Tutorial Link https://course.testpad.chitkara.edu.in/tutorials/Linux - Managing Users/60cc92c7cfaf5f6628f2d40b

TUTORIAL

Linux - Managing Users

Topics

Managing Users

All Linux users are assigned a unique user ID (uid), which is just an integer; normal users start with a uid of 1000 or greater. Linux uses groups for organizing users. Groups are collections of accounts with certain shared permissions. Control of group membership is administered through the /etc/group file, which shows a list of groups and their members. By default, every user belongs to a default or primary group. When a user logs in, the group membership is set for their primary group and all the members enjoy the same level of access and privilege. Permissions on various files and directories can be modified at the group level.

Users also have one or more group IDs (gid), including a default one which is the same as the user ID. These numbers are associated with names through the files /etc/passwd and /etc/group. Groups are used to establish a set of users who have common interests for the purposes of access rights, privileges, and security considerations. Access rights to files (and devices) are granted on the basis of the user and the group they belong to.

User Accounts

A user account will provide you the system access privileges for a particular Linux system. In other words, to use the resources of a

Linux system, you need a valid user account and resource permissions.

Generally accounts will be of three types: root, system, users.

Root account:

The root user has complete control over the system. Complete can be used here, as root can run any command which may be destroy the system also. So the root user can do anything to the system with no restrictions.

System Account:

These are the accounts required to configure the system like mail account, ssh account etc. They are generally associated with an software to run or stop the services of that particular software.

User Account:

These are used to provide interactive access to the system. These are generally used to use the system with limited functionalities as compared to root account. We can manipulate the user accounts on our system by using some commands. Also there are some files used by default to save the settings and configurations for users. /etc/passwd is a file used to store the information of user accounts and passwords (in encrypted form). The typical entry in this file is as:

```
user1:x:510:500:CodeQuotient:/home/user1:/bin/bash
```

Each information is separated by (:) colon symbol. Following are the components: -

```
user1        : Name of the user or called Login ID
x        : Password in encrypted form
510        : User ID
500        : Default GID
CodeQuotient    : Comment (if any)
/home/user1    : Home directory
/bin/bash    : Login shell name
```

If the passwords are used as shadow passwords, then encrypted password will be stored in /etc/shadow file instead of here.

To manage the user accounts following commands will be used: -

```
Command         Description
useradd         Add a user account
usermod         Modify account attributes
userdel         Delete an user's account
groupadd        Add a group to system
groupmod        Modify group attributes
groupdel        Deletes the group
```

User management

useradd, usermod, userdel:

The useradd command is used to add the users. Its syntax is as below:

```
useradd -c comment -d home_directory -e expiration_date -f
inactive_days -g primary
_group -G secondary_groups -m -s shell -u user_id accountname
```

For example, to create an account for a user named user2, whose real name is Gopal. Gopal needs the account until Dec 31, 2030. His primary group is users, and authors is his secondary group. He

has requested the Bourne shell for his default shell. He isn't sure he will be using this system, so let's disable his account if he hasn't used it within 60 days. The useradd command to create this account is:

```
user1@home:~$ useradd -c “Gopal” -d /home/user2 -e 311235 -f 60 -
g users -G authors -m -s
```

```
  /bin/bash -u 1000 user2
```

After this command runs, a password to the account must be set using the passwd accountname command. To create the password for Gopal's account, root would type:

```
user1@home:~$ passwd user2
```

The new account owner should change the password immediately.

useradd command will modify all required files automatically and create the home directory specified and then configure the user account accordingly.

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 instance, to change Gopal's account name to Girdhar and provide him with a home directory, you'd issue the following command:

```
user1@home:~$ usermod -l Girdhar Gopal
```

The above command will change the name from Gopal to new name as Girdhar.

The userdel command is extremely easy to use and can therefore be dangerous 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 spool (mail file). Here's how to remove the Girdhar account:

```
user1@home:~$ userdel -r Girdhar
```

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

Typing id with no argument gives information about the current user, as in:

```
$ id
```

Group Management

groupadd, groupmod, groupdel:

groupadd is used to add a new user group. It has below syntax:

```
groupadd -g group_id group_name
```

To create a new group for the CodeQuotient Developers called CQ-Develop and assign it a unique GID of 535, for example, you'd use:

```
user1@home:~$ groupadd -g 535 CQ_Develop
```

This command makes the appropriate entry in the /etc/group file.

To modify a group, use the groupmod syntax:

```
groupmod -n new_modified_group_name old_group_name
```

To change the CQ_develop group name to development, type:

```
user1@home:~$ groupmod -n development CQ_develop
```

You can also use the -g option with groupmod to change the GID of the group.

To delete an existing group, all you need are the groupdel command and the group name. To delete the development group, the command is:

```
user1@home:~$ groupdel development
```

This removes only the group, not any files associated with that group. (The files are still accessible by their owners.) We also have some GUI based tool in different distributions of Linux to manage the users just like windows environment.