Laboratory 5: Managing Users and Groups

In today's lab we will explore how to work with users and groups in Linux. The lab has three purposes:

- Examination of configuration files used for authentication and database files containing user account information.
- Creation and modification of user and group accounts at the command line.
- Management of users and groups with the **User and Group Management** utility from **Ubuntu Settings**.

1. Managing Users

For users to log into a Linux system, access resources, and perform actions based on their account permissions, the users' accounts must be confirmed as valid by using authentication. **Authentication** is the process of verifying someone's identity by checking the username and password against a central user database. In Linux, the user database consists of two configuration files: `/etc/passwd` and `/etc/shadow`.

1.1 The /etc/passwd File

This file contains the user account information. Each entry has **seven** fields separated by colons. Everyone has permission to view this file's contents. Consider the following example:

amber:x:1000:100:Amber Dawn:/home/amber:/bin/bash

- *amber* → Username: This field contains the name the user enters to log in. Using a combination of uppercase and lowercase letters is possible, but Ubuntu recommends using lowercase letters (some mail exchange protocols do not respond to usernames containing uppercase letters).

- *x* → Password: This field contains the user's encrypted password stored in the `/etc/shadow` file. If there is no `/etc/shadow` file, the actual password is stored in the `/etc/passwd` file.

- *1000* → User identifier (UID): The UID is a unique number representing the username. Regular users typically have a UID greater than 100 because numbers less than 100 are reserved for system accounts (which are nonuser accounts typically used by services, such as www and ftp). The root UID is 0.

- *100* → Group identifier (GID): The GID is a unique number that corresponds to each group. In the `/etc/passwd` file, that group is the user's primary group, which is the group owner for all new files the user creates. Users can be assigned to multiple groups, but they can have only one primary group. More details on group management will be discussed in the managing groups section of this lab.

- *Amber Dawn* → General Electric Comprehensive Operating System (GECOS): Used to store additional user information, such as the user's full name.

- */home/amber* → Home directory: This field lists the absolute path to the user's home directory.
- */bin/bash* → Shell: This field specifies the default shell that starts when the user logs in. user account information.

1.2 The /etc/shadow File

The `/etc/shadow` file is a configuration file containing passwords and their expiration information for all user accounts. It can be read only by the root user. The following example can be used to explain each field:

```
Chloe:x:14743:0:99999:7::
```

- *chloe* → *Username*: This field contains the name the user enters to log in.
- *x* → *Encrypted password*: This field stores the user's encrypted password.
- *14743* → *Last password change*: This field shows the number of days since 1st January, 1970, that the password was last changed. In this example, the number 14'743 translates to 14-05-2010.
- *0* → *Minimum*: This field specifies the number of days before the password can be changed. In this example, the password can be changed at any time.
- *99999* → *Expiration*: This field shows the number of days before the password must be changed. In this example, the user's password "never" expires.
- *7* → *Warning*: This field represents the number of days remaining until the password has to be changed. For example, the user gets a warning message a week (7 days) before his or her password expires.
- *Disable timer*: This field contains the number of days after a password has expired until the user account is disabled. If there is no entry, as in the example, the account is disabled immediately after the password expires.
- *Date disabled*: This field specifies the number of days since 1st January, 1970, that the account has been disabled. No entry in this field means the account is active.

1.3 User Account Information

User information, such as the user's PATH variable, password expiration information, default primary group, location of the home directory and others, is stored in two configuration files: `/etc/default/useradd` and `/etc/login.defs`.

1.3.1 The /etc/default/useradd File

The `/etc/default/useradd` file is a text file that can be read by all users and contains basic parameters that set certain values for new user accounts. The following list describes each value shown in this example of the `/etc/default/useradd` file:

```
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
GROUPS=video
CREATE_MAIL_SPOOL=no
```

- GROUP=100 → This line lists the primary GID. By default, every new user in Ubuntu has the same primary group. To find out which group has a GID of 100, for example, use the command:

  ```
  more /etc/group | grep 100.
  ```
- HOME=/home → This line shows the location of a new user's home directory. In the example, since the specified value is /home, a new user named john has his home directory in /home/john.
- INACTIVE=-1 → This value represents the number of days of inactivity after a password has expired before the account is locked. In the example, the setting -1 disables this feature.
- EXPIRE= → This value specifies the number of days after 1st January, 1970, after which the account will expire. The exact expiration date (in YYYY-MM-DD format) can be inserted instead of calculating the number of days. In the example, omitting a number or date means that the account is set to never expire.
- SHELL=/bin/bash → This line indicates the default shell used when the user first logs in. The /bin/bash entry means all new users use BASH as the default shell.
- SKEL=/etc/skel → This line defines the location of the skeleton directory, which contains files copied automatically to each new user's home directory. Typically, they are hidden files that affect the user environment, such as .bash_history, which records each command you run and .profile, used to enable different language outputs for applications that use this feature.
- GROUPS=video → This line lists the groups a new user is assigned to by default. When assigning users to multiple groups, separate each group name with a comma and no space in between.
- CREATE_MAIL_SPOOL=no — this line specifies whether new users have a mail spool, which is in the /etc/var/spool/mail directory.

A **mail spool** is a temporary storage location where mail is first delivered.

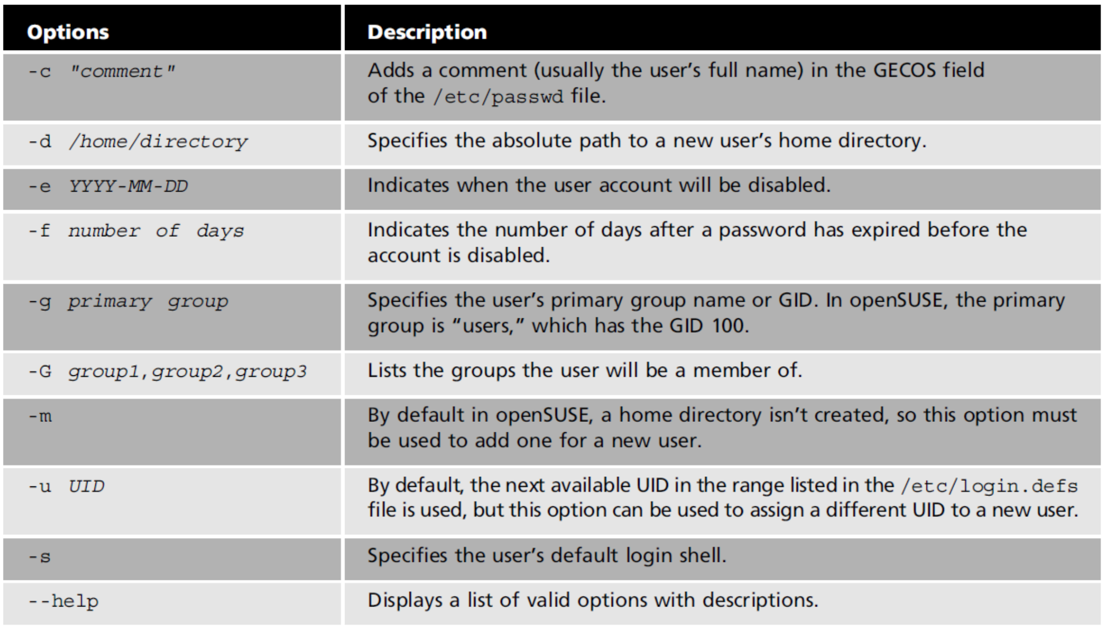
1.3.2 The /etc/login.defs File

The /etc/login.defs file is also a text file that can be read by all users and contains parameters that set certain values for new user accounts, such as default PATH settings, maximum and minimum number of days between password changes, and the range of UIDs and GIDs that can be allocated to user and group accounts.

It is possible to view the contents of your /etc/login.defs file or /etc/default/useradd file with the less command. For instance, less /etc/login.defs displays the contents of the login.defs configuration file.

1.4 Creating User Accounts

The useradd command is used to create user accounts and update default information for new users. Table 5.1 describes the options you can use with this command.



Options	Description
`-c "comment"`	Adds a comment (usually the user's full name) in the GECOS field of the `/etc/passwd` file.
`-d /home/directory`	Specifies the absolute path to a new user's home directory.
`-e YYYY-MM-DD`	Indicates when the user account will be disabled.
`-f number of days`	Indicates the number of days after a password has expired before the account is disabled.
`-g primary group`	Specifies the user's primary group name or GID. In openSUSE, the primary group is "users," which has the GID 100.
`-G group1,group2,group3`	Lists the groups the user will be a member of.
`-m`	By default in openSUSE, a home directory isn't created, so this option must be used to add one for a new user.
`-u UID`	By default, the next available UID in the range listed in the `/etc/login.defs` file is used, but this option can be used to assign a different UID to a new user.
`-s`	Specifies the user's default login shell.
`--help`	Displays a list of valid options with descriptions.

TABLE 5.1. Options used with the *`useradd`* command.

The first thing you must do after creating a user is set the password. The following is an entry in the `/etc/shadow` file for a new user who does not have a password yet:

```
user1:!:14745:0:99999:7:::
```

The *!* character in the encrypted password field means no password has been set for this user. Only the root user can set passwords for new users. The following example shows the root user using the *`passwd`* command to set a password for user1:

```
passwd  user1
```

After issuing the *`passwd`* command, you must enter a new password and then enter it again to confirm. After the password has been set, the user1 entry in the `/etc/shadow` file looks as follows:

```
user1:x:14745:0:99999:7:::
```

Activity 5.1

Creating Users

Approx. Time Required: 15 minutes.

Objective: Create users with the *`useradd`* command and user database files.

In this activity, you will view your own entry in both user database files and will examine the configuration files containing default information for creating users. You will create a user called **student1** and set a password for this account.

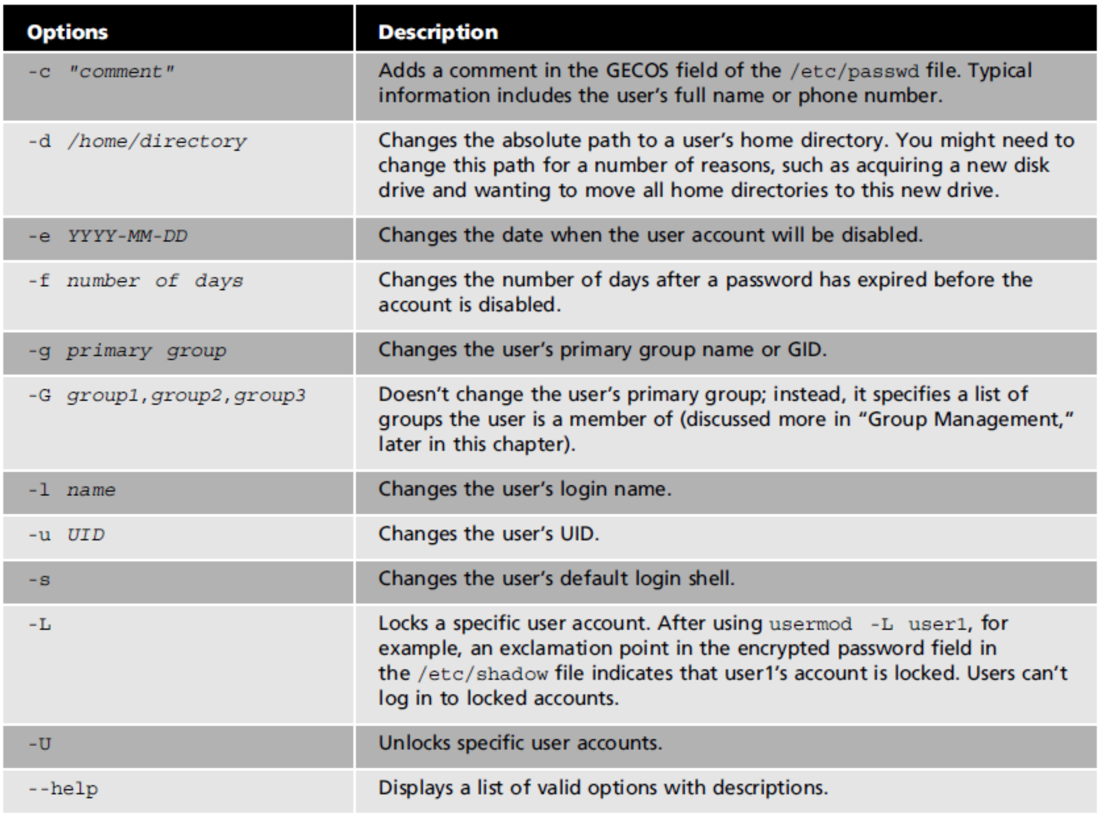
1. Start an Ubuntu virtual machine and open a terminal window.
2. Display the last 10 lines of your password file by typing *`tail /etc/passwd`* and pressing **Enter**. Find the entry for your user account. For instance, the VM created in Lab 1 will have the following information:

```
administrator:x:1000:1000:student,,,:/home/administrator:/bin/bash
```

3. Try to view the `/etc/shadow` file by typing `tail /etc/shadow` and pressing **Enter**. You should be unable to view this file.
4. Type `sudo su` and your password, then press **Enter** to enter the root.
5. To see the last 10 lines of the `/etc/shadow` file, type `tail /etc/shadow` and press **Enter**. Find the user you are currently logged in as and examine the entry for this user account.
6. View the contents of one of your user account configuration files (`login.defs`) by typing `less /etc/login.defs` and then **Enter**. Use the arrow keys (or the mouse wheel) to scroll through this file. Find the password aging settings and compare the entries in this file with the information you viewed in the previous step. Which fields in the `/etc/shadow` file show the same password-setting information as in `/etc/login.defs`? When you are finished examining this file, press `q` to exit the `less` command.
7. The `useradd` file contains basic user account creation information, such as the location of the skeleton directory. To find your skeleton directory (`SKEL=`), type `cat /etc/default/useradd` and press **Enter**. In Ubuntu, the default skeleton directory should be `/etc/skel`. Change to the skeleton directory by typing `cd /etc/skel` and pressing **Enter**.
8. Display a long listing of files and subdirectories in the skeleton directory by `ls -l` and pressing **Enter**. The files and directories you see are the ones copied to a new user's home directory.
9. Review the options for the `useradd` command by typing `useradd --help` and pressing **Enter**.
10. Create a user and add a home directory by typing `useradd -m student1` and pressing **Enter**.
11. Type `tail /etc/shadow` and press **Enter**. Find the `student1` entry and notice that the password field has only the `!` symbol, which means the password has not been set for this user.
12. Set the password for student1 by typing `passwd student1` and pressing **Enter**. Type **`U8un+u`** as the new password and press **Enter**, then enter the password again to confirm.
13. Retype `tail /etc/shadow` and press **Enter**. Find the student1 entry, and confirm that the password field now contains the encrypted password instead of the `!` symbol.
14. Display a long listing of files and subdirectories in student1's home directory by typing `ls -l /home/student1` and pressing **Enter**. Compare the results with the result of Step 8.
15. Type `exit` and press **Enter** to exit the root user account. Type `su student1` and press **Enter** to switch to the student1 user account, and then type **`U8un+u`** as the password and press **Enter**.
16. Now that you have logged in as student1, you can exit this user account and return to your regular user account by typing `exit` and pressing **Enter**.
17. Leave the terminal window open and the virtual machine running for the next activity.

1.5 Modifying User Accounts

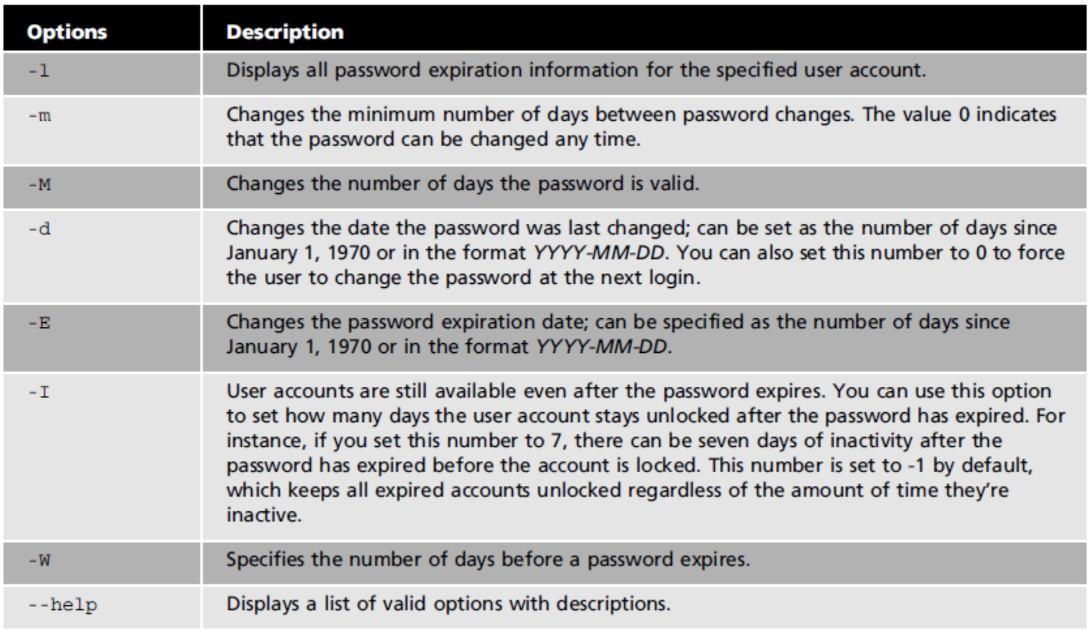
The `usermod` command is used to change user account information, but you cannot change the name, UID, or home directory of a user who is currently logged in (only the root user can issue this command). Table 5.2 describes common options used with the `usermod` command.



Options	Description
`-c "comment"`	Adds a comment in the GECOS field of the /etc/passwd file. Typical information includes the user's full name or phone number.
`-d /home/directory`	Changes the absolute path to a user's home directory. You might need to change this path for a number of reasons, such as acquiring a new disk drive and wanting to move all home directories to this new drive.
`-e YYYY-MM-DD`	Changes the date when the user account will be disabled.
`-f number of days`	Changes the number of days after a password has expired before the account is disabled.
`-g primary group`	Changes the user's primary group name or GID.
`-G group1,group2,group3`	Doesn't change the user's primary group; instead, it specifies a list of groups the user is a member of (discussed more in "Group Management," later in this chapter).
`-l name`	Changes the user's login name.
`-u UID`	Changes the user's UID.
`-s`	Changes the user's default login shell.
`-L`	Locks a specific user account. After using `usermod -L user1`, for example, an exclamation point in the encrypted password field in the /etc/shadow file indicates that user1's account is locked. Users can't log in to locked accounts.
`-U`	Unlocks specific user accounts.
`--help`	Displays a list of valid options with descriptions.

TABLE 5.2. Options used with the *usermod* command.

To list and modify password expiration information for user accounts, the *chage* command is used. It allows the root user to modify information such as the number of days between allowed and required password changes and the expiration date. It also enables the root user to change warning information, such as the number of days before the user's account is set to expire. Table 5.3 describes options used with this command.



Options	Description
`-l`	Displays all password expiration information for the specified user account.
`-m`	Changes the minimum number of days between password changes. The value 0 indicates that the password can be changed any time.
`-M`	Changes the number of days the password is valid.
`-d`	Changes the date the password was last changed; can be set as the number of days since January 1, 1970 or in the format *YYYY-MM-DD*. You can also set this number to 0 to force the user to change the password at the next login.
`-E`	Changes the password expiration date; can be specified as the number of days since January 1, 1970 or in the format *YYYY-MM-DD*.
`-I`	User accounts are still available even after the password expires. You can use this option to set how many days the user account stays unlocked after the password has expired. For instance, if you set this number to 7, there can be seven days of inactivity after the password has expired before the account is locked. This number is set to -1 by default, which keeps all expired accounts unlocked regardless of the amount of time they're inactive.
`-W`	Specifies the number of days before a password expires.
`--help`	Displays a list of valid options with descriptions.

TABLE 5.3. Options used with the *chage* command.

If the *chage* command is used without options, it is in interactive mode, meaning that you will be asked questions on password expiration information, and you can enter new information or accept the current values. The current values are placed between brackets, as shown in the following example:

```
chage ap
Changing aging information for ap.
Minimum Password Age[0]:
```

Notice that the value for the minimum password age is set to *0*, pressing **Enter** keeps *0* as the setting.

All users can use the *chage* command to display their password expiration information, but only the root user can use this command to edit users' expiration information.

Activity 5.2

Modifying User Accounts

Approx. Time Required: 15 minutes.

Objective: Modify user accounts with the *usermod* and *chage* commands.

In this activity, you will change from the command line the home directory, username, and password expiration information within a user account.

1. Switch to the root user by typing *sudo su*, pressing **Enter**, and then entering the correct root user password.
2. Find the student1 password entry in the /etc/passwd file by typing *cat /etc/passwd |grep student1* and pressing **Enter**. It should look like this:

```
student1:x:1001:1001::/home/student1:
```

3. Create a new directory by typing *mkdir /home/student1/scripts* and pressing **Enter**.
4. Change student1's home directory by typing *usermod -d /home/student1/scripts student1* and pressing **Enter**.
5. Now retype *cat /etc/passwd |grep student1* and press **Enter**, and then compare the new entry with the one from Step 2. The new entry should look as follows:

```
student1:x:1001:1001::/home/student1/scripts:
```

6. Change student1's login name by typing *usermod -l student2 student1* and pressing **Enter**.
7. Type *cat /etc/passwd |grep student2* and press **Enter**. Compare the result with that in Step 4. Notice that all the information remains the same (including the home directory) except for the first field, which is the login name. The new student2 entry should look something like the following:

```
Student2:x:1001:1001::/home/student1/scripts:
```

8. Switch to the student2 user by typing *su student2* and pressing **Enter**. Display the expiration information for the student2 by typing *chage student2 -l* and

pressing **Enter** (If requested, enter student2's password **U8un+u**). The results should look as follows:

```
Last password change                                    : Oct 17, 2018
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

9. Switch back to the root user by typing *exit* and pressing **Enter**.
10. Use the *chage* command in interactive mode by typing *chage student2* and pressing **Enter**. To set the minimum password age, type *3* and press **Enter**. To set the maximum password age, type *90* and press **Enter**. Press **Enter** on Last password change Next, you set the password expiration warning. The number *7* in brackets is the default, and if you press **Enter**, this setting is accepted. Press **Enter** to accept the rest of the defaults. When you are done, your screen should look like the following:

```
root@cm3020-client:/home/administrator# chage student2
Changing the aging information for student2
Enter the new value, or press ENTER for the default

Minimum Password Age [0]: 3
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2016-07-06]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [-1]:
```

11. Exit the root user account and return to your regular user account by typing *exit* and pressing **Enter**.
12. Leave the terminal window open and the virtual machine running for the next activity.

1.6 Deleting User Accounts

The *userdel* command is used to delete user accounts and remove all entries from user database files (`/etc/passwd` and `/etc/shadow`). It does not remove the user's home directory; to do this, the *-r* option is required. When a user is deleted, all files s/he owned are then owned by the UID. For instance, if student1's UID is 600 and you delete this user, ownership of this user's files are transferred to the UID 600. If you create a new account with the UID 600, this user then owns all files previously owned by student1.

2. Managing Groups

Groups are helpful for streamlining the process of designating which users can perform certain tasks. For instance, the root user could create a group called ftp and make all users who need to send and receive files via File Transfer Protocol (FTP) a member of this group. Every user is a member of at least one group, called the **primary** group. The root user can then assign an unlimited number of additional groups called **secondary** groups. In Linux, the group database configuration file is `/etc/group`. All the groups a user belongs to and their corresponding group id (GID) can be viewed by using the id command as shown in the example below. The first group listed, i.e. users, is the user's primary group. All other groups are secondary groups and are separated by commas.

```
id uid=1000(student1)  gid=100(users)  groups=33(video),1000(ftp)
```

2.1 The `/etc/group` File

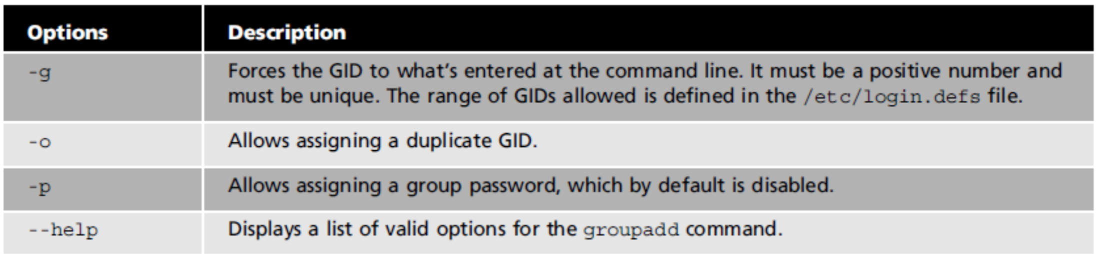
The `/etc/group` file is a configuration file that stores group information and can be read by everyone on the system. The following example shows an entry in this file, followed by a description of each field:

```
video:x:33:student1,student2
```

- **Group name:** This field shows the group's descriptive name rather than the GID.
- **Password:** this field contains the group's encrypted password. Typically, a group password is not used, but if one is set, other users can join the group by using the `newgrp` command.
- **GID:** This unique number represents the user's primary group.
- **List of members:** A list of group members; each member is separated by a comma.

2.2 Creating Groups

The `groupadd` command is used to add a group account. Only the root user has permission to use this command. Table 5.4 describes the options you can use with the `groupadd` command.



Options	Description
-g	Forces the GID to what's entered at the command line. It must be a positive number and must be unique. The range of GIDs allowed is defined in the `/etc/login.defs` file.
-o	Allows assigning a duplicate GID.
-p	Allows assigning a group password, which by default is disabled.
--help	Displays a list of valid options for the `groupadd` command.

TABLE 5.4. Options used with the `groupadd` command.

The newgrp command, shown below, is used to change a user's primary group temporarily; it stays in effect until the user logs out. All new files the user creates will have this new group owner.

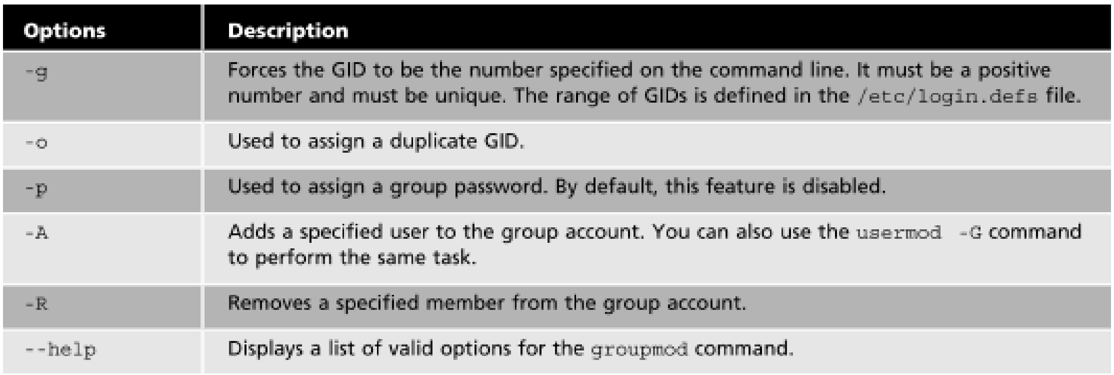
```
newgrp projects
id
uid=1002(student1) gid=1000(projects)  groups=100(users),1000(video)
```

The newgrp command opens a new instance of the BASH shell and changes the user's primary group from users to video.

> The newgrp command opens a new shell even if the command fails. For instance, if you try to use the newgrp command to switch to a group that does not exist, you get an error, but you are still placed in a new shell.

2.3 Modifying Groups

You use the groupmod command to modify existing groups, using values specified at the command line. Only the root user has permission to use this command. Table 5.5 describes the options you can use with the groupmod command.



Options	Description
-g	Forces the GID to be the number specified on the command line. It must be a positive number and must be unique. The range of GIDs is defined in the /etc/login.defs file.
-o	Used to assign a duplicate GID.
-p	Used to assign a group password. By default, this feature is disabled.
-A	Adds a specified user to the group account. You can also use the usermod -G command to perform the same task.
-R	Removes a specified member from the group account.
--help	Displays a list of valid options for the groupmod command.

TABLE 5.5. Options used with the groupmod command.

Activity 5.3

Working with Groups
Approx. Time Required: 15 minutes.
Objective: Create and modify groups with the groupadd, groupmod, and newgrp commands.

In this activity, you will create a group, examine the group database file, and find the new group's entry. You will also add a user to your group, change the user's primary group, and remove a specific user from the group.

1. Switch to the root user by typing `sudo su`, pressing **Enter**, and then entering the correct root user password.
2. View all groups and group members on your system by typing `less /etc/group` and pressing **Enter**. Scroll through the file with the arrow keys or mouse wheel. Press `q` to exit the command.
3. Create a new group by typing `groupadd students` and pressing **Enter**. Display the students group by typing `cat /etc/group |grep students` and pressing **Enter**. You should see the following

`students:x:1002:`

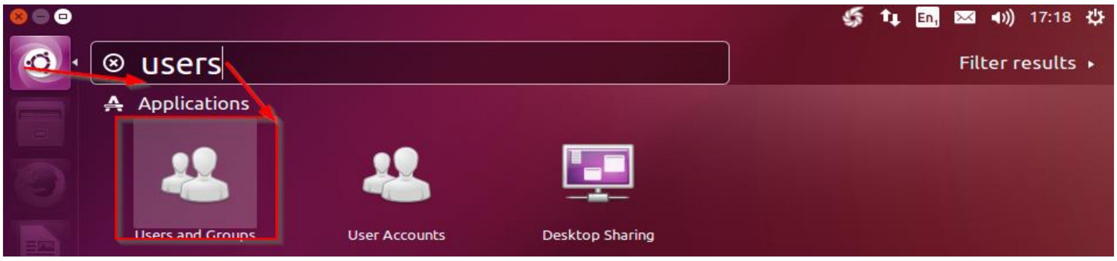
4. Add your regular user account (i.e. administrator) as a member of the students group by typing *`usermod -G students administrator`* and pressing **Enter**.
5. Display the students group again by typing `cat /etc/group |grep students` and pressing **Enter**. The command should output the following:
   `students:x:1002:administrator`
6. Switch to your regular user account by typing *`exit`* and pressing **Enter**.
7. Create a new directory by typing *`mkdir Act5-3`* and pressing **Enter**.
8. Switch to this directory by typing *`cd Act5-3`* and pressing **Enter**.
9. Create an empty file by typing *`touch file1`* and pressing **Enter**.
10. Switch your primary group to students by typing *`newgrp students`* and pressing **Enter**. Then, create a second file by typing *`touch file2`* and pressing **Enter**.
11. View a long listing of files and directories by typing *`ls -l`* and pressing **Enter**. Observe the difference between file1 and file2.
12. Switch to the root user by typing *`sudo su`* and entering the root password. Remove the user you added in Step 4 from the students group by typing *`gpasswd -d administrator students`* and pressing **Enter**.
13. Type *`cat /etc/group |grep students`* and press **Enter**. Notice that now there are no users in the fourth field.
14. Exit the root user account and return to your regular user account by typing *`exit`* and pressing **Enter**.
15. Leave the terminal window open and the virtual machine running for the next activity.

3. Managing User and Group Accounts with User Settings

By default, Ubuntu has the user accounts panel installed under the unity control centre (settings). However, it lacks the extended functionality to edit and manage groups in a GUI. To get around this, we need to install the **gnome-system-tools** package. Open a terminal window and enter the following command:

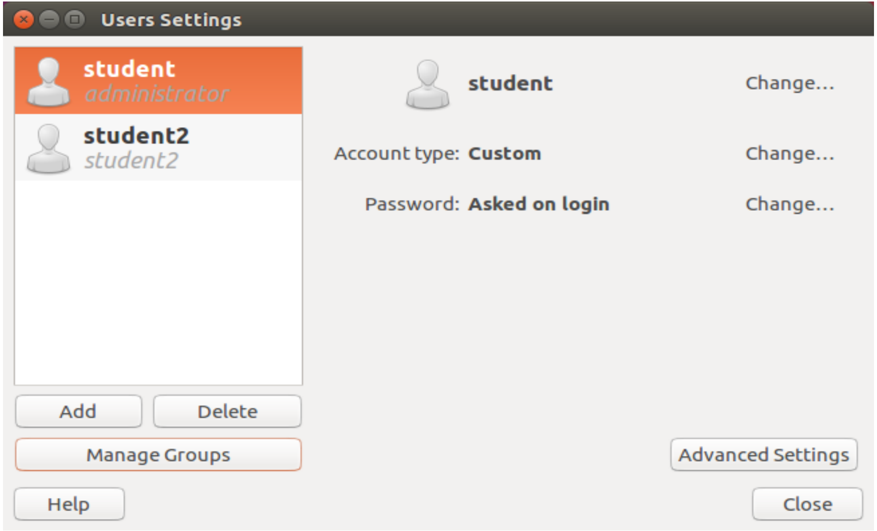
```
sudo apt-get install gnome-system-tools
```

To find the user/group panel go the **Ubuntu DASH**, enter users and select users and groups.



Ubuntu comes with a graphical utility for centrally managing users and groups. Its basic features enable administrators to create and remove users and groups as well as assign home directories, passwords, and automatic logins and default shells.

This utility can be started with the `users-admin` command.



student

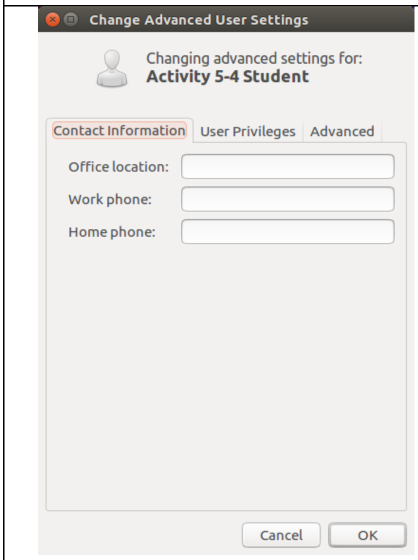
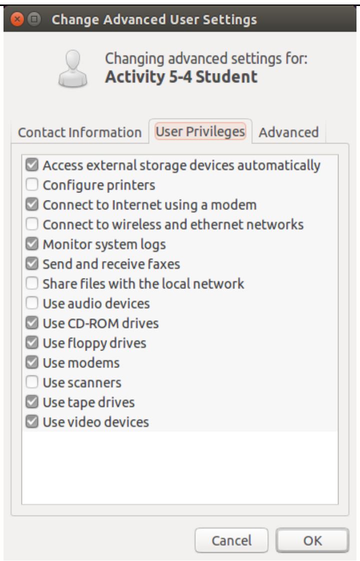
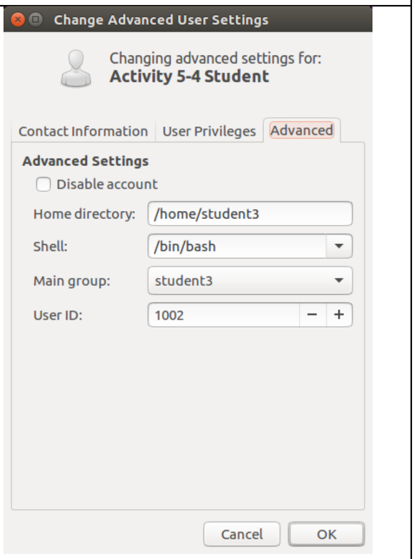
Account type: **Custom**

Password: **Asked on login**

3.1 Adding or Editing User Accounts

When adding or editing a user account with the User Setting Utility, you have four buttons in which you can add or change information:

- **Add:** The add button is used to bring up the created new user panel. This is where you enter the user's full name, username, and password. The username should contain only letters, numbers, and any of the characters ".", "-" and "_"; no other characters are allowed. The encryption scheme currently used for passwords is SHA-512, meaning the password should be between 5 and 84 characters long. The password must be entered two times for confirmation.
- **Manage Groups**: This button opens up a pop-up with groups' settings where you can add a new group, edit the properties of a group, or delete a group. This is where you can add users to specific groups and where users can be put into multiple groups.
- **Delete**: This button deletes the selected user after root authorisation has been granted. When deleting a user, there is an option to keep or delete the files.

Advanced Settings		
 Change Advanced User Settings Changing advanced settings for: **Activity 5-4 Student** Contact Information | User Privileges | Advanced Office location: Work phone: Home phone: Cancel OK	 Change Advanced User Settings Changing advanced settings for: **Activity 5-4 Student** Contact Information | User Privileges | Advanced ☑ Access external storage devices automatically ☐ Configure printers ☑ Connect to Internet using a modem ☐ Connect to wireless and ethernet networks ☑ Monitor system logs ☑ Send and receive faxes ☐ Share files with the local network ☐ Use audio devices ☑ Use CD-ROM drives ☑ Use floppy drives ☑ Use modems ☐ Use scanners ☑ Use tape drives ☑ Use video devices Cancel OK	 Change Advanced User Settings Changing advanced settings for: **Activity 5-4 Student** Contact Information | User Privileges | Advanced **Advanced Settings** ☐ Disable account Home directory: /home/student3 Shell: /bin/bash Main group: student3 User ID: 1002 Cancel OK
Contact information: Used to add additional information for the user such as office location, work phone and home phone.	**User Privileges tab:** Used to assign different privileges to the user such as the use of video devices and access storage devices automatically.	**Advanced Tab:** Used to assign a UID, home directory, login shell, default group, and additional user information such as contact details.

Activity 5.4

Utilising the Users Settings

Time Required: 15 minutes.

Objective: Add users and groups in the gnome system tools User and Group Management utility.

In this activity, you will use the gnome system tools utility to create a user, edit the user account information, create a group, and add the new user to the group.

1. Open the User Settings panel from the Ubuntu Dash by typing **Users** and select **Users and Group** (to access the User Settings utility you can also type **users-admin** in the Terminal).
2. Create a new user by clicking **Add** and then entering the following information:

```
User's Full Name: Activity 5.4 Student
Username: student3
Password: U8un+u
```

> It is possible that the **Users and Groups** interface asks you for the root password. This will NOT be the same as the sudo password for student! In case you want to change the root password, type `sudo passwd root` and enter a new password. It is recommended to use something simple to avoid confusion when typing, for instance "lalala".

3. Create the user account by clicking **OK** at the bottom. To view the user account details, select the user, then click the **Advanced Settings** button and then the **Advanced** tab.
4. Create a new group by clicking the **Manage Groups** button and clicking **Add**. Enter the following information:

```
Group Name: classroom
Group ID (gid): Leave the default information
```

5. In the **Group Members** section, add "Activity 5.4 Student" as a member of the classroom group by clicking its corresponding check box. Confirm the new group by clicking **OK** at the lower right.
6. Save the user and group information by clicking **Close** at the lower right.
7. You can use the command line to make sure everything was done correctly. To do this, open a terminal window, and then switch to the student3 user by typing `su - student3` and pressing **Enter**. Then, type student3's password and press **Enter**. To verify that student3 is a member of the classroom group, type `id` and press **Enter**. You should obtain the following information:

```
uid=1002(student3) gid=1003(student3) groups=1003(student3),1004(classroom)
```