Chapter 7. Configuring and Administering Linux

This chapter equips you to perform common system administration tasks. By using a few simple commands and a text editor, you can override many configuration choices made during system installation. You'll learn how to add, delete, and modify user accounts, and how to add and delete groups and change their membership. You'll learn how to configure swap space and how to cause Linux to automatically mount filesystems. This chapter doesn't deal with the configuration and administration of network facilities, such as servers. Those topics are covered in **Chapter 10**, **Chapter 11**, and **Chapter 12**.

User and Group Administration

In this section, you'll learn how to perform common administrative tasks affecting users and groups. Most system administration tasks require that you login as `root`. Throughout this section and subsequent section, you should assume that you must be logged in as `root`, unless directed otherwise.

Creating a User Account

To create a user account, you use the **adduser** command, which has the form:

```
adduser userid
```

where *userid* specifies the name of the user account that you want to create. The command prompts you for the information needed to create the account.

Here's a typical example of using the command, which creates a user account named `newbie`:

```
debian:~# adduser newbie
Adding user newbie...
Adding new group newbie (1001).
Adding new user newbie (1001) with group newbie.
Creating home directory /home/newbie.
Copying files from /etc/skel
Changing password for newbie
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
Re-enter new password:
Password changed.
Changing the user information for newbie
Enter the new value, or press return for the default

    Full Name []: Newbie Dewbie
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [y/n] y
debian:~#
```

Notice that the lines where the password was typed were overwritten by the subsequent lines. Moreover, for security, passwords are not echoed to the console as they are typed.

Notice also that several of the information fields were omitted—for example, Room Number. You can specify such information if you think it may be useful, but the system makes no use of the information and doesn't require you to provide it.

TIP

The similarly named **useradd** command also creates a user account, but does not prompt you for the password or other information.

When the command establishes a user account, it creates a home directory for the user. In the previous example, the command would have created the directory `/home/newbie`. It also places several configuration files in the home directory, copy-

ing them from the directory `/etc/skel`. These files generally have names beginning with the dot (.) character, so they are hidden from an ordinary **ls** command. Use the *–a* argument of **ls** to list the names of the files. The files are generally ordinary text files, which you can view with a text editor, such as **ae**. By modifying the contents of such files, you can control the operation of the associated application. For example, the `.bashrc` file controls the operation of the BASH shell, which you'll learn more about in **Chapter 13**.

Changing a User's Name

You can change the name associated with a user account, by using the **chfn** command:

```
chfn -f name userid
```

where *name* specifies the new name and *userid* specifies the account to be modified. If the name contains spaces or other special characters, it should be enclosed in double quotes (“). For example, to change the name associated with the account `newbie` to Dewbie Newbie, you would enter the following command:

```
chfn -f "Dewbie Newbie" newbie
```

Changing a User Account Password

From time to time, you should change your password, making it more difficult for others to break into your system. As system administrator, you may sometimes need to change the password associated with a user's account. For instance, some users have a bad habit of forgetting their password. They'll come to you, the system administrator, seeking help in accessing their account.

To change a password, you use the **passwd** command. To change your own password, enter a command like this one:

```
passwd
```

This command changes the password associated with the current user account. You don't have to be logged in as `root` to change a password. Because of this, users can change their own passwords without the help of the system administrator. The `root`

user, however, can change the password associated with any user account, as you'll see shortly. Of course, *only* `root` can do so—other users can change only their own password.

The **passwd** command initiates a simple dialog that resembles the following:

```
$ passwd
Changing password for newbie
Old password:
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.
```

Notice the restrictions governing the choice of password, which are designed to prohibit passwords that might be easily guessed. If you choose a password that violates these restrictions, the command will refuse the password, prompting you for another.

As the `root` user, you can change the password associated with any user account. The system doesn't ask you for the current password, it immediately prompts for the new password:

```
debian:~# passwd newbie
Changing password for newbie
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.
```

Information on users is stored in the file `/etc/passwd`, which you can view using a text editor. Any user can read this file, though only the `root` user can modify it. If you selected shadow passwords, passwords are encrypted and stored in the file `/etc/shadow`, which can be read only by the `root` user.

Configuring Group Definitions

Recall from **Chapter 4** that Linux uses groups to define a set of related user accounts that can share access to a file or directory. You probably won't often find it necessary to configure group definitions, particularly if you use your system as a desktop system rather than a server. However, when you wish, you create and delete groups and modify their membership lists.

Creating a group

To create a new group, use the **groupadd** command:

```
groupadd group
```

where `group` specifies the name of the group to be added. Groups are stored in the file `/etc/group`, which can be read by any user but modified only by `root`.

For example, to add a group named `newbies`, you would enter the following command:

```
groupadd newbies
```

Deleting a group

To delete a group, user the **groupdel** command:

```
groupdel group
```

where `group` specifies the name of the group to be deleted. For example, to delete the group named `newbies`, you would enter the following command:

```
groupdel newbies
```

Adding a member to a group

To add a member to a group, you use a special form of the **adduser** command:

```
adduser user group
```

where `user` specifies the member and `group` specifies the group to which the member is added. For example, to add the user `newbie01` to the group `newbies`, you would enter the following command:

```
adduser newbie01 newbies
```

Removing a member from a group

Unfortunately, no command removes a user from a specified group. The easiest way to remove a member from a group is by editing the `/etc/group` file. Here's an excerpt from a typical `/etc/group` file:

```
users:x:100:
nogroup:x:65534:
bmccarty:x:1000:
newbies:x:1002:newbie01,newbie02,newbie03
```

Each line in the file describes a single group and has the same form as other lines, consisting of a series of fields separated by colons (:). The fields are:

Group name
: The name of the group.

Password
: The encrypted password associated with the group. This field is not generally used, containing an *x* instead.

Group ID
: The unique numeric ID associated with the group.

Member list
: A list of user accounts, with a comma (,) separating each user account from the next.

To remove a member from a group, first create a backup copy of the `/etc/group` file:

```
cp /etc/group /etc/group.SAVE
```

The backup can prove helpful if you modify the file incorrectly. Next, open the `/etc/group` file in a text editor. Locate the line that describes the group and delete the user name and the following comma, if any. Save the file, exit the editor, and check your work.

Deleting a User Account

To delete a user account, use the **userdel** command:

```
userdel user
```

where *user* specifies the account to be deleted. If you want to delete the user's home directory, its files and subdirectories, use this form of the command:

```
userdel -r user
```

WARNING

Because deleted files can't generally be recovered, you should backup potentially useful files before deleting a user account.

Configuring Access to Shells

The BASH shell, which you met in **Chapter 4**, is the most popular, but not the only Linux shell. Others include:

ash
: a version of the **sh** shell that resembles the System V shell

csh
: the C shell, favored by many users for interactive use

ksh
: the Korn shell, the third major Unix shell

sh
: the Bourne shell, a precursor of BASH, also known as the **bsh** shell

tcsh
: an enhanced version of **csh**

zsh
: the Z shell, a feature-packed version of the Korn shell

When you create a new user, the system automatically assigns the shell (command interpreter) that Linux presents to the user when the user logs in. Debian GNU/Linux assigns the BASH shell, as specified by the file `/etc/adduser.conf`. However, you can assign another shell, if you prefer. The shell must be on the list of available shells, which resides in the file `/etc/shells`.

Previous chapter
< Configuring GNOME

Next chapter
Filesystem Administration >