Subscribe to DeepL Pro to translate larger documents.
Visit www.DeepL.com/pro for more information.

Operating systems

Adil ENAANAI

Adil ENAANAI

adil.enaanai@gmail.com
Computer Science Department
Faculty of Science -Tetouan-

Chapter 4

User management under Linux

Adil ENNAANAI

User concept

Linux is a multi-user system, meaning that several users can have an account and work at the same time.

Each user has a certain number of privileges vis-à-vis the system. This means they have access rights to some files, and sometimes not to others.

These rights generally concern reading, writing and executing a file.

Adil ENNAANAI

Account types

Not all user accounts are created equal on Linux. There are three types:

Superuser (root) System

accounts Ordinary

accounts

Adil ENAANAI

Account types

Super-user (root): this is the most important user of the system from an administrative point of view. It is not affected by file access rights. His **UID is** 0 (zero), which makes him unique. This superuser will therefore be in charge of system administration tasks.

System accounts: The system contains a whole series of accounts that are not assigned to individuals (bin, daemon, sync, apache...). These are used to facilitate the management of access rights for certain applications and daemons. For example, if you run the Web server with the "apache" account, you can easily restrict access rights to certain files. UIDs between 1 and 999 are
generally used for these accounts

ADIL ENNAANAI

Account types

Ordinary accounts: All other user accounts are associated with people; their purpose is to enable standard users to log in. A user's **UID** will be a number greater than or equal to 1000.

Adil ENAANAI

Run as Administrator

As this is a system administration task, an ordinary user cannot access these administration rights.
To do this, we often use the **sudo** command (Eng - Substitute User DO), which allows a user to execute commands that can only be used by the superuser. It is used as follows:

$ **sudo** *command*

Your account password is then requested so that the system can verify your identity.

Adil ENHAANAI

Run as Administrator

Mais!!!! It might be tempting to use the superuser (root) as a session, so as not to have to change user when reconfiguring the system.
It's possible (and we'll see how...).

You mustn't do that!!!

In fact, with the superuser, one wrong operation can cause the irremediable loss of all or part of your data or render the machine unusable!

But with **sudo** user substitution, it's more a way of telling you and reminding you: **Are you sure you want to do this?**

Adil ENNAANAI

Open a terminal in root mode

Using **sudo** to run a single command doesn't cause any real inconvenience,
but it can be unpleasant to use it to run a long procedure requiring several interventions in super-user (root) mode.
By opening a terminal in root mode, you can avoid having to call sudo at every stage of this procedure, without having to activate access to the root user account.
The disadvantage of this method is that no trace of the actions performed is recorded in the sudo log (other than the opening of the root terminal itself).
Opening a root terminal is not recommended.

ADIL ENNAANAI

Open a terminal in root mode

To use a root terminal :
Open a terminal window;
Enter the following command: user@computer:~$ sudo -i Enter your password at the password prompt; Run your series of administration commands;
Close the root session: root@computer:~# exit or Ctrl+D

```
ubuntu@ubuntu-VirtualBox:~$ sudo -i
[sudo] Mot de passe de ubuntu :
root@ubuntu-VirtualBox:~# exit
déconnexion
ubuntu@ubuntu-VirtualBox:~$
```

Adil ENAANAI

Creating a user account

The **useradd** command adds a new user without specifying any account information (UID, password, etc.) Syntax :

useradd [options] login

The login defines the name of the account to be created.

For example:

```
ubuntu@ubuntu-VirtualBox:~$ useradd NewUser
useradd: Permission denied.
useradd : impossible de verrouiller /etc/passwd ; veuillez réessayer plus tard.
ubuntu@ubuntu-VirtualBox:~$
```

But why doesn't it work?!!!!

Only the administrator (root) has the right to add a new user

```
ubuntu@ubuntu-VirtualBox:~$ sudo useradd NewUser
```

The user management file

To understand how to configure a user account, we're going to look at how Unix manages these accounts. In Unix, everything is a file.
The first file we'll look at is /etc/passwd, which contains account information.
It's this file that the system consults when you log in to your account by typing your login and password. If what you've typed doesn't exist in this file, you won't be able to log in. This file contains text fields separated by **:** and in the following format:

account_name: password: user_number: group_number :
comment: directory: program_startup

ADIL ENAANAI

The user management file The

meaning of the fields :

- **account name** = user ID
- **password** = user password. This is encrypted
- **user number** = the **UID**. This identifier is unique. Values above 1000 are for user accounts
- **group number** = an integer that identifies the user's group. It's a unique identifier known as a **GID** (Group Identifier).
- **comment** = information about the user.
- **directory** = the directory in which the user finds himself after logging in.
- **command** = the default shell associated with this account.

AOUT ENTAANAI

The user management file

So, if we explore the contents of this file, we'll find quite a lot of information on the various accounts.
For example, the last two and eight lines:

```
ubuntu@ubuntu-VirtualBox:~$ head -2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

```
ubuntu@ubuntu-VirtualBox:~$ tail -8 /etc/passwd
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:Ubuntu,,,:/home/ubuntu:/bin/bash
smi:x:1001:1001:SMI,,,:/home/smi:/bin/bash
sma:x:1002:1002:SMA,,,:/home/sma:/bin/bash
NewUser:x:1003:1003::/home/NewUser:/bin/sh
```

The user management file

We've already noticed the information missing from the newly created **NewUser** account.

And if we explore the **/home** directory, we won't find the **NewUser** account's home folder.

```
ubuntu@ubuntu-VirtualBox:~$ ls /home
sma  smi  ubuntu
```

ADIL ENNAANAI

AddUser command options

Creating an account with configuration options :

 # useradd -u 2040 -g DepInfo -G S1 -c "comment" -e 2016-07-01 -s /bin/bash -d /home/newUser *newUser*

-u : to manually specify the account UID.
-g: to specify the default group.
-G: to specify secondary groups
-c: to assign a comment (exact name, email address...)
-e: to specify an expiration date for this account (after which i t will no longer be accessible)
-s: to specify the default shell
-d: to specify the home directory
-m: to create the personal folder

AGHENAANAI

Modify account properties -usermod-

To modify the properties of an account already created, you can use the **usermod** command, which works in a very similar way to **useradd**.

We make a few changes to the first newUser account:

sudo usermod -c "*this is my account*" -s /bin/bash newUser

Adil ENAANAI

Change default options -useradd-

The second file we're going to study is the one that allows you to change the default options of the **useradd** command.

We can do this at modifying directly the file **/etc/default/useradd** or by using the **useradd** command with the **-D** option.

To display default options: **useradd -D**

```
ubuntu@ubuntu-VirtualBox:~$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Changing default options :

- ❑ to change the home directory
  - **useradd -D -b /home_2**
- ❑ to change the default group
  - **useradd -D -g dev**
- ❑ to change the default shell
  - **useradd -D -s /bin/csh**

The hidden properties file -shadow-

The 3rd file, which always concerns Unix user administration, is **/etc/shadow**. This is a sensitive file (sudo) which contains the words of passwords and other information on the user accounts. Trying to detail its contents:

```
root:!:16719:0:99999:7:::
daemon:*:15630:0:99999:7:::
…
smi:$1$Pv$Omhf8yD/Pg3sViEtYQtHj/:16719:0:99999:7:::
smi1:!:16733:0:99999:7:::
```

As with the passwd file, each field in the shadow file is also separated by a colon ":", and we find the 9 following fields:

Adil ENNAANAY

The hidden properties file -shadow-

1. User name, must be present in /etc/passwd.
2. 13-character encrypted password.
   - A null entry (::) indicates that a password is not required to enter the system.
   - An entry (:!:) indicates that the password has never been initialized, so the account has not yet been activated.
   - An entry (:*:) indicates that the account will not have a password and will not be accessible.
3. The number of days from January 1, 1970 to the day the password was last changed.
4. The number of days before the password can be changed (a 0 indicates that it can be changed at any time).

ACHIENAANAI

The hidden properties file -shadow-

4. The number of days after which the password must be changed (99999 indicates that the user can keep his password unchanged for many, many years).
5. Number of days to notify the user that a password is no longer valid (7 for a full week)
6. Number of days before account deactivation after password expiry
7. The number of days since January 1, 1970 that an account has been deactivated
8. A field reserved for possible future use

Adil ENNAANAI

Change user password -passwd-

The passwd command is used to change a user's password. The user can change his personal password. On the other hand, only the administrator can change someone else's password.

```
ubuntu@ubuntu-VirtualBox:~$ sudo passwd user1
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd : le mot de passe a été mis à jour avec succès
```

Reviewing the effect of this modification in the **/etc/shadow** file:

smi1:6YfrGPySW$soeegfaoVcd1/AeGswS8Q31YsyN5DBghqxAl/wH3OelLXr7ZX6M90bWINvvsPl5YMloa09A9zO7PCLMgRb92E/:16734:0:99999:7:::

The smi1 account is now activated.

Deleting a user -userdel-

The command to delete a user :

Userdel user1

If you also want to delete your personal folder :

userdel -r user1

The previous command deletes the user **user1** and its home directory, but a problem remains: files belonging to user1 and located outside the home directory are not deleted. The following command finds and deletes them using the user's **UID** (assuming the **UID** in the **user1** example is **1002**)

find / -type f -uid 1002 -print -exec rm {} \;

Adil ENNAANAI

Groups : Definitions

A group is a collection of users who can share files and system resources. For example, users working on the same project can form a team. Such a team is traditionally known as a UNIX group.

Each group must have a name, an ID (the GID) and a list of user names belonging to the group. A GID identifies the group internally on the system.

Adil ENAANAI

Groups : Types

The two types of groups to which a user can belong are :

- **Main group**: group assigned by the operating system to files created by the user. Each user must belong to a main group (by default, the user's name is also the name of his main group when created).
- **Secondary groups**: groups to which a user can belong. Users can belong to a maximum of 15 subgroups.

For this purpose, there is a file containing the group names in your system.

Adil ENNAANAI

Groups : Files

The 4th file to examine is the group management file: **/etc/group**. By viewing its contents, we can distinguish the name and GID of the groups present on our system. In particular :

sudo:x:27:ubuntu,smi,sma
smi:x:1000:
sma:x:1002:

Note (from the **passwd** file) that the main user account (in this case **ubuntu**) has the following main group: ubuntu **GID-1000.** However, it belongs to other secondary groups, such as **sudo**, and therefore has administrative rights. We can change this by adding the user **user1** to the **sudo** group **(sudo:x:27:ubuntu,smi,sma,user1)**. However, it's always **best to** change secondary groups using the command :

sudo usermod -a -G sudo user1

AdilENAANAI

Groups : Files

Other files are linked to user and group administration operations (such as /etc/gshadow, ./etc/sudoers...).

It possible to more information on these files using man section 5 of the file.

File	Description	More information
/etc/passwd	User account information	man 5 passwd
/etc/shadow	hidden user account information	man 5 shadow
/etc/group	Defines the groups to which users belong to	man 5 group
/etc/gshadow	Hidden group information	man 5 gshadow
/etc/sudoers	List of who can run what with sudo	man 5 sudoers

Adil ENNAANAY

Groups : Control

- To find out a user's groups, use one of the following commands :

  groups user_name or

  id user_name

- To create a new group :

  groupadd group_name

- To add a user to a group :

  gpasswd -a username *group_name*

- Multiple groups can be added to a user:

  usermod -aG *grp1,grp2* username

- To delete a group :

  groupdel group_name

ADIL ENNAANAI

AddUser command options

Example

```
ubuntu@ubuntu-VirtualBox:~$ groupadd Informatique
groupadd: Permission denied.
groupadd : impossible de verrouiller /etc/group ; veuillez réessayer plus tard.
ubuntu@ubuntu-VirtualBox:~$ sudo groupadd Informatique
[sudo] Mot de passe de ubuntu :
ubuntu@ubuntu-VirtualBox:~$ sudo groupadd Mathematique
ubuntu@ubuntu-VirtualBox:~$ sudo groupadd Physique
ubuntu@ubuntu-VirtualBox:~$ useradd smi1 -m
useradd: Permission denied.
useradd : impossible de verrouiller /etc/passwd ; veuillez réessayer plus tard.
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smi1 -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smi2 -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smpc1 -g Informatique -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smpc2 -g Informatique -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd sma1 -g Mathematique -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd sma2 -g Mathematique -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smia1 -g Mathematique -m
ubuntu@ubuntu-VirtualBox:~$ sudo useradd smia2 -g Mathematique -G Informatique -m
ubuntu@ubuntu-VirtualBox:~$
```

Adil ENNAANAI

Groups : Controls Example

```
ubuntu@ubuntu-VirtualBox:~$ id smi1
uid=1005(smi1) gid=1008(smi1) groupes=1008(smi1)
ubuntu@ubuntu-VirtualBox:~$ id smi2
uid=1006(smi2) gid=1009(smi2) groupes=1009(smi2)
ubuntu@ubuntu-VirtualBox:~$ id smpc1
uid=1007(smpc1) gid=1005(Informatique) groupes=1005(Informatique)
ubuntu@ubuntu-VirtualBox:~$ id smpc2
uid=1008(smpc2) gid=1005(Informatique) groupes=1005(Informatique)
ubuntu@ubuntu-VirtualBox:~$ id sma1
uid=1009(sma1) gid=1006(Mathematique) groupes=1006(Mathematique)
ubuntu@ubuntu-VirtualBox:~$ id sma2
uid=1010(sma2) gid=1006(Mathematique) groupes=1006(Mathematique)
ubuntu@ubuntu-VirtualBox:~$ id smia1
uid=1011(smia1) gid=1006(Mathematique) groupes=1006(Mathematique)
ubuntu@ubuntu-VirtualBox:~$ id smia2
uid=1012(smia2) gid=1006(Mathematique) groupes=1006(Mathematique),1005(Informatique)
ubuntu@ubuntu-VirtualBox:~$
```

Groups : Controls Example

```
ubuntu@ubuntu-VirtualBox:~$ sudo gpasswd -a smi1 Informatique
Ajout de l'utilisateur smi1 au groupe Informatique
ubuntu@ubuntu-VirtualBox:~$ id smi1
uid=1005(smi1) gid=1008(smi1) groupes=1008(smi1),1005(Informatique)
ubuntu@ubuntu-VirtualBox:~$ sudo usermod -g Informatique smi2
ubuntu@ubuntu-VirtualBox:~$ id smi2
uid=1006(smi2) gid=1005(Informatique) groupes=1005(Informatique)
ubuntu@ubuntu-VirtualBox:~$ sudo usermod -g Physique smpc1
ubuntu@ubuntu-VirtualBox:~$ id smpc1
uid=1007(smpc1) gid=1007(Physique) groupes=1007(Physique)
ubuntu@ubuntu-VirtualBox:~$ sudo gpasswd -a smpc1 Physique
Ajout de l'utilisateur smpc1 au groupe Physique
ubuntu@ubuntu-VirtualBox:~$ sudo gpasswd -a smpc2 Physique
Ajout de l'utilisateur smpc2 au groupe Physique
ubuntu@ubuntu-VirtualBox:~$ id smpc2
uid=1008(smpc2) gid=1005(Informatique) groupes=1005(Informatique),1007(Physique)
ubuntu@ubuntu-VirtualBox:~$ id smpc1
uid=1007(smpc1) gid=1007(Physique) groupes=1007(Physique)
```

Groups : Controls Example

```
ubuntu@ubuntu-VirtualBox:~$ sudo deluser smpc2 Informatique
/usr/sbin/deluser: Impossible de retirer un utilisateur de son groupe primaire.
ubuntu@ubuntu-VirtualBox:~$ sudo usermod -g Physique smpc2
ubuntu@ubuntu-VirtualBox:~$ sudo deluser smpc2 Informatique
/usr/sbin/deluser: L'utilisateur « smpc2 » n'est pas membre du groupe « Informatique ».
ubuntu@ubuntu-VirtualBox:~$ id smpc2
uid=1008(smpc2) gid=1007(Physique) groupes=1007(Physique)
ubuntu@ubuntu-VirtualBox:~$ id smpc1
uid=1007(smpc1) gid=1007(Physique) groupes=1007(Physique)
ubuntu@ubuntu-VirtualBox:~$
```

Adil ENNAANAA

Access rights

- Linux has a mechanism for managing access rights. So it's possible for a file owner to give or omit certain rights to other users.

Access rights	On directories	On files
Read (read) (r)	Authorization to view the contents of a directory or sub-directories.	Authorization to view file contents.
Write (w)	Authorization to create, modify or delete files or sub-directories.	Authorization for entities to add, modify or delete the contents of a file.
Run (execute)(x)	Directory access authorization	Enable file execution
(-)	No authorization	No authorization

Adil ENNAANAI

Adil ENAANAI

- r w x r w - r - -

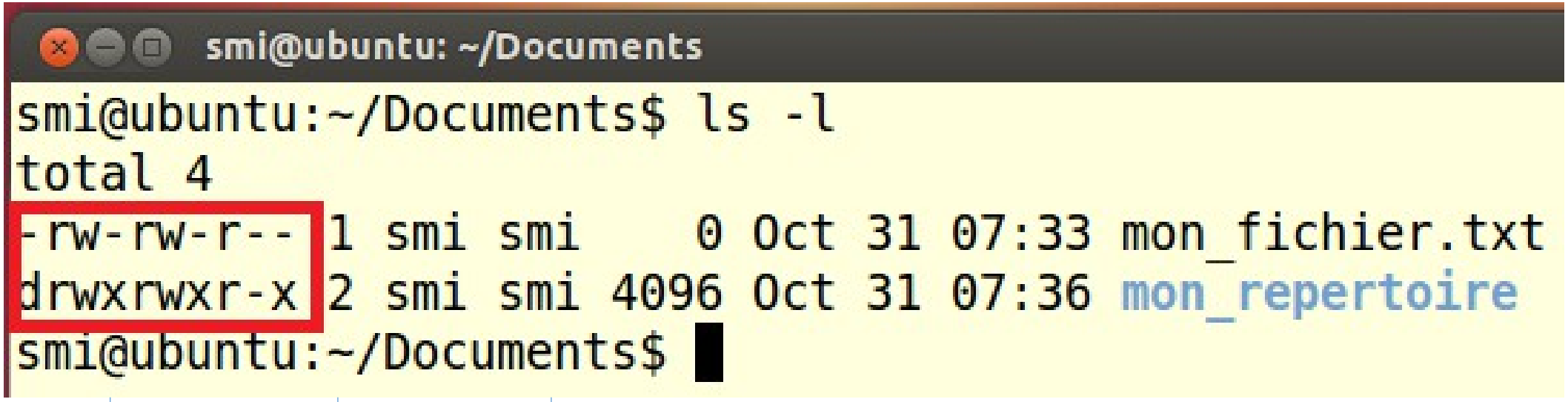
- -

d r w x r w - r

With:

w	Write
x	Execution

Type of file | Rights for owner | Rights for group | Rights for others



Access rights

The following rights are explained:

- r w x r w - r - -

- ✓ The owner user has full rights to the ordinary file.
- ✓ Users in the same group as the owner have read and write rights, but not execute.
- ✓ Users other than those in the owner's group have read-only rights.

ASILENAANA!

Changing access rights - chmod -

The CHMOD (Eng. Change Mode) command

There are two ways to use **chmod**:

- symbolic mode
- absolute mode

Adil ENAANAI

Change access rights - chmod -

CHMOD in symbolic mode

Operator chmod	Meaning	Example	Results
+	Add designated rights to a file or directory	chmod o+wx my_file.txt	Add modification and execution rights to other users
-	Delete the rights assigned to a file or directory	chmod u-x my _file.txt	Removes the owner's right to execute this file
=	Assign these rights exactly	chmod g=r-x my_file.txt	Gives exact read and execute rights to users in the group

Adil ENAANAI

Change access rights - chmod -

CHMOD in symbolic mode

NB: it is possible to group all these modifications into a single command:
chmod o+wx,u-x,g=rx my_file.txt

Examples

To add execution rights to the owner (User)

Chmod u+x file

To remove writing rights from group users

Chmod g-w file

To add read and execute rights to Others

Chmod o+rx file

To assign read and execute rights to the owner (User)

Chmod u=r-x file

AGHENAANAI

Change access rights - chmod -

CHMOD in absolute mode

The second way of assigning access rights via chmod is to use numbers for each set of rights. We can calculate these numbers if we remember the simple way of transforming a number in binary base to a decimal base. So if we have these simple rules to remember:

- (r, w, or x) is represented by 1. (-) is represented by 0.
- We assume that the 3 bits obtained are in binary, then calculate the equivalent number in decimal.

Adil ENNAANAI

Change access rights - chmod -

CHMOD in absolute mode

Examples

-r w x r w - r - -

1 1 1 1 1 0 1 0 0

7 6 4

To assign these rights in absolute mode

Chmod 764 file

Adil ENNAANAI

Change access rights - chmod -

CHMOD in absolute mode

Application exercise

Explain the following access commands

Chmod 012 file

Chmod 234 file

Chmod 345 file

Chmod 567 file

Chmod 000 file

Chmod 785 file

Chmod 002 file

Chmod 234 file

AGH ENNAANAI

Rights automatically assigned to a file

When a new file is created, it is automatically granted certain rights. These are defined by default in the session settings file: **/etc/pam.d/common-session**.

It is possible to define rights to files and directories when they are created. We use the **umask command** (Eng - user file creation mode mask).

Adil ENNAANAI

Rights automatically assigned to a file

First, to find out which mask is used by default, we type : **umask**
If you want to display the default rights in symbolic mode :
umask -S

```
ubuntu@ubuntu-VirtualBox:~/rep1$ umask
0022
ubuntu@ubuntu-VirtualBox:~/rep1$ umask -S
u=rwx,g=rx,o=rx
ubuntu@ubuntu-VirtualBox:~/rep1$ touch fichier1
ubuntu@ubuntu-VirtualBox:~/rep1$ mkdir répertoire1
ubuntu@ubuntu-VirtualBox:~/rep1$ ls -l
total 4
-rw-r--r-- 1 ubuntu ubuntu    0 oct.  22 10:26 fichier1
drwxr-xr-x 2 ubuntu ubuntu 4096 oct.  22 10:27 répertoire1
ubuntu@ubuntu-VirtualBox:~/rep1$
```

Adil ENAANAI

Rights automatically assigned to a file Explanation

Default permission = Initial permission - Mask (022)

NB

The initial file permission is 666

The initial permission of a directory is 777

Examples

To find permission file default:

666rw- rw- rw-

022--- -w- -w-

644rw- r-- r--

To find permission directory default:

777rwx rwx rwx

022--- -w- -w-

755rwx r-x r-x

Adil ENNAANAI

Rights automatically assigned to a file Permanent

change

To change the default rights **permanently**. You need to:

1. Edit /etc/pam.d/common-session

2. Find the ***optional session*** line ***pam_umask.so***

3. Change line to

4. ***session optional pam_umask.so*** *umask=mask_value*

5. Save and edit

Adil ENNAANAI

Change of owner and group

Commands for changing the owner and group of a file

The **chown** command is used to change the owner of a file. For security reasons, only **root** can change the owner of a file or directory.

The **chgrp** command allows you to change the group for the files or directories listed, provided that the user is part of the new group and owns these files or directories.

Adil ENNAANAI

Change of owner and group The chgrp

command

chgrp is used to change the file or directory group. The group can be changed by :

- The root
- The owner of the file, if this is one of the members of the group in question.

Syntax : **chgrp** [options] new_group file/directory

Interesting options :

-A: Change authorization for files in subdirectories of the directory in question.

-c: Change authorization for each file.

Adil ENAANAI

Change of owner and group The chgrp

command

Example

```
ubuntu@ubuntu-VirtualBox:~$ sudo chgrp Informatique fichier1
[sudo] Mot de passe de ubuntu :
ubuntu@ubuntu-VirtualBox:~$ ls -l fichier1
-rw-r--r-- 1 ubuntu Informatique 0 oct.  22 16:33 fichier1
ubuntu@ubuntu-VirtualBox:~$
```

Adil ENNAANAI

Change owner and group The chown

command

chown is used to change the owner and/or owner group of a file or directory.

The syntax of this command :

chown [-option] [user][:group] file [file1 file2 ..]

It can be used to change :

- Owner and group, Owner only, Group only (becomes similar to **chgrp**)

Interesting options :

-R: Recursively modifies all sub-directories and sub-files.

Adil ENNAANAI

Change owner and group The chown

command

Example

```
ubuntu@ubuntu-VirtualBox:~$ sudo chown :Mathematique fichier1
ubuntu@ubuntu-VirtualBox:~$ ls -l fichier1
-rw-r--r-- 1 ubuntu Mathematique 0 oct.  22 16:33 fichier1
ubuntu@ubuntu-VirtualBox:~$ sudo chown smi1:Informatique fichier1
ubuntu@ubuntu-VirtualBox:~$ ls -l fichier1
-rw-r--r-- 1 smi1 Informatique 0 oct.  22 16:33 fichier1
ubuntu@ubuntu-VirtualBox:~$
```

Adil ENNAANAI

ACL (Access Control List)

Access management limits

We have seen that by default the Unix system attaches three privileges to a file:

- Owner user privileges (u).
- Those of the owner group (g).
- Those of others (o).

This means that file sharing can only take place via the group principle. Imagine the following situation:

ADIL ENAANAI

ACL (Access Control List)

Access management limits

For a user X to make a file readable and writable by a user Y, and only by Y, there is no other solution than to create a group G to which X and Y belong, and then grant read and write rights to the file to group G.

Adil ENAANAI

ACL (Access Control List)

Access management limits

The problem here is that group creation is only possible for the system administrator, who alone can add a new user group.

In addition, if another user Z wished to join the project, a request would have to be made again to the administrator to add the new Z member to group G.

Adil ENAANAI

ACL (Access Control

Introduction to ACLs

Here, we'd like to introduce you to a few commands (ACLs) that enable you to set privileges more finely and independently.
But there are two prerequisites for working with LCDs:

- The kernel must support ACLs.
- The file system is mounted with the ACL option

Please note that ACLs can only be used if the kernel supports them (otherwise, you'll have to recompile the kernel...).
On Ubuntu, the kernel supports ACLs, but they are not natively enabled.

Adil ENNAANAI

ACL (Access Control List)

Introduction to ACLs

Check the kernel configuration with this command run in super-user mode:

```
# grep ACL /boot/config-*
```

The following line indicates that general LCD support is present:

CONFIG_**_FS_POSIX_ACL=y**

the stars are replaced by the file systems supported by the ACLs (examples: EXT2,EXT3,EXT4,JFS,...).

ACH ENAANAI

ACL (Access Control List)

Introduction to ACLs

```
ubuntu@ubuntu-VirtualBox:~$ sudo grep ACL /boot/config-*
/boot/config-4.15.0-29-generic:CONFIG_EXT4_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_REISERFS_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_JFS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_XFS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_BTRFS_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_F2FS_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_TMPFS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_HFSPLUS_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_JFFS2_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_NFS_V3_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_NFSD_V2_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_NFSD_V3_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_NFS_ACL_SUPPORT=m
/boot/config-4.15.0-29-generic:CONFIG_CEPH_FS_POSIX_ACL=y
/boot/config-4.15.0-29-generic:CONFIG_CIFS_ACL=y
```

ACL (Access Control List)

Introduction to ACL

Your file system is type: EXT4, so normally ACLs are supported and enabled, but we still can't modify them, to do this we need to install the **acl** package which includes the two commands needed for this modification, by typing the following command:

 # apt-get install acl

NB:

1- This may require an Internet connection

2- Sometimes it is necessary to remount partitions with the acl option

Adil ENNAANAI

ACL (Access Control

Introduction to ACLs

With the acl package installed, two new commands are available for managing ACLs under the three ACL types. These two commands are :

setfacl (Eng - set file's ACL): set these authorizations.
getfacl (Eng - get file's ACL) examines file authorizations.

Adil ENAANAI

ACL (Access Control List) ACL

allocation

Syntax:

setfacl -m u:user:file permissions

Allows you to modify (this is the purpose of option m) the user's permissions (argument u:). Permissions can be defined using symbolic notation on the file.

-m, --modify - modify the ACL of a file or directory

-x, --remove - deletes ACL entries

-b, --remove-all - deletes all ACL entries

-R, --recursive - recursive application of ACLs See the man setfacl manual for more details.

Adil ENNAANAI

ACL (Access Control List)

Examine the permissions associated with a file

Syntax:

getfacl [option] file

The getfacl command displays the file's ACLs. The few options :

-R lets you view ACLs recursively.

-L to follow symbolic links.

Adil ENAANAI

ACL (Access Control List)

Example

```
ubuntu@ubuntu-VirtualBox:~$ getfacl fichier1
# file: fichier1
# owner: smi1
# group: Informatique
user::rw-
group::r--
other::r--

ubuntu@ubuntu-VirtualBox:~$ setfacl -m u:smi1:rwx fichier1
setfacl: fichier1: Opération non permise
ubuntu@ubuntu-VirtualBox:~$ setfacl -m u:smi1:rwx Exercice1
ubuntu@ubuntu-VirtualBox:~$ getfacl Exercice1
# file: Exercice1
# owner: ubuntu
# group: ubuntu
user::rw-
user:smi1:rwx
group::r--
mask::rwx
other::r--
```

AOUFENAANAI

ACL (Access Control List)

Example

In the same way, ACL rights can be assigned to a group, by replacing the u with g

```
ubuntu@ubuntu-VirtualBox:~$ setfacl -m g:Mathematique:rwx Exercice1
ubuntu@ubuntu-VirtualBox:~$ getfacl Exercice1
# file: Exercice1
# owner: ubuntu
# group: ubuntu
user::rw-
user:smi1:rwx
group::r--
group:Mathematique:rwx
mask::rwx
other::r--
```

Adil ENNAANAI

ACL (Access Control List)

Example

If we want to remove smi1's ACL rights to this file: With this command: **setfacl** -x u:smi1 file
And he returns to being a normal user subject to the usual rules.

You can also remove all ACL rights from a file

setfacl -b file

Adil ENNAANAI

End of chapter 4

Adil ENAANAI