



Managing Users

Linux was designed from the ground up to be a multiuser system. When powerful Linux machines are deployed in huge data centers, they are capable of serving hundreds, if not thousands, of users at the same time. In a more domestic setting, such as when Ubuntu is installed on a desktop PC, *multiuser* means that several family members can have their own login on the PC. They'll get their own desktop environment that is separate from that of the other users and their own file storage area.

And even if you're the only person using your PC, you can still take advantage of Ubuntu's multiuser capabilities. Consider creating user accounts for various aspects of your life—perhaps one for work and one for time spent browsing the Web. Each user account can be tailored to a specific need.

In this chapter, you'll learn how to administer multiple user accounts.

Understanding User and Group Accounts

The concept of users and file ownership was explained in Chapter 14, but let's take a moment to recap and elaborate on some important points.

Users and Groups

Each person who wishes to log in to Ubuntu must have a user account. This will define what that user can and cannot do on the system, with specific reference to files and folders. Because Ubuntu is effectively one large file system, with even hardware devices seen as individual files (see Chapter 14), this means that user permissions lie at the heart of controlling the entire system. They can limit which user has access to which hardware and software, and therefore control access to various PC functions.

Each user also belongs to a group. Groups have the same style of permissions as individual users. File or folder access can be denied or granted to a user, depending on that person's group membership.

Note As in real life, a group can have many members and can be based around various interests. In a business environment, this might mean that a group is created for members of the accounting department and the human resources department, for example. By changing the permissions on files created by the group members, each group can have files that only the group members can access (although, as always, anyone with superuser powers can access all files).

On a default Ubuntu system with just a handful of users, the group concept might seem somewhat redundant. However, the concept of groups is fundamental to the way Ubuntu works and cannot be avoided. Even if you don't make use of groups, Ubuntu still requires your user account to be part of one.

In addition to actual human users, the Ubuntu system has its own set of user and group accounts. Various programs that access hardware resources or particular sets of files are part of these groups. Setting up system users and groups in this way makes the system more secure and easier to administer.

Root User

On most Linux systems, the root user has power over the entire system. Root can examine any file and configure any piece of hardware. Root typically belongs to its own unique group, also called root.

Ubuntu is different from most Linux distributions in that the root account isn't used by default. Instead, certain users, including the one set up during installation, can “borrow” root-like, or superuser, powers by simply typing their login password. This is done by preceding commands with `sudo` or `gksu` at the command-line prompt or as needed when using GUI programs that affect system settings. For some programs, including System Administration ► Users and Groups, you need to click an Unlock button to gain superuser powers. Until you unlock the Users and Groups program, most of the buttons are grayed out and unusable.

If you wish, you can activate the root user account on your system for administration purposes. To activate the root account, use the following command in a terminal window (see Chapter 13 for details on issuing commands in a terminal window):

```
sudo passwd root
```

After typing your own login password, you'll be invited to define a password for the root user.

Because of its power, the root user can cause a lot of accidental damage, so by default Ubuntu prevents you from logging in as root. Instead, you can switch to being the root user temporarily from an ordinary user account by typing the following in a terminal window:

```
su
```

You will be prompted for the root password, and then given root powers for as long as you need. When you've finished, type `exit`, and you'll be returned to your ordinary user account.

Tip You can tell when you're logged in as the root user because the command prompt will end with a hash symbol (#). When logged in as an ordinary user, it ends with a dollar sign (\$). The hash symbol should be seen as a warning that you now have unrestricted control over the system, so be careful what you type and double-check everything before hitting Enter!

As an alternative to setting the root password, you can simply type the following whenever you want to switch to the root user account:

```
sudo su
```

You'll be prompted for your login password, in exactly the same way as if you had just preceded a command with `sudo`. After this, you'll have the powers of the root user. To quit the root user account, type `exit`.

UIDs and GIDs

Although we talk of user and group names, these are only provided for the benefit of humans. Internally, Ubuntu uses a numerical system to identify users and groups. These are referred to as user IDs (UIDs) and group IDs (GIDs), respectively.

Under Ubuntu, all the GID and UID numbers below 1000 are reserved for the system. This means that the first non-root user account created during installation will probably be given a UID of 1000. In addition, any new groups created after installation are numbered from 1000. On one test system, the default user of `keir` had a UID of 1000 and a GID of 1000. The second user we added was given a UID of 1001.

Note UID and GID information isn't important during everyday use, and most commands used to administer users, groups, and file permissions understand the human-readable names. However, knowing about UIDs and GIDs can prove useful when you're undertaking more complicated system administration.

Adding and Deleting Users and Groups

The easiest and quickest way to add a new user or group is to use the Users and Groups tool under the System ► Administration menu. Of course, you can also perform these tasks through the command line.

Adding and Deleting Users via the GUI

To add a new user, select System ► Administration ► Users and Groups. Click the Unlock button. In the authentication window, supply your password and click Authenticate. Next, click Add User. You'll see the New User Account dialog box, as shown in Figure 29-1.



Figure 29-1. Adding new users and groups is easy with the Users and Groups program.

Fill out the fields on the Account tab, and optionally the User Privileges and Advanced tabs, as follows:

Account tab: As during initial installation (see Chapter 5), you're invited to enter a username as well as a real name. The username is how the user is identified to the system, while the real name is how the user will be identified to other users. Beneath this, you can select the profile you want the user to have—Administrator, Desktop User, or Unprivileged. Users with the Administrator profile can use `sudo` or `gksu` to administer the system. Although Desktop Users can't use these commands, they do have access to most other system resources. The Unprivileged profile removes access to virtually all resources, including external storage devices. Effectively, this is a lock-down account, although such users are still allowed to go online. For most users, the Desktop User profile is a good choice. Below the Profile setting, you can optionally enter contact information. In the Password area, an initial password for the user is required. You can enter it in the text box (and confirm it below) or let the system generate a random password from letters and numbers, but this may be harder for the user to remember.

User Privileges: The settings on this tab offer much more control over what a user can and cannot do on the system. Here, you can prevent users from using certain hardware, such as scanners or modems. You can also control whether the user is able to administer the system. Simply put a check alongside any relevant boxes.

Advanced: Here, you can alter additional settings if you wish. If you're not sure about these parameters, it's best to leave the default settings alone. You might like to change the main group for the user. By default, the user will belong to a newly created group based on the user's own username. For example, if you add the user `john`, he will be added to the group `john`. This private group approach enforces a more stringent policy regarding personal file access. Alternatively, you could create a single group and assign several users to that group for file-sharing purposes. We'll discuss adding and removing groups in the next section.

■ **Caution** Many groups are listed in the Main Group drop-down list. Nearly all of these relate to the way the Linux operating system works and can be ignored. You should never, ever delete any of these groups!

Deleting a user is simply a matter of highlighting the username in the list and clicking the Delete button. Note that the user's `/home` directory won't be deleted. You must do this manually with superuser powers, and it's best accomplished from the command-line prompt (see Chapter 13 for an introduction to basic file-manipulation shell commands).

Creating and Deleting Groups via the GUI

Adding a group is simply a matter of clicking the Manage Groups button in the Users and Groups program window (System ► Administration ► Users and Groups). Don't forget to click the Unlock button, if you haven't already done so. After clicking the Add Group button, you'll be prompted to give the group a name. The GID will be filled in for you automatically, but you could choose a different number if you have good reason to do so. (Remember to use a number above 1000, to keep in line with the way Ubuntu operates.)

It isn't essential that you add users to the group there and then, but a list of users is provided at the bottom of the dialog box. Put a check alongside any user to grant that user access to your group.

Note Bear in mind that users can be members of more than one group, although all users have a main group that they belong to, which has the GID assigned to files they create.

As with user accounts, deleting a group is simply a matter of highlighting it in the list and clicking the Delete button. You should ensure that the group no longer has any members before doing this, because Ubuntu won't prevent you from removing a group that has members (although it will warn you that this is a bad thing to do).

Note Ubuntu appears to offer protection against the havoc caused by deleting a group that is the main group of users on your system. When we deleted an entry that was the main group of a different user, and then logged in as that user, the group was automatically re-created! You shouldn't rely on this kind of protection, however, and should always check before deleting a group.

Adding and Deleting Users and Groups at the Command Line

You can create new users at the command-line shell by using the `useradd` command. This command must be run with superuser powers, which is to say that it must be prefaced with the `sudo` command.

The command to add a user is normally used in the following way:

```
sudo useradd -m <username>
```

The `-m` command option tells the command to create a home directory for the user. Used on its own, `useradd` merely updates system files with the new user's details and nothing else. There are several other useful command options, which can be discovered by a quick

browse of the command's man page. You can't log in to the new user account you have just created until a password is assigned, as discussed in the next section.

Creating a new user this way will automatically create a new group, which will have a title that's exactly the same as the username you just created, and add this user to it.

Adding a new group is just as easy as adding a new user:

```
sudo groupadd <groupname>
```

To specify a different main group when creating a new user, use the `-g` switch:

```
sudo useradd -m -g <groupname> <username>
```

For example, the following command creates a user called `raymond` and adds him to the main group `users`:

```
sudo useradd -m -g users raymond
```

Note that the specified group will need to be created first; it won't be created automatically by the `useradd` command. In this particular example, the group `users` is a standard one, and should already be present on your Ubuntu system.

Although it's easy to do, creating users and groups at the command line is not advised, because there are a handful of annoying issues. One issue is that the new user is assigned the Bourne (`sh`) shell environment, rather than `BASH`, as is the default under Ubuntu. This can be overcome by the user simply typing `bash` at the command line after he has logged in.

Note For a permanent change to the user's shell, edit the `/etc/passwd` file. You'll need administrator powers to do this. Look for the line that begins with the name of the new user, probably at the end of the file, and change the end of the line to read `/bin/bash`, rather than `/bin/sh`. Be *extremely careful* editing this file! It's a central file without which your system could not operate. Ensure you make only the edit described here.

But there's another, more annoying issue relating to groups when you're creating a user account at the command line. Most users are members not only of their own group, but also of several system groups. These groups relate to various hardware and software functions. For example, membership of the `cdrom` group may be required if the user wants to be able to use the CD or DVD-ROM drive.

Strictly speaking, you should add new users to these groups if they're to make full use of the system. These groups are described as *supplementary groups*, as opposed to the main group that is assigned to files the user creates.

Use the `id` command to display user and group information. On our test system, typing the following:

```
id keir
```

revealed the following groups:

```
uid=1000(keir) gid=1000(keir)
groups=1000(keir),4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44
(video),46(plugdev),106(fuse),108(lpadmin),114(admin)
```

All those after the main group, `1000(keir)`, are supplementary groups. For a list of what they do, see Table 29-1.

Table 29-1. *System Groups Within Ubuntu*

Group	Definition
adm	Used for system logging
dialout	Required for use of serial port devices, such as older modems
cdrom	Allows the user to access the CD/DVD-ROM
floppy	Allows the user to access the floppy disk drive
audio	Enables sound output for the user
dip	Required for use of dial-up modems
video	Allows video acceleration for the user
plugdev	Allows the user access to removable storage, such as card readers, digital cameras, and so on
fuse	Allows the user to mount file systems in user space
lpadmin	Allows the user to administer the printer
admin	Gives the user system administration abilities (superuser powers)

As you might have guessed, to manually add a user under Ubuntu, you must not only create a group and then add the user to it, but you must also add that user to the required selection of supplementary groups. Some are mandatory for effective use of the computer, such as `audio`, while others are optional, depending on how much freedom you want to afford the new user.

You can add a new user to supplementary groups by using the `-G` (an uppercase `G`) switch with the `useradd` command. Here’s how to add a new user called `raymond` to the

system so that he is able to make full use of the system (having first created a group called raymond):

```
sudo useradd -m -g raymond -G adm,dialout,cdrom,floppy,audio,dip,video,plugdev,
fuse,lpadmin,admin raymond
```

Additionally, creating a new user using `useradd` won't automatically apply a password to the account. Ubuntu can't work accounts unless they have a password, so until one is applied, the new account will be locked. A user with administrative powers can assign a password using the `passwd` command, as discussed in the next section.

Deleting a user is mercifully simple compared to this! Use the `userdel` command, as follows:

```
sudo userdel <username>
```

This command alone won't remove the user's `/home` directory, however. To accomplish this task, add the `-r` switch to delete the user and the user's `/home` directory, like so:

```
sudo userdel -r <username>
```

Similarly, to delete a group, use the `groupdel` command:

```
sudo groupdel <groupname>
```

Note that you won't be able to remove a group if it's an existing user's main group.

Adding and Changing Passwords

On a default Ubuntu installation, ordinary users are able to change their own password in a terminal window. The command for any user to change her own password is simple:

```
passwd
```

The user will be asked to confirm her current password, and then to enter the new password twice, to make sure that it has been typed correctly.

Alternatively, by adopting superuser powers, a user can change the password of another account:

```
sudo passwd <username>
```

This is necessary after you create a new user account on the command line with `useradd`, because the new user isn't given a password automatically. For obvious security reasons, Ubuntu won't allow blank passwords.

You can enter just about anything as a password, but you should bear in mind some common-sense rules. Ideally, passwords should be at least eight characters long and contain letters, numbers, and even punctuation symbols. You might also want to include both uppercase and lowercase letters, because that makes passwords harder to guess.

A number of command-line options can be specified with the `passwd` command when it is invoked with superuser powers. For example, the `-l` option will lock the specified account so that it can't be accessed (the `-u` option will unlock it).

Tip You can temporarily switch into any user account by typing `su <username>`. When you've finished, simply type `exit` to return to your own account. Remember that typing `su` without a username will give you root powers, so be careful.

Summary

In this chapter, we looked at the principles behind user and group accounts under Ubuntu. We've examined how user and group accounts can be created, edited, and deleted using the GUI, as well as on the command line. We also looked at how passwords can be manipulated by individual users and by an administrator with superuser powers.

In the next chapter, we'll look at how the Ubuntu system can be optimized. You'll also learn about several interesting and important system tools.