Users and groups management. Configuration files

Ana Cifuentes Romero

DW1E – 11/11/2021

We have to start a root session to do the exercises.

Take into account that user and group names are CASE SENSITIVE (differentiating between capital and lower-case letters).

Root session: Sudo: SUBSTITUTE USER DO AND SU = SUBSTITUTE USER.

Sudo is a computer program that allows the user of a system to access the security privileges of another user, who is usually the administrator (or root user).

With [**sudo**] only one command or function is allowed in the console. Once executed, if you want to run another command or function you have to enter the password again.

With [**su**] that limitation problem disappears. Once executed you remain as administrator until you decide to stop being one. You can, therefore, run everything you need between setting the password and deciding to quit root. To quit root, run the exit command.

On the console, we can tell which user we have by seeing the following symbols:

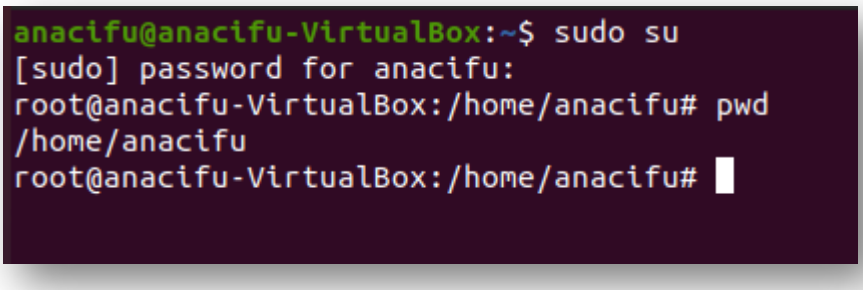
= you are as **root or administrator**. We had to enter the command [su] and the password to stay in that mode.

$ = you're like your **user** and you can't run things like root.

Pwd: current directory.

$ sudo su

 # pwd



/home: It is the directory of standard users, and therefore, intended to store all user files, such as documents, photos, videos, music, templates, etc. It also includes temporary files of applications run in user mode, that serve to save program settings, etc.

Inside /home are the personal directories of all users, named according to the username used.

1. Add two new groups named “daw” and “crey”.

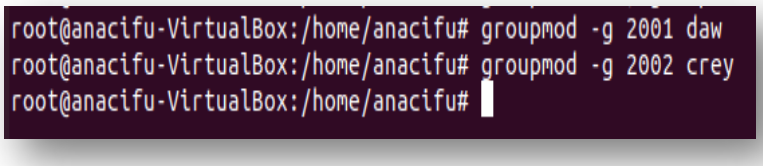
 # groupadd daw; groupadd crey;

To create a group, you need the **“groupadd”** command followed by the **group name**.

You can execute multiple commands in a single command using **«;»** to separate them. For example, Command 1; Command 2; Command 3. Or use **&&** if you only want the following command to be executed when the first command is successful.

2. Change “daw” and “crey” GIDS to 2001 and 2002, respectively.

 # groupmod -g 2001 daw
 # groupmod -g 2002 crey



The **"groupmod"** command allows you to modify the group information. With -g you change the group identifier to GID and followed by the **group name**.

3. Create a new group called “profesores” with GID of 2000. Then, modify the group name to teachers.

 # groupadd -g 2000 profesores
 # groupmod -n teachers profesores

GID of 2000 or Group ID is an identifier that allows to organize users by groups, is a unique data.

The **"groupmod"** command allows you to modify the group information. With -n change the name of the group to group new name.

4. Verify that you have correctly created the groups named “daw”, “crey” and “teachers”.

 # grep 2000 /etc/group; grep 2001 /etc/group; grep 2002 /etc/group

Grep: is a command that is used to find text within the file you are prompted for, followed by the GID we want to see.

/etc: is in charge of storing the configuration files both at the level of components of the operating system itself, as well as of the programs and applications installed afterwards. It is a directory that should contain only configuration files and should not contain binaries.

The **/etc/group** is a text file which defines the groups to which users belong under Linux and UNIX operating system. Under Unix / Linux multiple users can be categorized into groups. Unix file system permissions are organized into three classes, user, group, and others. The use of groups allows additional abilities to be delegated in an organized fashion, such as access to disks, printers, and other

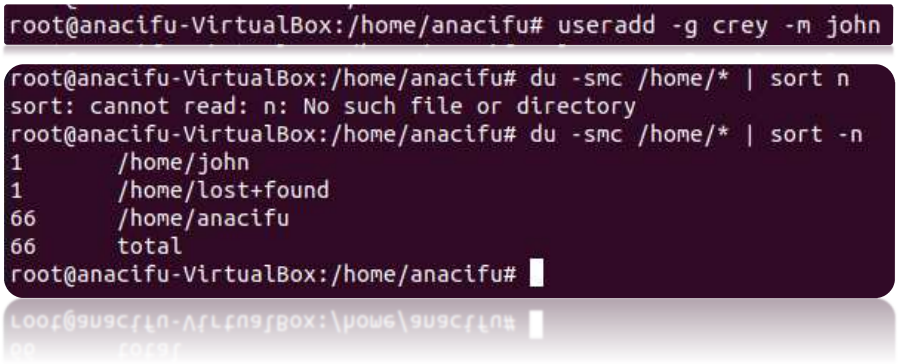
peripherals. This method, amongst others, also enables the Superuser to delegate some administrative tasks to normal users.

Teachers: Group name. Uniquely identifies a user account, and the group uses that username when logging in.

x: Group password. The password stored in the passwd file is encrypted (so an x will be displayed).

2000: group identification code (GID). The Linux system uses GID to identify groups, not group names.

5. Add a new user named “john” whose primary group is “crey”. Has the home directory been created with the default command?
   # useradd -g crey -m john



-g: The group name or number of the user's initial login group. The group name must exist. A group number must refer to an already existing group. Every user has their own **GID** (**Group Identifier**). We can create users with specific group IDs with the **-g** option.

useradd Command: The useradd is a command used for creating a user in any Linux-based operating system. It is a low-level or less secure command for creating a user because it only creates a user until we specify a flag. This command does not create a home directory until a -m flag is specified.

adduser Command: This adduser command is a relatively less complex and more secure command used for creating a user. It automatically sets the home directory and other user settings and saves all the configuration in the /etc/adduser.conf file.

6. Add a new user named “mary”, whose primary group is “daw” and the home directory /home/mary.
   # useradd -g daw -m mary

```
root@anacifu-VirtualBox:/home# useradd -g daw -m mary
root@anacifu-VirtualBox:/home# du -smc /home/* | sort -n
1       /home/john
1       /home/lost+found
1       /home/mary
66      /home/anacifu
66      total
root@anacifu-VirtualBox:/home#
```

   This exercise is the same as the last one above.

   List users by disk usage du -smc /home/* | sort -n

7. Add a new user named “martha”, whose primary group is “teachers”, the home directory /home/martha and belonging to the secondary group “crey”.

 # useradd -g teachers -G crey -m martha

```
root@anacifu-VirtualBox:/home# useradd -g teachers -G crey -m martha
```

grep /etc/group

```
john:x:1001:2002::/home/john:/bin/sh
mary:x:1002:2001::/home/mary:/bin/sh
martha:x:1003:2000::/home/martha:/bin/sh
root@anacifu-VirtualBox:/home#
```

Less /etc/group:

```
crey:x:2002:martha
```

   The “**-G**” option is used to add a user to additional groups. A list of groups containing the user as a member.

8. Add the following names to the users that you have just created:
   a. John= “John Doe”
   b. Mary = “Mary Williams”
   c. Martha = “Martha Jones”
   # usermod -c "John Doe" john
   # usermod -c "Mary Williams" mary
   # usermod -c "Martha Jones" martha

```
root@anacifu-VirtualBox:/home# usermod -c "John Doe" john
root@anacifu-VirtualBox:/home# usermod -c "Mary Williams" mary
root@anacifu-VirtualBox:/home# usermod -c "Martha Jones" martha
root@anacifu-VirtualBox:/home#
```

Usermod is the command to modify users.
The **-c** (or also --comment) option allows you to modify the value of the comment field in the /etc/passwd file.

9. How could you check that you have created all the users with the right primary groups?
   # cat /etc/passwd

```
root@anacifu-VirtualBox:/home# cat /etc/passwd
```

```
john:x:1001:2002:John Doe:/home/john:/bin/sh
mary:x:1002:2001:Mary Williams:/home/mary:/bin/sh
martha:x:1003:2000:Martha Jones:/home/martha:/bin/sh
root@anacifu-VirtualBox:/home#
```

10. Verify if crey and daw groups have martha as a member
    # grep "crey" /etc/group; grep "daw" /etc/group

```
root@anacifu-VirtualBox:/home# grep "crey" /etc/group; grep "daw" /etc/group
crey:x:2002:martha
daw:x:2001:
```

Another way to verify martha groups:

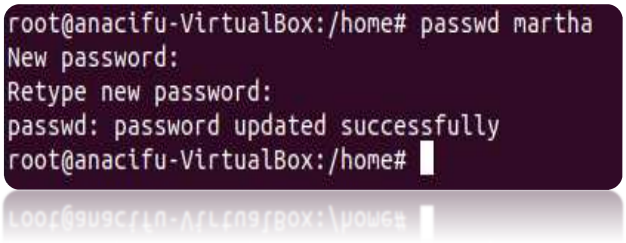
```
root@anacifu-VirtualBox:/home# id martha
uid=1003(martha) gid=2000(teachers) groups=2000(teachers),2002(crey)
root@anacifu-VirtualBox:/home#
```

Grep: is a command that is used to find text within the file you are prompted for, followed by the GID we want to see. With semicolon it is possible to find several users.

The **/etc/group** is a text file which defines the groups to which users belong under Linux and UNIX operating system. Under Unix / Linux multiple users can be categorized into groups. Unix file system permissions are organized into three classes, user, group, and others.

11. Can you log in with any of the users you have created?
    It is not possible because none of them have a password set.

12. Set the password “martha22” for the user “martha”.
    # passwd martha



    The passwd command is the necessary to add the password for every user.

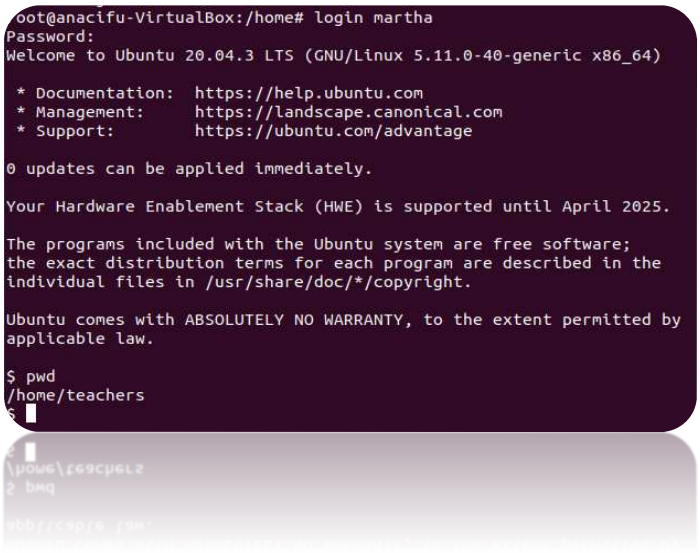
13. Display on the shell prompt the groups to which Martha belongs.
    # grep martha /etc/passwd; grep martha /etc/group

14. Create a directory named “teachers” in “/home”. Then, assign the directory “/home/teachers” to the user martha (you can do all the steps typing just one command).
    # usermod -d /home/teachers -m martha

If used in conjunction with the **-d** and **-m** options, the contents of the old startup directory will be moved to the new startup directory, which is created if it does not already exist.

15. Now, log in as user “martha”. Run the command “cd $SHOME” and check that the home directory is “/home/teachers”.
    # login marta

 # pwd



To login martha you need **login** command and then you must write the password (martha22). After that to see the current directory use the **pwd** command.

16. Go back to the root shell.
    # exit

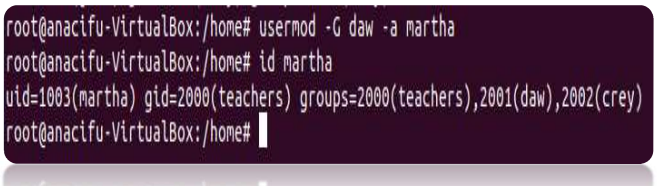
exit root mode.

17. Change the shell of the user named “john” to “sh”
    # usermod -s “/bin/sh” john

-s (shell): Allows you to set a new user login shell. Leaving this field blank causes the system to select the default login shell.

/bin/sh is an executable that represents the system shell.

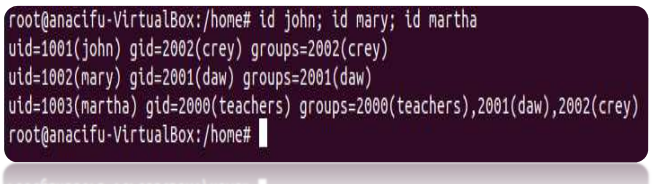
18. Add the user “martha” to the secondary group “daw” without removing the already assigned secondary groups.
    # usermod -G daw -a martha



-a (append): Adds the user to the complementary groups. Use only the -G option.

The -G (or also --groups) option allows us to modify all the additional groups to which the user belongs. If the user is a member of an additional group that is not in the list of new additional groups, the user will be removed from the group.

19. Run a command to print the following information for each user
    # id john; id mary; id martha



User	Primary group	Secondary groups
john	crey	
mary	daw	
martha	teachers	crey, daw

20. Delete all the groups you have created. Could you delete them? Why?
    # groupdel crey; groupdel daw; groupdel teachers;

    It is not possible because there is one user in each primary group.

```
root@anacifu-VirtualBox:/home# groupdel crey; groupdel daw; groupdel teachers;
groupdel: cannot remove the primary group of user 'john'
groupdel: cannot remove the primary group of user 'mary'
groupdel: cannot remove the primary group of user 'martha'
root@anacifu-VirtualBox:/home#
```

21. Delete all the users you have created, including the files and directories inside the home.
    #userdel -r john; userdel -r mary; userdel -r martha;

```
root@anacifu-VirtualBox:/home# userdel -r john; userdel -r mary; userdel -r martha;
userdel: john mail spool (/var/mail/john) not found
userdel: mary mail spool (/var/mail/mary) not found
userdel: martha mail spool (/var/mail/martha) not found
root@anacifu-VirtualBox:/home#
```

    The **userdel** command removes 1 or more users from the system. This is not reversible. Then the name of the users separated by semicolon.

    -r Delete user login directory and all files in the directory

22. Try again to delete the groups.
    # groupdel crey; groupdel daw; groupdel teachers;

```
root@anacifu-VirtualBox:/home# groupdel crey; groupdel daw; groupdel teachers;
```

```
root@anacifu-VirtualBox:/home# id john
id: 'john': no such user
root@anacifu-VirtualBox:/home# id mary
id: 'mary': no such user
root@anacifu-VirtualBox:/home# id martha
id: 'martha': no such user
root@anacifu-VirtualBox:/home#
```

The **groupdel** command is used to delete an existing group. It will delete all entry that refers to the group, modifies the system account files, and it is handled by superuser or root user. Then the names of groups separated by semicolon.