts

User accounts provide interactive access to the system for users and groups of users. General users are typically assigned to these accounts and usually have limited access to critical system files and directories.

Unix supports a concept of *Group Account* which logically groups a number of accounts. Every account would be a part of another group account. A Unix group plays important role in handling file permissions and process management.

Managing Users and Groups

There are four main user administration files –

- **/etc/passwd** – Keeps the user account and password information. This file holds the majority of information about accounts on the Unix system.
- **/etc/shadow** – Holds the encrypted password of the corresponding account. Not all the systems support this file.
- **/etc/group** – This file contains the group information for each account.
- **/etc/gshadow** – This file contains secure group account information.

Check all the above files using the **cat** command.

The following table lists out commands that are available on majority of Unix systems to create and manage accounts and groups –

Sr.No.	Command & Description
--------	-----------------------

1	useradd Adds accounts to the system
2	usermod Modifies account attributes
3	userdel Deletes accounts from the system
4	groupadd Adds groups to the system
5	groupmod Modifies group attributes
6	groupdel Removes groups from the system

You can use [Manpage Help](#) to check complete syntax for each command mentioned here.

Create a Group

We will now understand how to create a group. For this, we need to create groups before creating any account otherwise, we can make use of the existing groups in our system. We have all the groups listed in */etc/groups* file.

All the default groups are system account specific groups and it is not recommended to use them for ordinary accounts. So, following is the syntax to create a new group account –

```
groupadd [-g gid [-o]] [-r] [-f] groupname
```

The following table lists out the parameters –

Sr.No.	Option & Description
1	<p>-g GID</p> <p>The numerical value of the group's ID</p>
2	<p>-o</p> <p>This option permits to add group with non-unique GID</p>
3	<p>-r</p> <p>This flag instructs groupadd to add a system account</p>
4	<p>-f</p> <p>This option causes to just exit with success status, if the specified group already exists. With -g, if the specified GID already exists, other <i>unique</i> GID is chosen</p>
5	<p>groupname</p> <p>Actual group name to be created</p>

If you do not specify any parameter, then the system makes use of the default values.

Following example creates a *developers* group with default values, which is very much acceptable for most of the administrators.

```
$ groupadd developers
```

Modify a Group

To modify a group, use the **groupmod** syntax –

```
$ groupmod -n new_modified_group_name old_group_name
```

To change the *developers_2* group name to *developer*, type –

```
$ groupmod -n developer developer_2
```

Here is how you will change the financial GID to 545 –

```
$ groupmod -g 545 developer
```

Delete a Group

We will now understand how to delete a group. To delete an existing group, all you need is the **groupdel command** and the **group name**. To delete the financial group, the command is –

```
$ groupdel developer
```

This removes only the group, not the files associated with that group. The files are still accessible by their owners.

Create an Account

Let us see how to create a new account on your Unix system. Following is the syntax to create a user's account –

```
useradd -d homedir -g groupname -m -s shell -u userid accountname
```

The following table lists out the parameters –

Sr.No.	Option & Description
1	-d homedir Specifies home directory for the account
2	-g groupname Specifies a group account for this account
3	-m Creates the home directory if it doesn't exist
4	-s shell Specifies the default shell for this account

5	-u userid You can specify a user id for this account
6	accountname Actual account name to be created

If you do not specify any parameter, then the system makes use of the default values. The **useradd** command modifies the **/etc/passwd**, **/etc/shadow**, and **/etc/group** files and creates a home directory.

Following is the example that creates an account **mcmohd**, setting its home directory to **/home/mcmohd** and the group as **developers**. This user would have Korn Shell assigned to it.

```
$ useradd -d /home/mcmohd -g developers -s /bin/ksh mcmohd
```

Before issuing the above command, make sure you already have the **developers** group created using the **groupadd** command.

Once an account is created you can set its password using the **passwd** command as follows –

```
$ passwd mcmohd20
Changing password for user mcmohd20.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

When you type **passwd accountname**, it gives you an option to change the password, provided you are a superuser. Otherwise, you can change just your password using the same command but without specifying your account name.

Modify an Account

The **usermod** command enables you to make changes to an existing account from the command line. It uses the same arguments as the **useradd** command, plus the **-l** argument, which allows you to change the account name.

For example, to change the account name **mcmohd** to **mcmohd20** and to change home directory accordingly, you will need to issue the following command –

```
$ usermod -d /home/mcmohd20 -m -l mcmohd mcmohd20
```

Delete an Account

The **userdel** command can be used to delete an existing user. This is a very dangerous command if not used with caution.

There is only one argument or option available for the command **.r**, for removing the account's home directory and mail file.

For example, to remove account *mcmohd20*, issue the following command –

```
$ userdel -r mcmohd20
```

If you want to keep the home directory for backup purposes, omit the **-r** option. You can remove the home directory as needed at a later time.