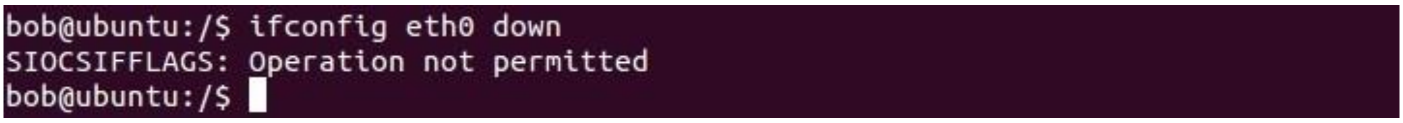
GENERAL ADMINISRTATION IN LINUX (UNIT 4 PART1 AND 2)

The **root user**, also known as the **superuser** or **administrator**, is a special user account in Linux used for system administration. It is the most privileged user on the Linux system and it has access to all commands and files. The **root** user can do many things an ordinary user cannot, such as installing new software, changing the ownership of files, and managing other user accounts.

It is not recommended to use **root** for ordinary tasks, such as browsing the web, writing texts, e.g. A simple mistake can cause problems with the entire system, for example if you mistype a command. It is advisable to create a normal user account for such tasks. If root permissions are needed, the *su* and *sudo* commands can be used.

For example, if we try to bring the **eth0** interface down with an ordinary user, we will get the following message:



```
bob@ubuntu:/$ ifconfig eth0 down
SIOCSIFFLAGS: Operation not permitted
bob@ubuntu:/$ 
```

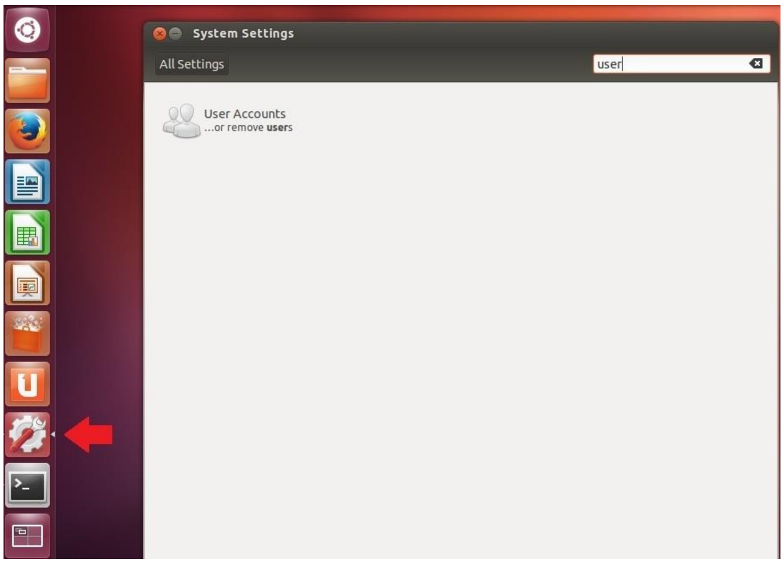
Create users

Every person on your Linux system should have its own account. You need to be familiar with tools that enable you to create, modify and delete user accounts.

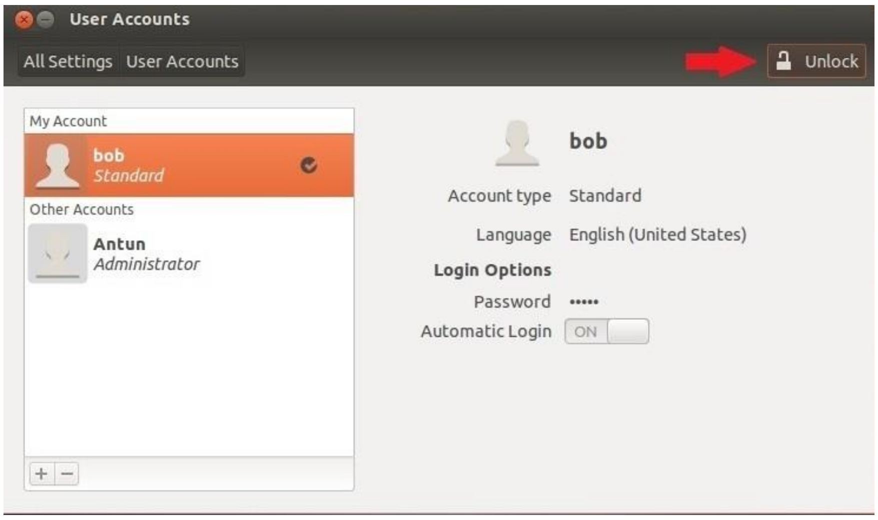
In Ubuntu, you can create users using shell commands or the GUI tool.

Create users using the GUI tool

To create users using the GUI tool, click on the **System Setting** button on the left side of your screen. In the window that opens, type **user**. This should find the GUI tool called **User accounts**:



In the window that opens, click on the **Unlock** button in the upper right corner. You will need to provide the root password.



bob
Standard

Antun
Administrator

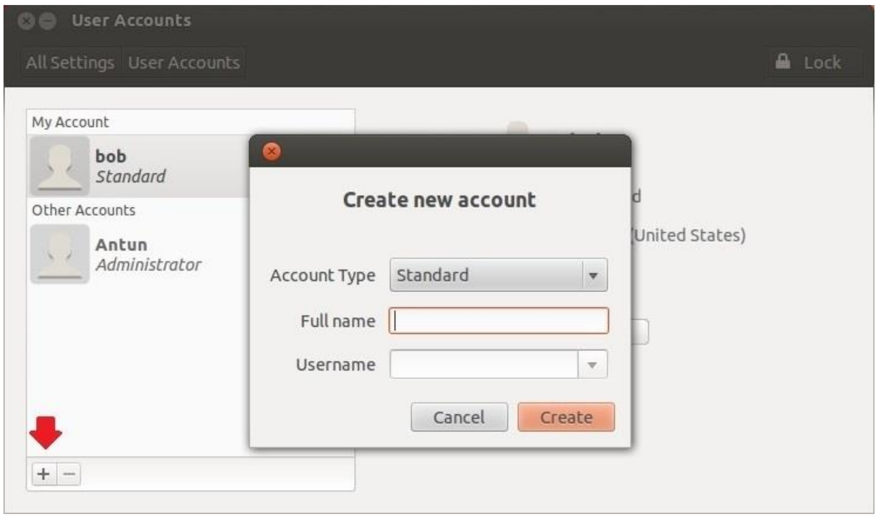
bob

Account type: Standard
Language: English (United States)

Login Options

Password: •••••
Automatic Login: ON

Now, click the plus button in the lower left corner. This opens up a new window:



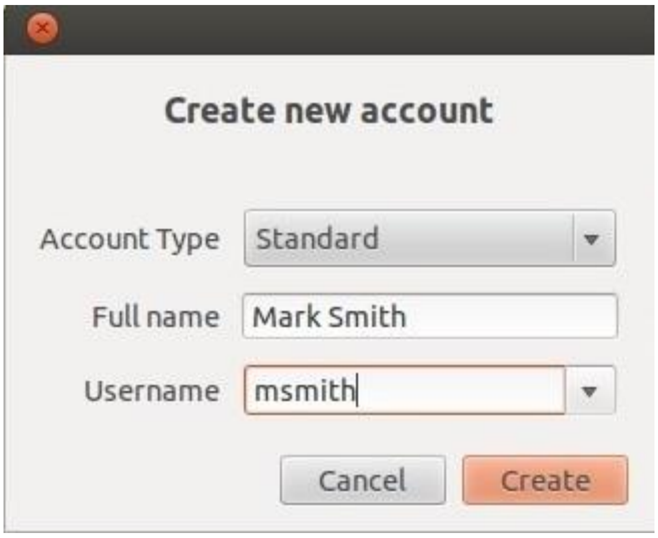
bob
Standard

Antun
Administrator

Create new account

Account Type: Standard
Full name:
Username:

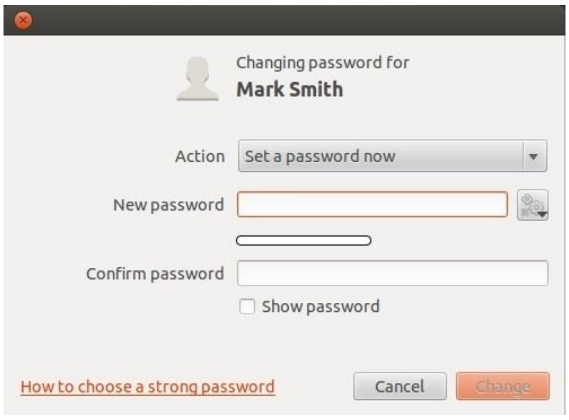
Choose the account type and enter the **Full name** and **Username** of the user. When done, click **Create**.



Create new account

Account Type: Standard
Full name: Mark Smith
Username: msmith

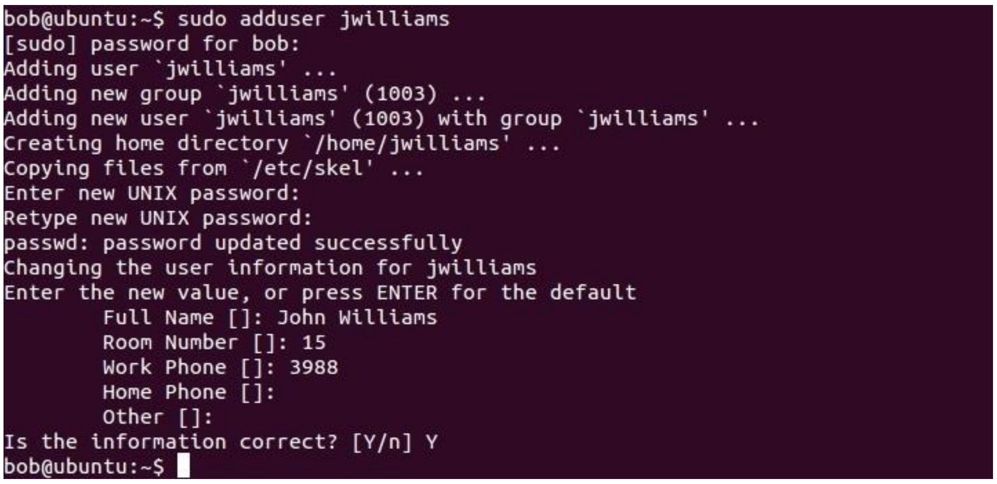
The user is now created, but it is currently disabled. To change that, click on the **Password** field. This opens up a new window:



Type in the password and click **Change**. The user can now log in to the system.

Create users using the *adduser* command

You can also use the shell command *adduser* to create a user. This commands opens up a little wizard that helps to create a user. Here is an example:



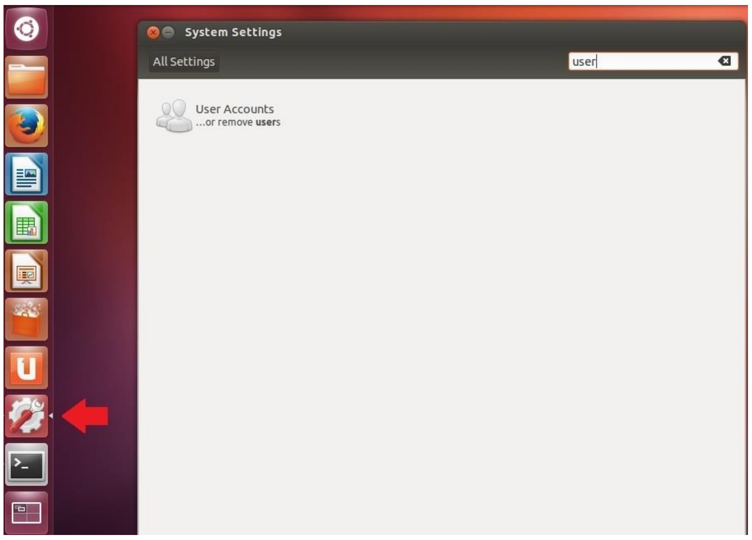
```
bob@ubuntu:~$ sudo adduser jwilliams
[sudo] password for bob:
Adding user `jwilliams' ...
Adding new group `jwilliams' (1003) ...
Adding new user `jwilliams' (1003) with group `jwilliams' ...
Creating home directory `/home/jwilliams' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for jwilliams
Enter the new value, or press ENTER for the default
        Full Name []: John Williams
        Room Number []: 15
        Work Phone []: 3988
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
bob@ubuntu:~$ 
```

Delete users

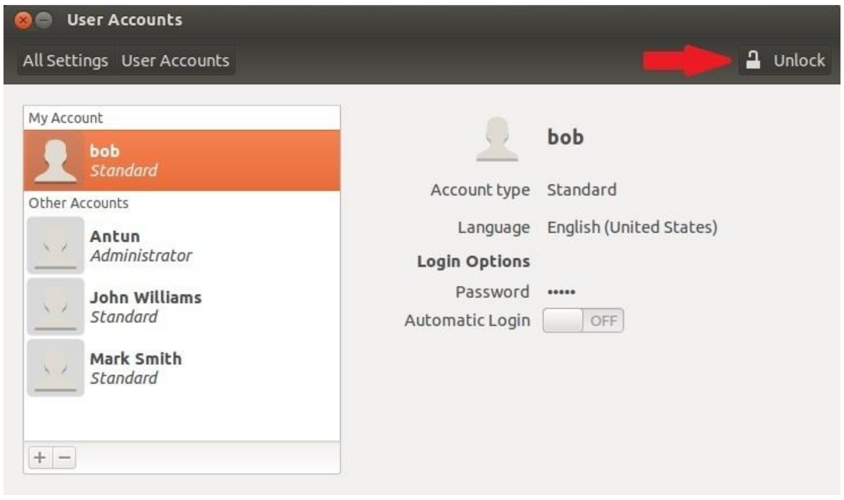
In Ubuntu, you can delete users using shell commands or the GUI tool.

Delete users using the GUI tool

To delete users using the GUI tool, click on the **System Setting** button on the left side of your screen. In the window that opens, type **user**. This should find the GUI tool called **User accounts**:



In the window that opens, click on the **Unlock** button in the upper right corner. You will need to provide the root password.



bob

Account type Standard

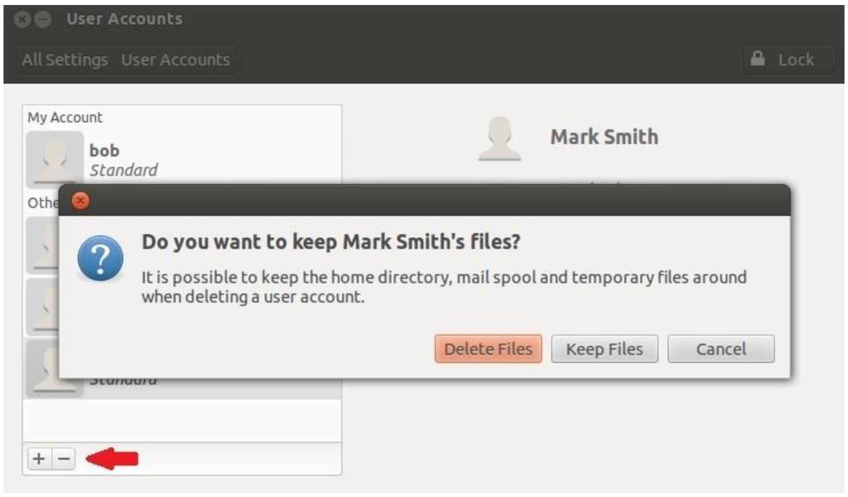
Language English (United States)

Login Options

Password •••••

Automatic Login OFF

Now, click the minus button in the lower left corner. This opens up a new window:



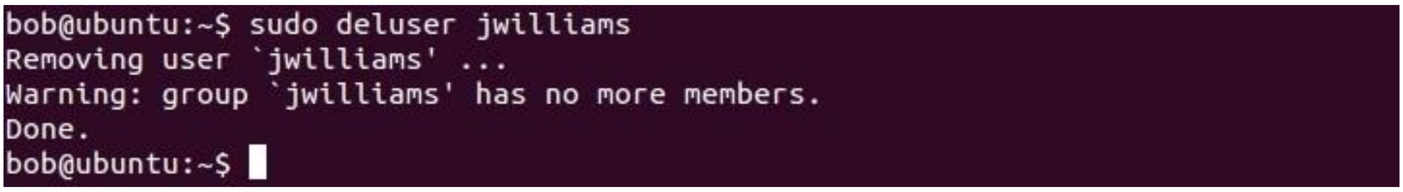
Do you want to keep Mark Smith's files?

It is possible to keep the home directory, mail spool and temporary files around when deleting a user account.

Choose whether you want to keep the user’s files. If you don’t want to keep them, click the **Delete files** button. And that’s it! The user is removed from the system.

Delete users using the *deluser* command

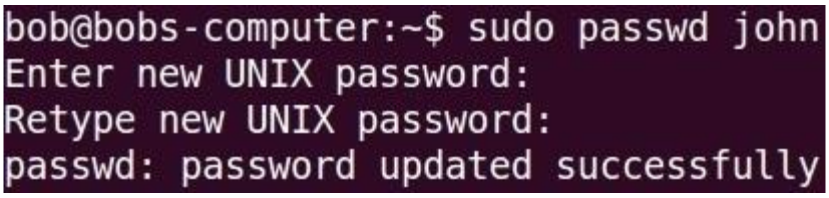
You can use the shell command *deluser* to delete a user. Here is an example:



```
bob@ubuntu:~$ sudo deluser jwilliams
Removing user `jwilliams' ...
Warning: group `jwilliams' has no more members.
Done.
bob@ubuntu:~$ █
```

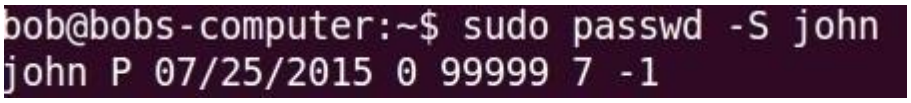
Change passwords

The *passwd* command is used to change passwords of Linux users. To change a password of a specific user account, simply type the *passwd* command, followed by the name of the user whose password you would like to change:



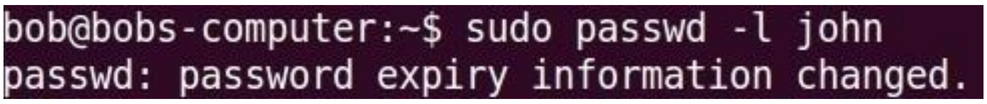
```
bob@bobs-computer:~$ sudo passwd john
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

To view information about the password for a specific user account, use the *-S* option:



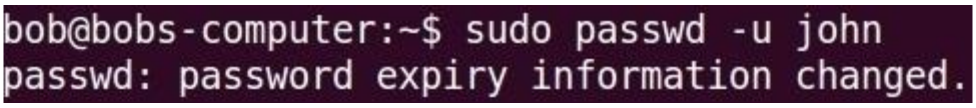
```
bob@bobs-computer:~$ sudo passwd -S john
john P 07/25/2015 0 99999 7 -1
```

To lock an account, you can use the ***-l*** option:



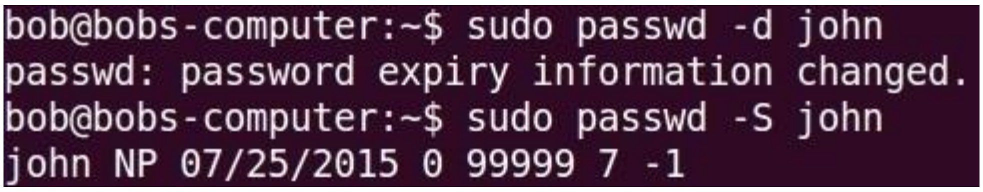
```
bob@bobs-computer:~$ sudo passwd -l john
passwd: password expiry information changed.
```

To unlock an account, use the *-u* option:



```
bob@bobs-computer:~$ sudo passwd -u john
passwd: password expiry information changed.
```

To remove a password from an account, use the *-d* option:



```
bob@bobs-computer:~$ sudo passwd -d john
passwd: password expiry information changed.
bob@bobs-computer:~$ sudo passwd -S john
john NP 07/25/2015 0 99999 7 -1
```

/etc/shadow file format

Most modern Linux distributions use the **/etc/shadow** file to store encrypted password data. Passwords are stored using a **hash** (a one-way type of encryption). This file also stores various password information, such as the date of the last password change, password expiration date, etc.

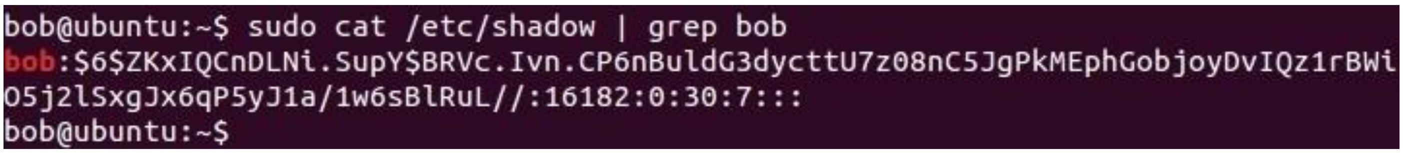
Here is how an entry in the **/etc/shadow** file looks like:

username: encrypted password: last password change: minimum:maximum:warning:disabled:disabled date

Here is a brief description of each field:

- **username** – the name of the user.
- **encrypted password** – the password in encrypted form.
- **last password change** – the date of the last password change. This date is stored as the number of days since **January 1, 1970**.
- **minimum** – the number of days before a password change is allowed. The value of **0** means the password can be changed any time.
- **maximum** – the number of days before the password must be changed. The value **99999** means the user's password never expires.
- **warning** – the number of days before a password is going to expire during which the user will be warned.
- **disabled** – the number of days after a password has expired until the user account is disabled. No entry in this field means that the account is disabled immediately after the password expires.
- **disabled date** – the number of days since **January 1, 1970** that the account has been disabled. No entry in this field means the account is not disabled.

Here is an entry for our user **bob**:



```
bob@ubuntu:~$ sudo cat /etc/shadow | grep bob
bob:$6$ZKxIQCnDLNi.SupY$BRVc.Ivn.CP6nBuldG3dycttU7z08nC5JgPkMEphGobjoyDvIQz1rBWi
O5j2lSxgJx6qP5yJ1a/1w6sBlRuL//:16182:0:30:7:::
bob@ubuntu:~$
```

In the picture above you can see the following information:

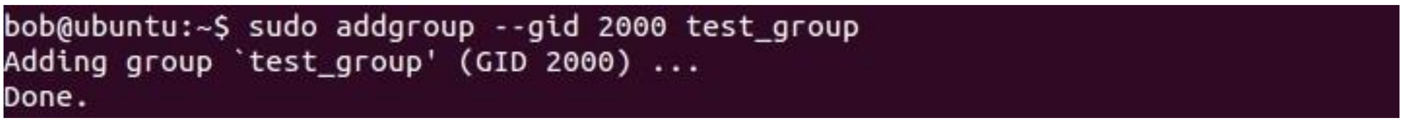
- **username** – bob
- **encrypted password**
- **last password change** – the password has last been changed **16182** days since **January 1, 1970 (April 22, 2014)**.
- **minimum** – **0** means that the password can be changed at any time.
- **maximum** – bob's password expires **30** days after the last password change (**May 22, 2014**)
- **warning** – bob will be warned **7** days before password is going to expire
- **disabled** – no value means that the account is disabled immediately after the password expires
- **disabled date** – no entry in this field means the account is not disabled.

Create groups

To create groups in Linux, you can use a tool called *addgroup*. *addgroup* is a friendlier front-end to the low level tool *groupadd*.

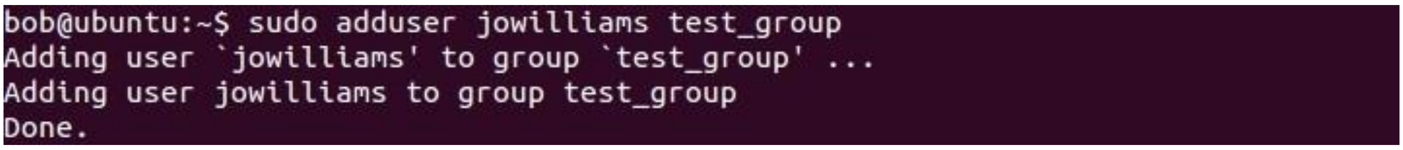
The syntax of this command is simple: you just type *addgroup*, followed by the name of the group. You can also provide a specific **GID (Group ID)** for the group with the *–gid NUMBER* parameter. If you omit this parameter, *addgroup* will use the next available GID.

Let's create a group called **test_group** with the GID of **2000**:



```
bob@ubuntu:~$ sudo addgroup --gid 2000 test_group
Adding group `test_group' (GID 2000) ...
Done.
```

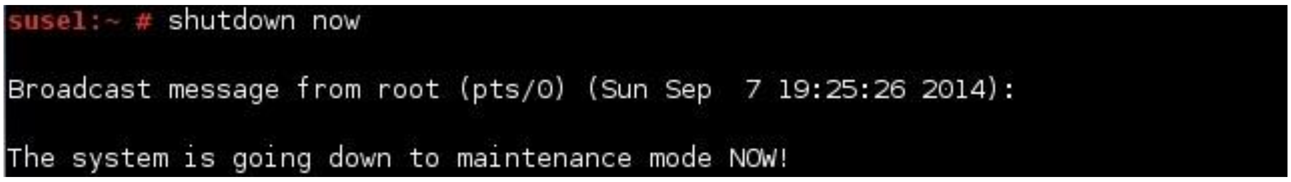
Groups are created with no users. To add a user to the group, use the *adduser* command with two parameters: the username and group. For instance, to add the user **jowilliams** to the group **test_group**, we would use the following command:



```
bob@ubuntu:~$ sudo adduser jowilliams test_group
Adding user `jowilliams' to group `test_group' ...
Adding user jowilliams to group test_group
Done.
```

Shut down the system

To bring your Linux system down in a secure way and prevent users from logging in during the process, you can use the *shutdown* command. You need to specify a time argument, like in this example:



```
susel:~ # shutdown now

Broadcast message from root (pts/0) (Sun Sep  7 19:25:26 2014):

The system is going down to maintenance mode NOW!
```

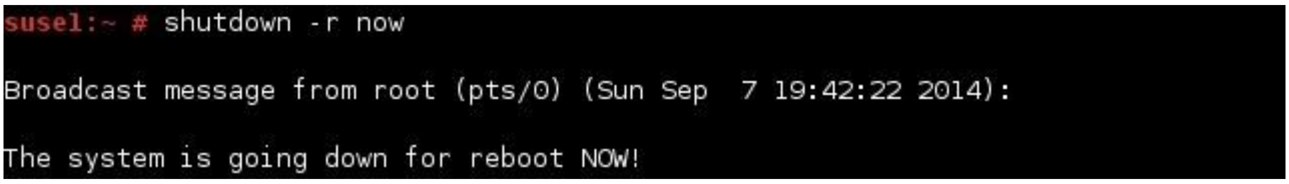
The runlevel **1** is the default, so the command above will put the system into the **single-user mode**.

You can change the default behavior by specifying the extra parameters:

-*r* reboots the system,

-*H* halts it, and

-*P* powers it off. For example, to reboot the system, use the -*r* option:

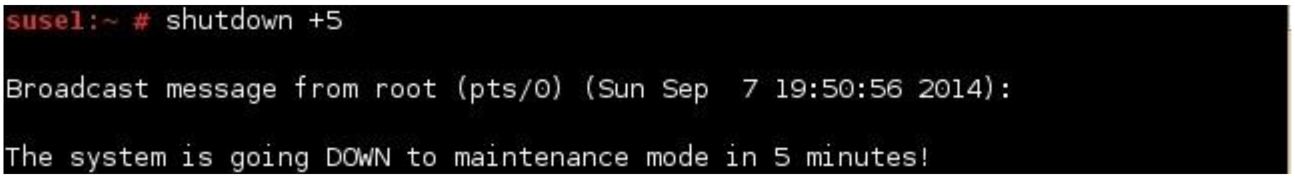


```
susel:~ # shutdown -r now

Broadcast message from root (pts/0) (Sun Sep  7 19:42:22 2014):

The system is going down for reboot NOW!
```

You can specify the time for the shutdown. You can specify the time in the 24-hour clock format (for example, **14:30** for **2:30 p.m.**). You can also specify the number of minutes to wait before the shutdown by using the **+m** format (for example, **+5** to shutdown the system in **5** minutes). Here is an example:



```
susel:~ # shutdown +5

Broadcast message from root (pts/0) (Sun Sep  7 19:50:56 2014):

The system is going DOWN to maintenance mode in 5 minutes!
```