

# Administration d'un système GNU / Linux

## 01 — Les bases

Anthony Labarre

上海师范大学

16 octobre 2025



## Mises en garde

Le français est plus difficile que le chinois (pour vous); donc:

- si le cours va trop vite, interrompez-moi;
- si le français n'est pas clair: faites-moi répéter;
- si la matière n'est pas claire: posez des questions.

# Organisation du cours

 “**Cours magistral**” (CM): explications sur la matière;

 “**Travaux pratiques**” (TP): exercices sur ordinateur;

 Matériel de cours: <https://shorturl.at/3FwpZ>

 Anthony . Labarre @ univ-eiffel . fr (sans espaces)



## Planning prévisionnel

jeudi 16/10



vendredi 17/10



lundi 20/10



mardi 21/10



mercredi 22/10

jeudi 23/10

vendredi 24/10

lundi 27/10

mardi 28/10

mercredi 29/10

jeudi 30/10



**EXAM**

Deux versions des transparents: “normale” et “imprimable” (= *handout*).



Imprimez les versions *handout* pour pouvoir noter pendant le cours.

## Attentes

- Révisez les cours et préparez vos questions pour la prochaine séance;
- Terminez les exercices de TP chez vous: les séances seront notées;
- Pendant les séances de TP: n'hésitez pas à consulter le cours!
- Si un exercice est trop difficile: dites-le, on le corrigera ensemble;

Toute fraude sera évidemment sanctionnée.



Problèmes techniques? Envoyez-moi un mail, je pourrai créer une FAQ.

## Rendre les devoirs

- ✓ Envoyez vos devoirs par e-mail;
- ✓ **Subject:** `id french_name homework_number`  
Exemple: "1234567 Pierre TP01"
- ✓ **Attachment:** un seul fichier nommé  
`id-french_name-homework_number`  
Exemple: "1234567-Pierre-TP01.tar.bz2"

✗ PAS DE LIEN , , 163, ...

✗ PAS D'ENVOI PAR 

## Pour les exercices



Vous aurez besoin d'installer Debian sur Virtual-Box: suivez les instructions sur la page du cours.

# Plan d'aujourd'hui

① Introduction

② Le terminal

③ Fichiers

④ Utilisateurs

⑤ Groupes

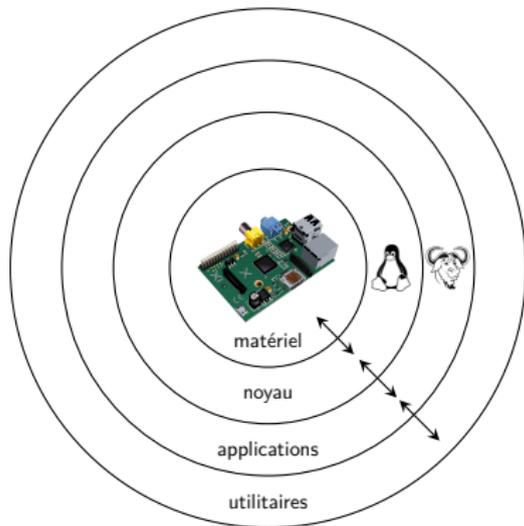
⑥ Permissions

# Introduction

# Qu'est-ce que GNU? Linux? GNU / Linux?

GNU <sup>a</sup> est un **système d'exploitation** (cf.  Windows,  OS X, ...). Il utilise le **noyau** Linux  pour former le système GNU / Linux.

<sup>a</sup>Logo de Aurelio A. Heckert



Souvent, on raccourcit “GNU / Linux” en “Linux”. Mais d'autres noyaux peuvent être utilisés (Hurd, BSD, ...).

# Composantes de GNU / Linux



## Le système GNU

- Créé en 1983 par Richard M. Stallman;
- Clone gratuit et “libre” (voir plus loin) de Unix;
- GNU = “**G**NU’s **N**ot **U**nix” (acronyme récursif);



## Le noyau Linux

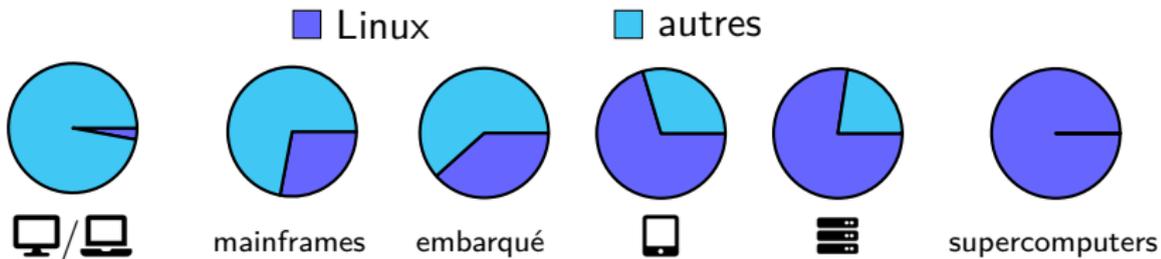
- Créé en 1992 par Linus B. Torvalds;
- Torvalds a également créé git, sur lequel sont basés  github,  gitlab, ...;

## Free software

- GNU et Linux sont du **free software**; il s'agit de “logiciel libre”, et pas “logiciel gratuit” (*free as in freedom*)!
- Un logiciel est libre si l'on peut:
  - ① l'utiliser sans restrictions;
  - ① l'étudier et modifier son code source;
  - ② redistribuer sans restrictions le logiciel ...
  - ③ ... et ses modifications!
- On peut donc étudier et améliorer le système à souhait ... et même créer son propre système GNU / Linux (sa **distribution**);

## Utilisation de Linux

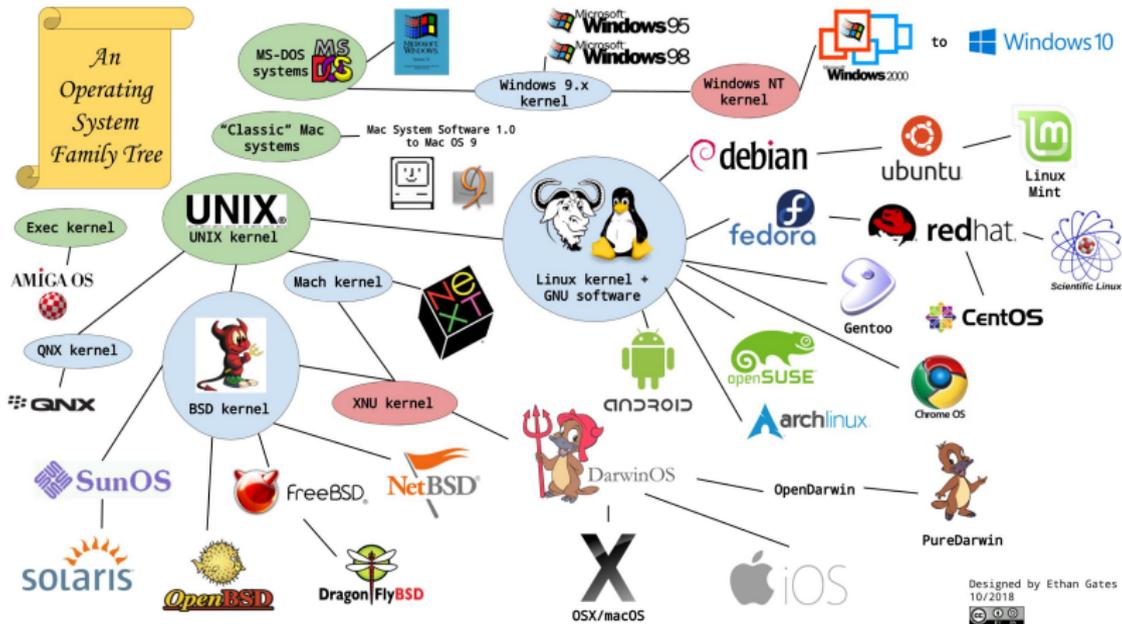
Linux est-il répandu? Les parts de marché dépendent du secteur:



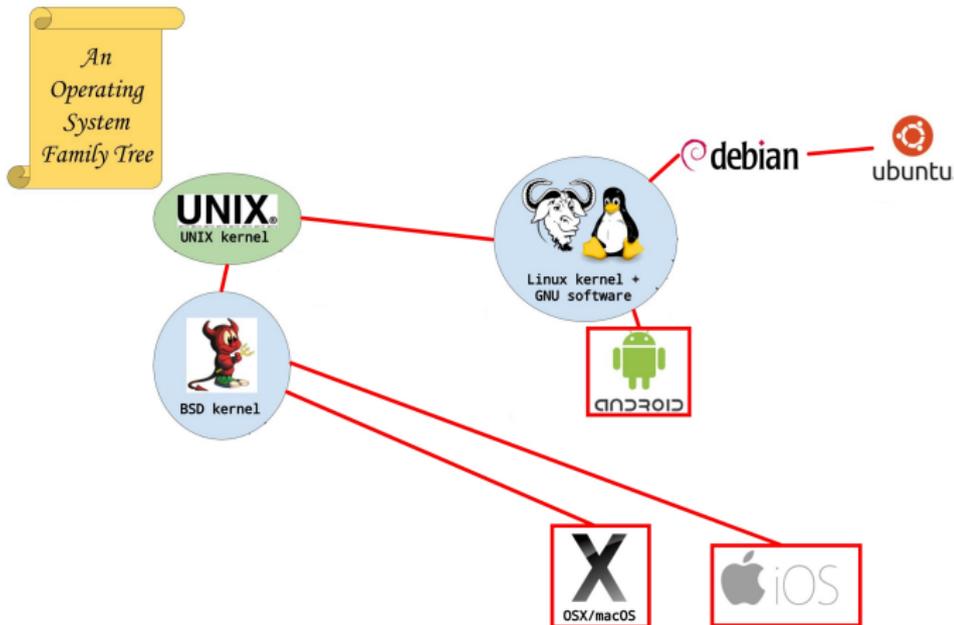
(source:

[https://en.wikipedia.org/wiki/Usage\\_share\\_of\\_operating\\_systems](https://en.wikipedia.org/wiki/Usage_share_of_operating_systems);  
on parle bien ici du **noyau** Linux, avec ou sans GNU)

# GNU / Linux et le restant du monde



# GNU / Linux et le restant du monde



Designed by Ethan Gates  
10/2018  


## Quelle distribution choisir?

- Il existe beaucoup de variantes de GNU / Linux, que l'on appelle des **distributions**;
- On utilisera **Debian**;



Les distributions se ressemblent très fort  $\Rightarrow$  ce qu'on apprend sous Debian nous servira sous les autres distributions.

- ⦿ Les notions spécifiques à Debian seront précédées de son logo;

## Caractéristiques de GNU / Linux

GNU / Linux se distingue par:

- son utilisation intensive de la “ligne de commande”;
- ▣ sa modularité, qui rend le système très personnalisable;
- sa gestion des périphériques (“tout est un fichier”).

Nous allons maintenant voir comment GNU / Linux gère:

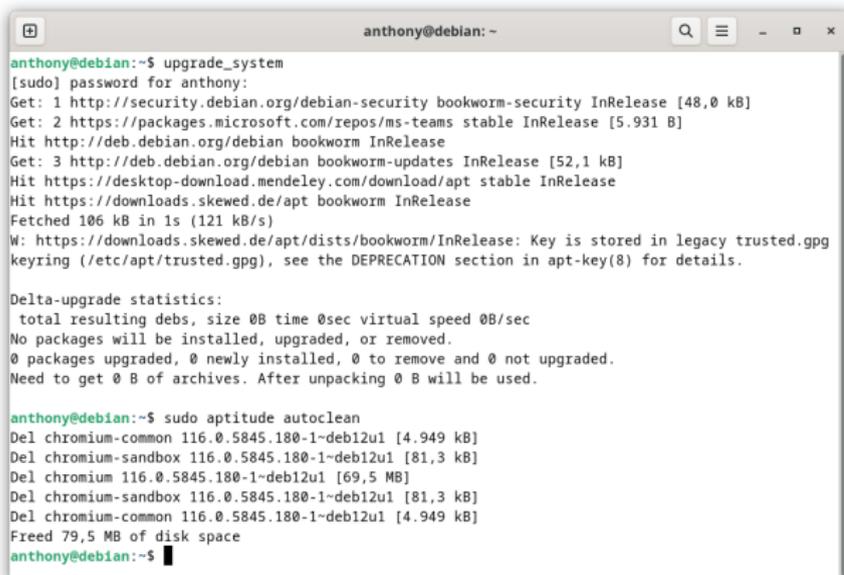
- les fichiers;
- les utilisateurs;
- les programmes;

## Le terminal



# Le terminal et la ligne de commande

- L'administration d'un système GNU / Linux se fait principalement à l'aide de la **ligne de commande**;
- On ouvre un **terminal** dans lequel on écrit des **commandes** pour réaliser les tâches voulues;



```
anthony@debian: ~  
anthony@debian:~$ sudo upgrade_system  
[sudo] password for anthony:  
Get: 1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]  
Get: 2 https://packages.microsoft.com/repos/ms-teams stable InRelease [5.931 B]  
Hit http://deb.debian.org/debian bookworm InRelease  
Get: 3 http://deb.debian.org/debian bookworm-updates InRelease [52,1 kB]  
Hit https://desktop-download.mendeley.com/download/apt stable InRelease  
Hit https://downloads.skewed.de/apt bookworm InRelease  
Fetched 106 kB in 1s (121 kB/s)  
W: https://downloads.skewed.de/apt/dists/bookworm/InRelease: Key is stored in legacy trusted.gpg  
keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.  
  
Delta-upgrade statistics:  
total resulting debs, size 0B time 0sec virtual speed 0B/sec  
No packages will be installed, upgraded, or removed.  
0 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
Need to get 0 B of archives. After unpacking 0 B will be used.  
  
anthony@debian:~$ sudo aptitude autoclean  
Del chromium-common 116.0.5845.180-1-deb12u1 [4.949 kB]  
Del chromium-sandbox 116.0.5845.180-1-deb12u1 [81,3 kB]  
Del chromium 116.0.5845.180-1-deb12u1 [69,5 MB]  
Del chromium-sandbox 116.0.5845.180-1-deb12u1 [81,3 kB]  
Del chromium-common 116.0.5845.180-1-deb12u1 [4.949 kB]  
Freed 79,5 MB of disk space  
anthony@debian:~$
```

## Le terminal et la ligne de commande

### ✓ Avantages:

- administration de systèmes à distance;
- programmation de scripts de maintenance;
- beaucoup de moyens différents d'effectuer une même tâche;
- ...

- × Inconvénient: il faut apprendre toutes ces commandes;  
heureusement, des manuels très complets existent;

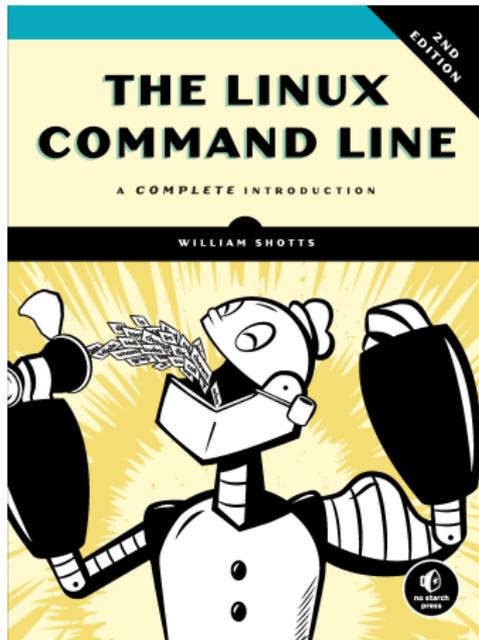


Nous verrons ensemble de nombreuses commandes;  
rédigez un résumé pour vous souvenir de leurs noms  
et de leurs effets!

## Référence utile

Le livre “The Linux Command Line” de William Shotts est un guide très complet sur l’utilisation de la ligne de commande.

 Il est disponible gratuitement en PDF [ici!](#)



# Format général des commandes

## Syntaxe

```
commande [option(s)] [paramètre(s)]
```

Le format des commandes est très souple:

## Exemple

Les quatre commandes suivantes font exactement la même chose:

```
# commande   options   paramètre
$ ls         -h -a      /home/anthony   # format "classique"
```

```
$ ls /home/anthony -h -a                               # options à la fin
```

```
$ ls /home/anthony --human-readable --all # options "longues"
```

```
$ ls /home/anthony -ha                               # options courtes combinées
```

## *What is that command?*



La commande `whatis` commande donne une description de la commande!

## Options, options, options, ...

- Les commandes basiques possèdent des options parfois très nombreuses (ls en possède 40!);
- Elles permettent de modifier le comportement de la commande;
- Si la commande que vous utilisez ne fait pas exactement ce que vous voulez, consultez le manuel pour trouver l'option qui vous arrange (slide suivant);

## man: le manuel

- 🧐 Doit-on vraiment apprendre toutes ces commandes et leurs options par cœur?
- 😊 Heureusement non! Si l'on a un trou de mémoire, on peut consulter le manuel;
- 👍 Pour tout savoir sur la commande voulue, taper `man` commande dans un terminal.
- ? Si `man` ne fonctionne pas, essayez `help`;



Il est utile de mémoriser **ce que font** les commandes, mais pas leurs options dans le moindre détail.

## Complétion automatique



La touche  permet de compléter automatiquement les commandes.

Tapez quelques lettres dans le terminal, puis :

- s'il n'existe qu'une seule commande commençant par votre mot, elle sera complétée;
- sinon, appuyez une seconde fois sur  donnera la liste de toutes les commandes commençant par ce mot.

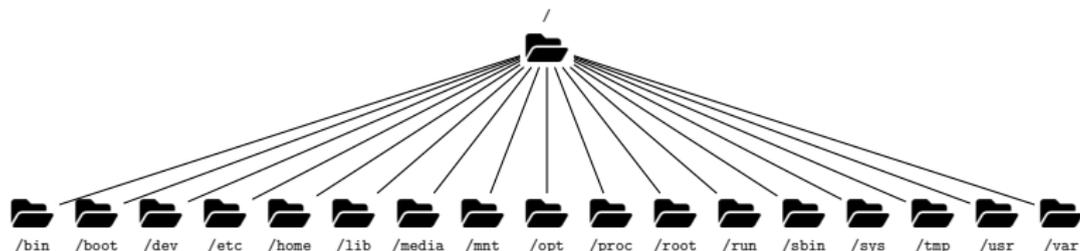


*Essayez dans le terminal!*

# Fichiers

## Structure du système de fichiers

Le **système de fichiers** structure les données sur le(s) disque(s).  
C'est une **arborescence** qui suit les conventions Unix:



- / est la **racine** (à peu près comme C:\ sous Windows);
- Le caractère / sépare les répertoires (= \ sous Windows);
- Les majuscules et minuscules importent (*case sensitivity*);

## Quelques répertoires importants

→ . est le répertoire actuel;

↑ .. est le répertoire **parent**;

🏠 ~ est le répertoire **personnel**;

- /dev (pour **devices**) contient le matériel (disques durs, processeurs, ...);

📄 /etc contient les fichiers de configuration globaux;

🏠 /home contient les répertoires personnels des utilisateurs;

- /mnt et /media contiennent les disques “montés”;

🕒 /tmp contient des fichiers temporaires: il est vidé à chaque redémarrage;

- /var contient diverses données (en particulier des “logs” dans /var/logs);

## Commandes basiques pour manipuler les fichiers

 `pwd`: affiche le répertoire actuel (le chemin complet vers `.`);

 `ls chemin`:

 si `chemin` est un fichier, affiche son nom;

 si `chemin` est un répertoire, affiche son contenu;

- si `chemin` est vide, alors `chemin` = `.`;

→ `cd chemin`: change le répertoire actuel vers `chemin` (si vide, alors `chemin` = `~`);

 `cp source cible`: copie le fichier source vers le fichier cible;

 `rm fichier`: efface le fichier;

 `mkdir chemin`: crée le répertoire `chemin`;

 `rmdir chemin`: efface le répertoire `chemin` **s'il est vide**;

## Types de systèmes de fichiers

- De très nombreux types de systèmes de fichiers existent, avec des performances différentes;
- En pratique, on retrouve surtout:
  -  apfs, hfsplus;
  -  ext2, ext3, ext4;
  -  fat, vfat, ntfs;

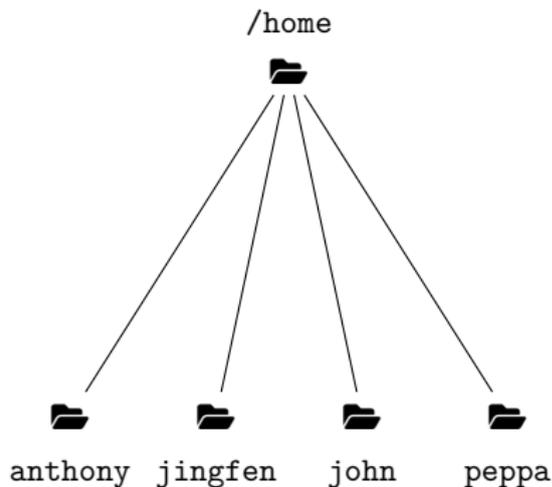


Pour pouvoir lire vos fichiers “n’importe où”, utilisez vfat ou ntfs pour vos clés USB.

# Utilisateurs

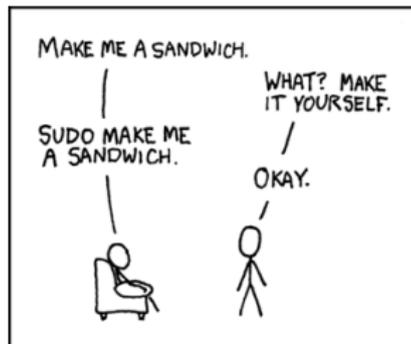
## Utilisateurs

- GNU est un système **multi-utilisateurs**: plusieurs utilisateurs peuvent s'y connecter en même temps et partager des ressources;
- Chaque utilisateur a un nom (ex: anthony) et un répertoire personnel dans /home (ex: /home/anthony/);
- Le système peut théoriquement accueillir  $2^{32}$  utilisateurs;



## Super-utilisateur

- Pour des raisons de sécurité, un utilisateur ordinaire ne peut pas tout faire;
  - Seul le **super-utilisateur** peut:
    - installer / supprimer des programmes;
    - accéder à des fichiers “dangereux”;
    - ...
- © La commande `sudo` permet d'exécuter une commande en tant que super-utilisateur;



<https://xkcd.com/149>



## Le super-utilisateur

- Le super-utilisateur s'appelle root;
- Son répertoire personnel est /root;



Il est possible, mais très dangereux, de devenir super-utilisateur: **ne le faites pas.**

- À la place, on lui **demandera** d'exécuter certaines commandes avec sudo;

## La commande su

- Exceptionnellement, on aura besoin de la commande su;
- On ne l'utilisera que si on n'a pas le droit d'exécuter sudo:

```
pingouin@debian:~$ sudo apt update
[sudo] password for pingouin:
pingouin is not in the sudoers file.
This incident has been reported to the administrator.
```

## La commande su

- `su -c 'commande options paramètres'` permet d'exécuter une seule commande en tant que super-utilisateur;
- On doit rajouter pingouin au fichier `/etc/sudoers` avec les bonnes autorisations:

```
pingouin@debian:/tmp$ su -c 'nano /etc/sudoers'
Password: # entrer le mot de passe de root
...
# User privilege specification
root    ALL=(ALL:ALL) ALL
pingouin ALL=(ALL:ALL) ALL # <--- ajouter cette ligne
```

- Après cela, pingouin pourra lancer les mêmes commandes que root avec `sudo`;

# Utilisateurs

- Les utilisateurs sont identifiés par un nom et un UID (**u**ser **i**dentifier);
- Le super-utilisateur root a toujours l'UID 0;
- Il y a beaucoup d'utilisateurs "spéciaux";

## Exemple (fichier /etc/passwd)

Le fichier /etc/passwd contient les utilisateurs:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
...
anthony:x:1234:1234:Anthony Labarre,,,:/home/anthony:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:./:/usr/sbin/nologin
...
```

## Gestion des utilisateurs

Les commandes suivantes permettent de gérer les utilisateurs:

-  `sudo adduser nom`: ajoute l'utilisateur avec le nom donné;
-  `sudo deluser nom`: supprime l'utilisateur avec le nom donné;  
(son répertoire personnel n'est pas supprimé; rajoutez l'option `--remove-home` ou `--remove-all-files` pour ça)
-  `sudo usermod nom`: modifie l'utilisateur avec le nom donné;
-  `passwd`: change le mot de passe de l'utilisateur actuel;

# Groupes

## Groupes

- Les groupes sont identifiés par un nom et un GID (**group identifier**);
- Ils servent à définir des **permissions** (voir plus loin) de manière plus globale et simple; par exemple:
  - faire partie de `audio` permet d'utiliser les périphériques audio;
  - faire partie de `lpadmin` permet de configurer les imprimantes;
  - faire partie de `sudo` permet d'utiliser `sudo` (!);
  - ...
- La commande `groups` affiche les groupes dont vous faites partie;
- Chaque utilisateur fait partie de son propre groupe;

## Gestion des groupes

Les commandes suivantes permettent de gérer les groupes:

- `sudo addgroup nom`: ajoute le groupe avec le nom donné;
- `sudo delgroup nom`: supprime le groupe avec le nom donné;
- `sudo groupmod nom`: modifie le groupe avec le nom donné;

Si l'on veut ajouter ou supprimer un utilisateur d'un groupe, on utilise `usermod`.

# Permissions

# Permissions

Des **permissions** associées à chaque fichier déterminent ce que chaque utilisateur peut faire d'un fichier.

## Types d'utilisateurs

- 1 le propriétaire du fichier;
- 2 le groupe du fichier;
- 3 les autres (ni propriétaire ni groupe);

## Types de permissions

- 1 r (= **r**ead: lecture);
- 2 w (= **w**rite: écriture);
- 3 x (= **x**ecute: exécution);

Si le fichier est un répertoire, x permet de le traverser. Un fichier exécutable se lance avec la commande `./fichier`

## Permissions: exemple avec `ls -l`

Pour connaître les permissions d'un fichier, utilisez la commande:

```
$ ls -l /chemin/vers/fichier
```

Exemple (résultat de `ls -l /var` pour `/var/log`)

type	permissions				propriétaire	groupe	taille	dernière modification	nom			
d	r	w	x	r	w	x	15	root	syslog	4096	oct 6 12:33	log
												
	propriétaire			groupe								
												
			autres									

Les types de fichier les plus fréquents sont:

-  - pour un fichier ordinaire,
-  d pour un répertoire, ou
-  l pour un lien (on en parlera plus tard).

## ! Exception: le super-utilisateur

Le super-utilisateur peut contourner les permissions:

- 1 il peut lire et écrire dans tous les fichiers, **même sans les permissions valides**;
- 2 il peut exécuter tout fichier **du moment qu'il y ait au moins un x dans les permissions**;

### Exemple (droits pour )

Soit un fichier bla avec pour permissions -----:

- ✓ `sudo cat bla` et `su -c 'cat bla'` affichent son contenu;
- ✓ `sudo echo bonjour > bla` et `su -c 'echo bonjour > bla'` écrivent "bonjour" dans ce fichier;
- ✗ `sudo ./bla` et `su -c ./bla` échouent;

## Changer les attributs d'un fichier

Les commandes suivantes modifient les propriétés d'un fichier:

- `chmod`: change les permissions du fichier
- `chown`: change le propriétaire et / ou le groupe du fichier
- `chgrp`: change le groupe du fichier

# Utilisation de chmod

## Syntaxe de chmod

```
chmod CATÉGORIE(S)±PERMISSION(S) fichier
```

avec: CATÉGORIE:

- u (pour **u**ser): propriétaire du fichier;
- g (pour **g**roup): groupe du fichier;
- o (pour **o**thers): les autres;
- a (pour **a**ll): tout le monde;

+ ajoute une permission (r, w, x), - la supprime.

## Utilisation de chmod: exemple

Modifions les permissions du fichier suivant:

```
$ ls -l fichier
----- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod u+r fichier; ls -l fichier
```

```
-r----- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod g+w fichier; ls -l fichier
```

```
-r---w---- 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod o+x fichier; ls -l fichier
```

```
-r---w---x 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod a+r fichier; ls -l fichier
```

```
-r--rw-r-x 1 anthony anthony 0 oct 31 03:33 fichier
```

```
$ chmod ug-r fichier; ls -l fichier
```

```
-----w-r-x 1 anthony anthony 0 oct 31 03:33 fichier
```

*user group others*  
(*rwX rwX rwX*)



Si vous n'avez pas tout trouvé, réessayez chez vous.

Vous pouvez créer la configuration de départ avec:

```
$ touch fichier
$ chmod -rwx fichier
```

`touch nom` crée le fichier vide  
`nom` s'il n'existe pas, ou  
change sa dernière date  
d'accès.

## Changer les attributs d'un fichier

On peut aussi utiliser un numéro par catégorie; chaque numéro (dés)active alors les trois permissions de la catégorie:

code	signification	code	signification
0	---	4	r--
1	--x	5	r-x
2	-w-	6	rw-
3	-wx	7	rwX

### Exemple

```
$ touch fichier_vide
$ chmod 123 fichier_vide
$ ls -l fichier_vide
---x-w--wx 1 anthony anthony 0 sep  2 11:27 fichier_vide
```