

AES. Systèmes d'exploitation

I/ Rôle d'un système d'exploitation

1/ Principe d'un système d'exploitation

Un **système d'exploitation** (en anglais *Operating System*) est un programme exécuté au démarrage d'une machine. Il permet de **gérer** les **fichiers**, les **répertoires**, les **processus**, les **périphériques** (à l'aide de pilotes, *driver* en anglais) en proposant des outils pour cela. Les principaux OS sont Windows, MacOS et GNU/Linux.

Un fichier avec sa gestion est un concept abstrait. Pour le mettre en œuvre physiquement, chaque **OS** utilise un **format d'enregistrement** avec un système d'adresses, comme NTFS pour Windows qui met à la disposition de l'utilisateur l'outil « Explorateur de fichiers » pour copier, déplacer, supprimer un fichier ou un répertoire.

Le système d'exploitation permet aussi de gérer l'authentification de chaque utilisateur ainsi que de ses droits d'accès aux différents fichiers (lecture/exécution – écriture – suppression) à l'aide de **sessions**. On distingue en particulier les **comptes administrateurs** (avec tous les droits) des comptes **utilisateurs** (avec des droits variables).

2/ Organisation du disque dur

A la sortie d'usine, un disque dur est formaté. Il s'agit d'un **formatage** de bas niveau qui organise le disque en différentes parties appelées pistes, cylindres et secteurs. Ces parties vont permettre de localiser les informations inscrites sur le disque. Ensuite, lors de l'installation d'un OS, une **partition** de celui-ci peut être organisé (c'est même conseillé) pour le séparer des autres programmes ou pour pouvoir installer un autre OS.

Avant de pouvoir installer un OS, un formatage de haut niveau est effectué. Il permet de regrouper les différents secteurs du disque dur en blocs (en anglais *cluster*). Le résultat est un **système de fichiers particulier** au **système d'exploitation**. La liste des clusters est inscrite dans une table, appelée la FAT (*File Allocation System*).

On ne peut pas avoir deux fichiers enregistrés sur le même cluster, ainsi, deux fichiers de 1 ko chacun occuperont deux clusters différents (soit 8 ko sur le système NTFS de Windows où chaque cluster occupe 4 ko). Autre exemple, un fichier de 5 ko occupera deux clusters soit 8 ko.

Le premier secteur du disque dur est le **secteur de démarrage** appelé **MBR** (*Master Boot Record*). C'est là qu'est inscrit le code qui va faire démarrer le système ainsi que des informations relatives au disque dur comme le nom du fabricant, le numéro de série, les partitions éventuelles etc.

3/ Fonctions d'un système d'exploitation

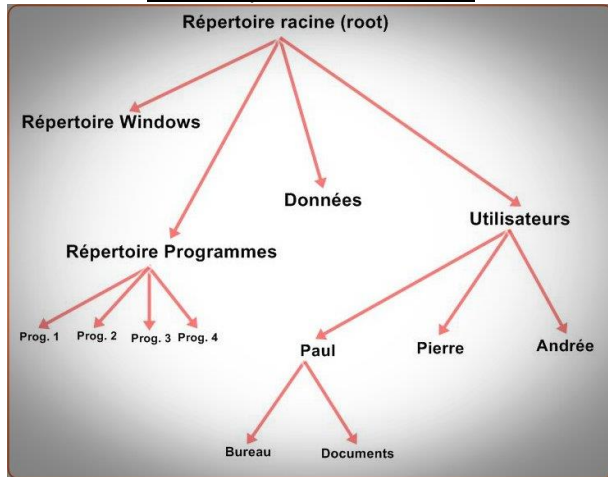
Un système d'exploitation propose depuis de nombreuses années un environnement graphique convivial afin que l'utilisateur puisse sélectionner les actions qu'il souhaite faire accomplir par la machine.

Un explorateur de fichiers permet par exemple de gérer les fichiers et les répertoires. Il présente, sous forme **d'arborescence**, l'organisation de l'ensemble des données.

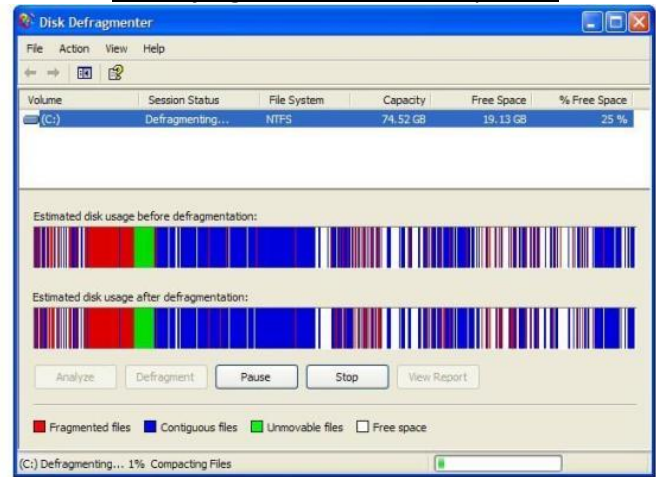
A noter : dans la partie « IHM », il y a un cours sur le parcours des arborescences « IHM.Fiches.URL »

Un fichier, en particulier s'il a une taille importante, peut être découpé en plusieurs parties qui sont enregistrées à différentes parties du disque dur. L'utilisateur n'a pas besoin de le savoir mais cela peut conduire à une **fragmentation importante** du disque dur et ralentir les processus. Cela se produit à force d'installer / désinstaller des logiciels par exemple.

Un exemple d'arborescence



Une défragmentation d'un disque dur



Les fonctions d'un OS sont nombreuses. Elles sont accessibles à l'aide d'outils comme le panneau de configuration, le gestionnaire de périphériques, le gestionnaire de tâches pour Windows. Elles permettent de gérer le fonctionnement de tous les éléments constituant le système informatique.

Un résumé ici : <https://www.youtube.com/watch?v=YScMI8lsy9s>

Auteur : codeur-pro, durée : 7 min 55 sec

En savoir plus sur les systèmes d'exploitation : <https://www.youtube.com/watch?v=SpCP2oaCx8A>

Auteur : pixeesScienceparticipative, durée : 9 min 22 sec

4/ Historique des systèmes d'exploitation

Description rapide et petit historique sur les systèmes d'exploitation ici :

<https://www.youtube.com/watch?v=40hUDAtmAUo>

Auteur : Netprof , durée 3 min 38 s

Le système **UNIX** est un système dit "**propriétaire**" (certaines personnes disent "privateur"), c'est-à-dire un système non libre. Mais plus généralement, qu'est-ce qu'un **logiciel libre** ?

D'après Wikipédia : *"Un logiciel libre est un logiciel dont l'utilisation, l'étude, la modification et la duplication par autrui en vue de sa diffusion sont permises, techniquement et légalement, ceci afin de garantir certaines libertés induites, dont le contrôle du programme par l'utilisateur et la possibilité de partage entre individus"*. Le système UNIX ne respecte pas ces droits (par exemple le code source d'UNIX n'est pas disponible, l'étude d'UNIX est donc impossible), UNIX est donc un système "propriétaire" (le contraire de "libre"). Attention qui dit logiciel libre ne veut pas forcément dire logiciel gratuit (même si c'est souvent le cas), la confusion entre "libre" et "gratuit" vient de l'anglais puisque "free" veut à la fois dire "libre", mais aussi gratuit.

En 1991, un étudiant finlandais, Linus Torvald, décide de créer un clone libre d'UNIX en ne partant de rien puisque le code source d'UNIX n'est pas public. Ce clone d'UNIX va s'appeler Linux (Linus + UNIX).

En savoir plus sur l'histoire d'UNIX (et du langage C 😊) : <https://www.youtube.com/watch?v=Za6vGTLp-wg&t=158s>

Auteur : MOOC de l'IMT, durée : 4 min 57 sec

En savoir plus sur linux : https://www.youtube.com/watch?v=lquNF_DXcF8

Auteur : Cookie interconnecté, durée : 4 min 27 sec

Difficile de parler des systèmes d'exploitation sans parler de Microsoft !

Microsoft a été créée par **Bill Gates** et **Paul Allen** en 1975. Microsoft est surtout connue pour son système d'exploitation **Windows**. Windows est un système d'exploitation "propriétaire", la première version de Windows date 1983, mais à cette date Windows n'est qu'un ajout sur un autre système d'exploitation nommé MS-DOS.

Aujourd'hui Windows reste le système d'exploitation le plus utilisé au monde sur les ordinateurs grand public, il faut dire que l'achat de Windows est quasiment imposé lorsque l'on achète un ordinateur dans le commerce.

En savoir plus sur l'histoire de Windows : <https://www.youtube.com/watch?v=-5evLFh3ZAc>

Auteur : Clubic, durée : 10 min 07 sec

Bill Gates présentant Windows 98 avec un plantage en direct ici 😊 avec « l'écran bleu de la mort »:

<https://www.youtube.com/watch?v=IW7Rqwwth84>

Auteur : failednation, durée : 39 sec

Enfin pour terminer, quelques mots sur le système d'exploitation des ordinateurs de marque Apple : tous les ordinateurs d'Apple sont livrés avec le système d'exploitation MacOS. Ce système MacOS est un système d'exploitation UNIX, c'est donc un système d'exploitation propriétaire.

A noter : il y a un lien dans le chapitre sur les architectures matérielles sur l'histoire d'Apple.

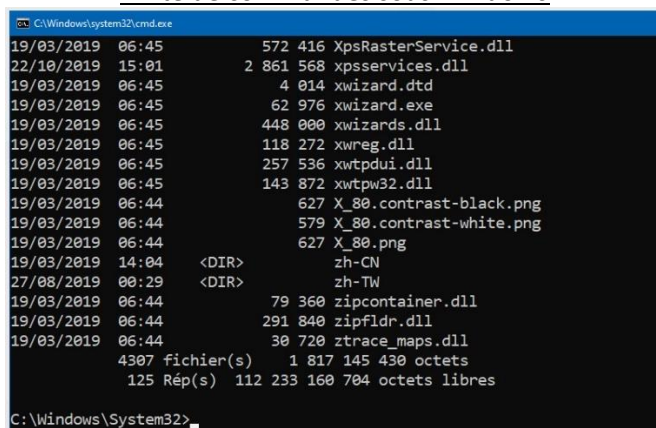
II/ Invite de commandes et terminal d'un système d'exploitation

1/ Introduction

En général, on utilise des **menus** pour interagir avec l'ordinateur pour choisir les demandes souhaitées. Auparavant, la seule façon était d'écrire des instructions avec un clavier et ma machine répondait en affichant du texte sur l'écran. Cette possibilité existe avec **l'invite de commandes** sous Windows ou avec un **terminal** sous Linux.

Il s'agit d'un logiciel qui affiche une interface en ligne de commande, ce qui signifie que la communication entre l'humain et la machine a lieu en mode texte. Ce logiciel interprète les commandes passées. Il en est de même pour un terminal sous Linux.

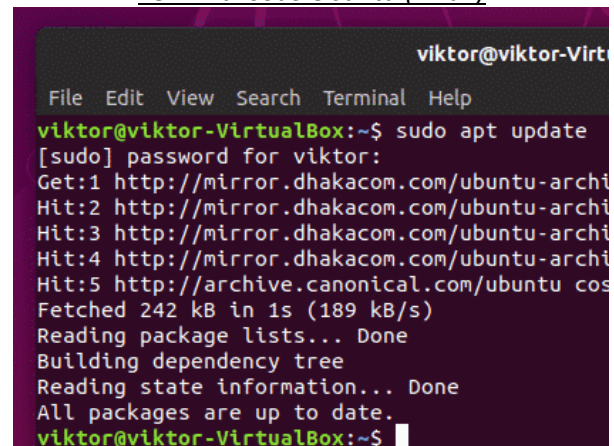
Invite de commandes sous Windows



```
C:\Windows\system32\cmd.exe
19/03/2019 06:45      572 416 XpsRasterService.dll
22/10/2019 15:01      2 861 568 xpsservices.dll
19/03/2019 06:45          4 014 xwizard.dtd
19/03/2019 06:45      62 976 xwizard.exe
19/03/2019 06:45     448 000 xwizards.dll
19/03/2019 06:45     118 272 xwreg.dll
19/03/2019 06:45     257 536 xwtpd.dll
19/03/2019 06:45     143 872 xwtpw32.dll
19/03/2019 06:44      627 X_80.contrast-black.png
19/03/2019 06:44      579 X_80.contrast-white.png
19/03/2019 06:44      627 X_80.png
19/03/2019 14:04    <DIR>      zh-CN
27/08/2019 00:29    <DIR>      zh-TW
19/03/2019 06:44      79 360 zipcontainer.dll
19/03/2019 06:44     291 840 zipfldr.dll
19/03/2019 06:44      30 720 ztrace_maps.dll
4307 fichier(s)    1 817 145 430 octets
125 Rép(s)    112 233 160 704 octets libres

C:\Windows\System32>
```

Terminal sous Ubuntu (Linux)



```
viktor@viktor-Virt
File Edit View Search Terminal Help
viktor@viktor-VirtualBox:~$ sudo apt update
[sudo] password for viktor:
Get:1 http://mirror.dhakacom.com/ubuntu-archi
Hit:2 http://mirror.dhakacom.com/ubuntu-archi
Hit:3 http://mirror.dhakacom.com/ubuntu-archi
Hit:4 http://mirror.dhakacom.com/ubuntu-archi
Hit:5 http://archive.canonical.com/ubuntu cos
Fetched 242 kB in 1s (189 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
viktor@viktor-VirtualBox:~$
```

2/ L'invite de commandes sous Windows

On trouve l'invite de commande dans le menu *Démarrer*, puis dans le sous-menu *Système Windows*. On peut aussi taper la commande *cmd* dans la barre de recherche puis la valider avec la touche *entrée*. Pour quitter la fenêtre, on écrit la commande *exit* que l'on valide.

Il existe plus de 200 commandes possibles sous Windows !

En voici quelques-unes :

ver : affiche la version du système d'exploitation.

systeminfo : affiche les propriétés et la configuration de l'ordinateur.

cls : efface le contenu de l'écran.

color value : change la couleur de l'affichage, *value* est un chiffre en base hexadécimale entre 1 et F. Les autres valeurs seront ignorées.

dir : affiche les fichiers et répertoires du répertoire courant.

cd name : permet d'accéder à un sous-dossier appelé *name* à partir du répertoire courant.

cd .. : « remonte » dans le répertoire parent (on « remonte de 1 » dans l'arborescence).

cd ../.. : on « remonte de 2 » dans l'arborescence. On peut combiner plusieurs « ../ ».

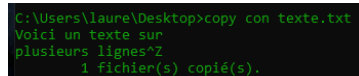
mkdir (ou md) name : crée un nouveau dossier appelé *name* dans le répertoire courant.

rmdir (ou rm) name : supprime le dossier nommé *name* dans le répertoire courant.

Quelques commandes pour manipuler des fichiers textes, on supposera que l'on reste dans le répertoire courant :

copy con name.txt : crée un fichier nommé *name.txt* avec le contenu ci-dessous dans le répertoire courant.

*Ecrire ici le contenu
du fichier même sur plusieurs
lignes*



```
C:\Users\laure\Desktop>copy con texte.txt
Voici un texte sur
plusieurs lignes^Z
1 fichier(s) copié(s).
```

Important : le contenu du fichier doit se terminer par la combinaison de touches *ctrl + z* puis être validé par la touche *entrée*.

type name.txt : affiche le contenu du fichier nommé *name.txt* dans la console.

name.txt : ouvre le fichier texte.

copy name.txt name1.txt : duplique le fichier nommé *name.txt* en un fichier nommé *name1.txt*.

copy name.txt + name1.txt name2.txt : concatène (additionne) le contenu des fichiers *name.txt* et *name1.txt* dans un fichier nommé *name2.txt*.

del name.txt : supprime définitivement le fichier nommé *name.txt*.

del *.txt : supprime tous les fichiers *.txt* (du répertoire courant).

move name.txt chemin absolu/relatif : déplace le fichier nommé *name.txt* dans un autre dossier.

Exemple : déplacement du fichier *texte.txt* du répertoire *rep1* dans le répertoire *rep2*.

```
C:\Users\laure\Desktop>mkdir rep1
C:\Users\laure\Desktop>mkdir rep2
C:\Users\laure\Desktop>cd rep1
C:\Users\laure\Desktop\rep1>copy con texte.txt
çanouspasparlerkobold^Z
1 fichier(s) copié(s).
C:\Users\laure\Desktop\rep1>move texte.txt ../rep2
1 fichier(s) déplacé(s).
```

Création des répertoires *rep1* et *rep2* sur le bureau.

On se place dans le répertoire *rep1*.

Création d'un fichier dans *rep1* nommé *texte.txt*.

Déplacement de ce fichier dans *rep2* : pour cela, on « remonte » dans *Desktop* puis on accède à *rep2*.

Enfin, une commande utile : **commande help** permet d'avoir le détail des attributs de la commande en question (s'ils existent).

Exemple à droite avec la commande **color**.

```
C:\Users\laure>color help
Change les couleurs par défaut du premier plan et de l'arrière-plan de la console

COLOR [attr]

attr   Spécifie les attributs de couleurs de l'apparence de la console

Les attributs de couleurs sont spécifiés par DEUX chiffres hexadécimaux -- le
premier correspond à l'arrière-plan, le second au premier plan. Chaque chiffre
peut prendre n'importe quelle de ces valeurs :

0 = Noir      8 = Gris
1 = Bleu      9 = Bleu clair
2 = Vert      A = Vert clair
3 = Bleu-gris B = Cyan
4 = Rouge     C = Rouge clair
5 = Violet    D = Violet clair
6 = Jaune     E = Jaune clair
7 = Blanc     F = Blanc brillant

Si aucun argument n'est donné, cette commande restaure les couleurs
sélectionnées au moment où CMD.EXE a été ouvert. Cette valeur vient soit de la
fenêtre de la console, du commutateur en ligne de commande /T, ou de la valeur
DefaultColor du registre.

La commande COLOR met ERRORLEVEL à 1 si vous tentez de l'exécuter
avec la même couleur pour l'arrière et le premier plan.
```

En savoir plus au sujet des commandes en ligne de Windows 10 ici : <https://waytolearnx.com/2019/06/liste-de-toutes-les-commandes-cmd-sous-windows-10-8-et-7.html>

Plutôt que d'écrire des commandes une par une dans l'invite de commande, il est possible d'écrire un ensemble de commandes dans un fichier.

Les commandes sont écrites dans l'éditeur Notepad++ et le fichier est enregistré sous un nom avec pour extension un *.bat*. Il s'agit d'un fichier **batch** (qui signifie lot). Le langage batch est un précurseur des langages de scripts. Cette possibilité permet d'automatiser certaines tâches au niveau du système d'exploitation et même du réseau local. Les commandes écrites dans le fichier sont exécutées de manière séquentielle après un double-clic sur l'icône du fichier.

Un tutoriel ici sur le langage batch de Windows : <https://devstory.net/11589/batch>

3/ Le terminal sous Linux

Le noyau Linux a été créé en 1991 par Linus Torvald. Le système Linux est un système libre. On parle aussi de GNU/Linux (noyau Linux plus logiciels GNU alors que les systèmes comme Android sont fondés sur Linux mais pas GNU), GNU est un projet lié aux logiciels libres lancé en 1983 par Richard Stallman.

Linux fait partie de la famille UNIX. Tous les systèmes d'exploitation d'ordinateurs ou de mobiles sont basés sur UNIX sauf Windows.

Le terminal sous Linux a un fonctionnement proche de l'invite de commandes de Windows. De manière générale, la syntaxe est la suivante : **nom [options] [--] [arguments]**.

Le **nom** est bien sûr le nom de la commande.

Une **option** est composée d'un signe moins suivi d'une lettre ; on peut ajouter zéro, une ou plusieurs options sont séparés par des espèces et -- signifie la fin des options.

Les **arguments** sont une liste de zéros, un ou plusieurs arguments sont séparés par des espaces.

Et enfin, la commande **man** affiche une aide sur la commande donnée en argument.

Voici quelques commandes en lien avec les répertoires :

cd : change le répertoire courant. S'il n'y a pas d'arguments, le répertoire courant devient le répertoire *home*.

mkdir : crée le répertoire précisé en argument.

rmdir : supprime le répertoire (qui doit être vide).

Voici quelques commandes en lien avec les fichiers :

dir ou **ls** : liste les fichiers du répertoire courant ou des répertoires donnés en arguments.

De nombreuses options sont disponibles, ainsi :

-**a** : liste tous les fichiers.

-**l** : affiche les fichiers avec les droits d'accès.

-**s** : affiche la taille occupée par les fichiers en nombre de blocs (qui dépend de l'OS).

cp : copie les fichiers, par exemple **cp fich1.py fich2.py**.

mv : renomme et déplace des fichiers, par exemple **mv fich1.py fich2.py**.

rm : supprime définitivement les fichiers donnés en argument.

4/ Droits et permissions sous Linux

a) Le monde selon Unix

Unix sépare le monde en trois catégories du point de vue des droits :

- L'utilisateur (*user* en anglais) noté « **u** ».
- Le groupe (*group* en anglais) noté « **g** ».
- Le reste du monde (*others* en anglais) noté « **o** ».

Unix permet de regrouper le monde sous l'appellation **user + group + others** noté « **ugo** » ou « **a** ».

b) Lecture de droits

```
drwxr-xr-x
```

Il se traduit de la manière suivante :

- **d** : c'est un répertoire.
- **rw**x pour le 1er groupe de 3 symboles : son propriétaire peut lire, écrire et exécuter.
- **r**-**x** pour le 2e groupe de 3 symboles : le groupe peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.
- **r**-**x** pour le 3ème groupe de 3 symboles : le reste du monde peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.

En pratique, en exécutant la commande suivante :

```
ls -l
```

on obtient la liste du contenu du répertoire courant, par exemple :

```
drwxr-xr-x  6 cyrille cyrille  4096 2008-10-29 23:09 Bureau
drwxr-x---  2 cyrille cyrille  4096 2008-10-22 22:46 Documents
lrwxrwxrwx  1 cyrille cyrille    26 2008-09-22 22:30 Examples -> /usr/share/example-content
-rw-r--r--  1 cyrille cyrille 1544881 2008-10-18 15:37 forum.xcf
drwxr-xr-x  7 cyrille cyrille  4096 2008-09-23 18:16 Images
drwxr-xr-x  2 cyrille cyrille  4096 2008-09-22 22:45 Modèles
drwxr-xr-x 267 cyrille cyrille 20480 2008-10-27 22:17 Musique
drwxr-xr-x  2 cyrille cyrille  4096 2008-09-22 22:45 Public
drwxr-xr-x  2 cyrille cyrille  4096 2008-10-26 13:14 Vidéos
```

On retrouve dans la première colonne le groupe de 10 caractères permettant de connaître les droits pour chaque fichier.

Ainsi, pour le fichier `forum.xcf`, on a :

```
-rw-r--r--
```

- Le 1er caractère est `-` ⇒ c'est un fichier.
- Le premier groupe de 3 caractères est `rw-` ⇒ le propriétaire a le droit de lecture et écriture (mais pas d'exécution) sur le fichier.
- Les 2 groupes suivants sont `r--` ⇒ Les utilisateurs du groupe et les autres n'ont que le droit de lecture (pas d'écriture, ni d'exécution).

c) Modifications des droits

Seul le propriétaire d'un fichier peut changer ses permissions d'accès. Il le fait avec la commande **chmod** (*change mode* en anglais).

Il va donc pouvoir choisir :

1. À qui s'applique le changement
 - **u** (user, utilisateur) représente la catégorie "propriétaire" ;
 - **g** (group, groupe) représente la catégorie "groupe propriétaire" ;
 - **o** (others, autres) représente la catégorie "reste du monde" ;
 - **a** (all, tous) représente l'ensemble des trois catégories.
2. La modification que l'on veut faire
 - **+** : ajouter
 - **-** : supprimer
 - **=** : affectation
3. Le droit que l'on veut modifier
 - **r** : read ⇒ lecture
 - **w** : write ⇒ écriture
 - **x** : execute ⇒ exécution

Par exemple :

```
chmod o-w fichier3
```

enlèvera le droit d'écriture pour les autres.

```
chmod a+x
```

ajoutera le droit d'exécution à tout le monde.

On peut aussi combiner plusieurs actions en même temps :

- On ajoute la permission de lecture, d'écriture et d'exécution sur le fichier **fichier3** pour le **propriétaire** ;
- On ajoute la permission de lecture et d'exécution au **groupe propriétaire**, on retire la permission d'écriture ;
- On ajoute la permission de lecture aux **autres**, on retire la permission d'écriture et d'exécution.

```
chmod u+rwx,g+rx-w,o+r-wx fichier3
```

Source : Wiki ubuntu-fr, la documentation francophone d'Ubuntu, <https://doc.ubuntu-fr.org/permissions>

En savoir plus sur le terminal de Linux ici : <https://www.sitedetout.com/tutoriels/commandes-linux-de-base/>