
Chapitre 3: Commandes Linux

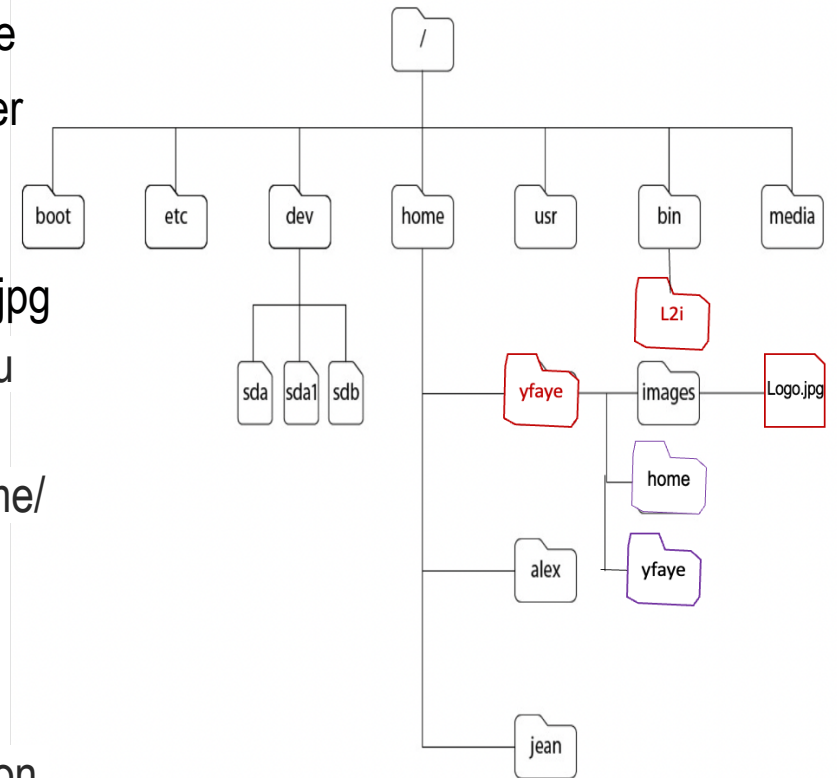
Pr Youssou FAYE

Nom des fichiers et répertoires

- ▶ Les noms sont des séquences de maximum 255 caractères sensibles à la casse. Les
- ▶ Ne pas utiliser les caractères: ?, *, & ,. , ` , ' , " , < , >
- ▶ Utiliser les lettres de A à Z et de a à z, les chiffres de 0 à 9, le underscore `_`, le tiret `-` et le point `.`
- ▶ Le point (.) comme premier caractère d'un nom signifie un fichier caché.
- ▶ Éviter les noms contenant des caractères accentués ou des blancs.
- ▶ Le point (.) ne sépare pas forcément le nom de son extension
- ▶ Exemples: filiere.l2i.pdf, Semestre-1_2_3

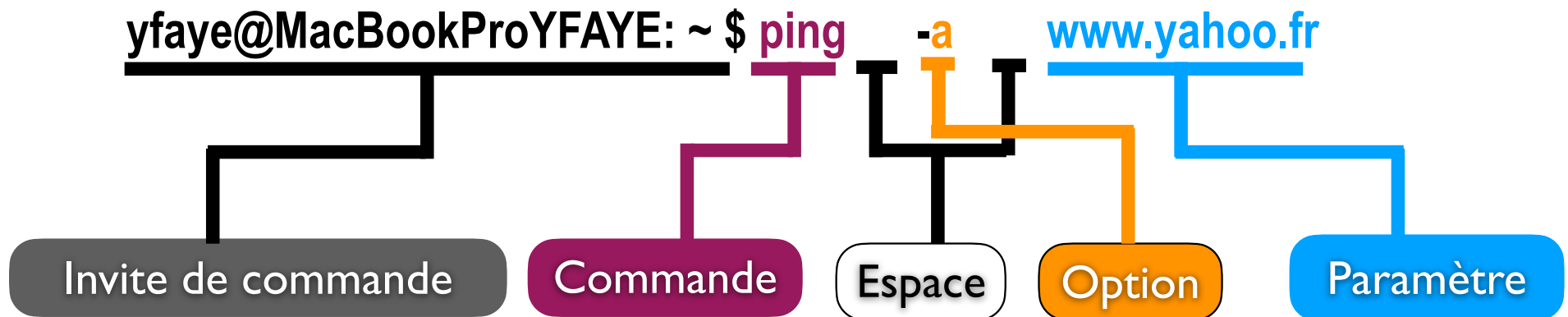
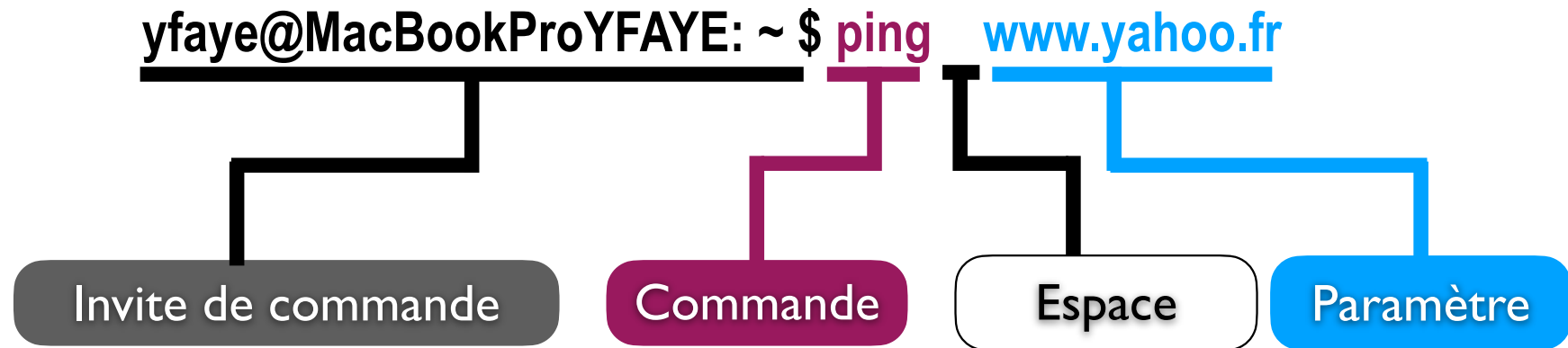
Se déplacer dans l'arborescence

- ▶ Un fichier peut être désigné de façon absolue ou relative
- ▶ **Chemins absolus**: identifie de manière unique un fichier ou un répertoire depuis la racine
 - `/usr/home/yfaye` : chemin absolu du dossier yfaye
 - `/home/yfaye/images/logo.jpg`: chemin du fichier logo.jpg
- ▶ **Chemin relatif**: identifie de manière unique un fichier ou un répertoire depuis le dossier courant
 - `images/Logo.jpg`: si on est déjà dans le dossier `/home/yfaye`
- ▶ Le point (`.`): désigne le répertoire courant
- ▶ Les points (`..`) désigne le répertoire parent
- ▶ Depuis `/home/yfaye`, si on souhaite se déplacer de façon relative dans d'autres branches de l'arborescence:
 - ▶ `home`: identifie le dossier home fils et `..` identifie home parent.
 - ▶ `../bin/L2i`: identifie de manière unique L2i



Commandes de base

► Rappel: syntaxe des commandes



Commandes de base

- ▶ **ls** pour **list** affiche les fichiers dans le répertoire courant, en ordre alphanumérique, sauf ceux qui commencent par le caractère “.”
- ▶ **ls a** (option «**all**»): affiche tous les fichiers
- ▶ **ls l** (option **long**): affiche en format long (type, date, taille, propriétaire, permissions)
- ▶ **ls t** (option **temps**): affiche les fichiers les plus récents en premier
- ▶ **ls S** (option “**size**”): affiche les fichiers les gros en premier

Commandes de base

- ▶ **cd**: **change directory** permet de changer de répertoire
- ▶ **cd** (sans «**paramètre**»): fait du répertoire utilisateur le nouveau répertoire courant
- ▶ **cd** **—**: fait du répertoire précédent le répertoire courant
- ▶ **cd** **/yfaye**: fait du répertoire **yfaye** le répertoire courant
- ▶ **cd** **..** : fait du répertoire **/** le répertoire courant si on se trouve déjà dans le répertoire **/yfaye**.

Commandes de base

► cd

yfaye@MacBookProYFAYE: ~ \$ **cd ..**

yfaye@MacBookProYFAYE: /home\$ **cd ..**

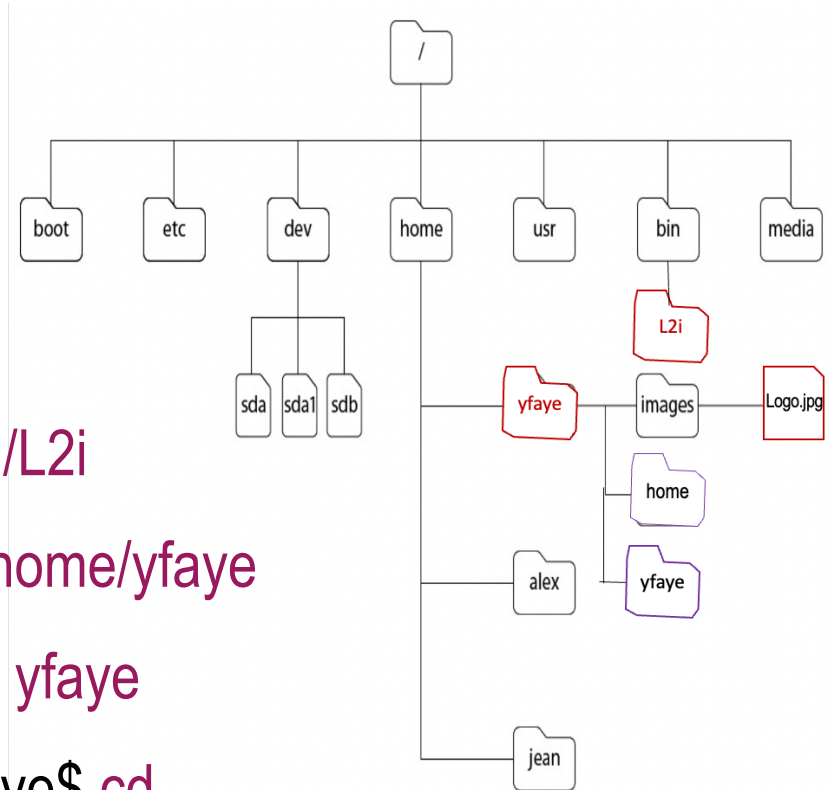
yfaye@MacBookProYFAYE: /home\$ **cd ../bin/L2i**

yfaye@MacBookProYFAYE: /bin/L2i\$ **cd ../../home/yfaye**

yfaye@MacBookProYFAYE: /home/yfaye\$ **cd yfaye**

yfaye@MacBookProYFAYE: /home/yfaye/yfaye\$ **cd**

yfaye@MacBookProYFAYE: /home/yfaye/yfaye\$ **cd image/Logo.jpg** //logo.jpg n'est pas un répertoire



Commandes de base

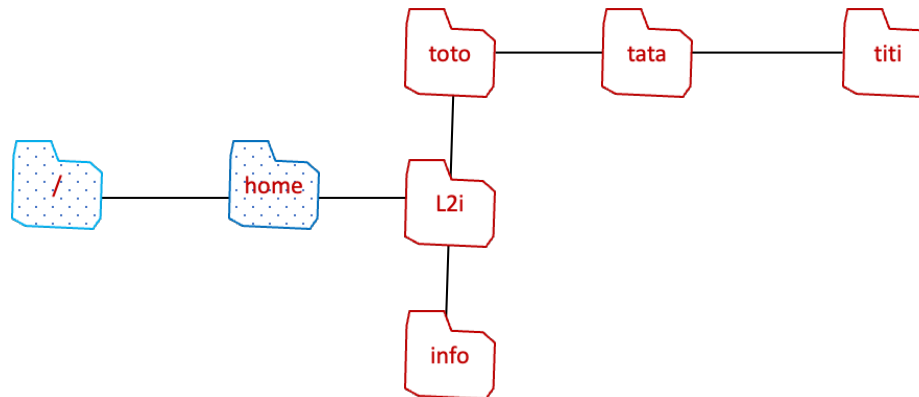
- **mkdir: make directory** permet de créer un nouveau répertoire dans le courant s'il n'existe pas ou génère une erreur s'il existe déjà

```
yfaye@MacBookProYFAYE: /home$
```

```
yfaye@MacBookProYFAYE: /home$ mkdir toto
```

```
yfaye@MacBookProYFAYE: /home$ mkdir toto/tata //il faut toto existe déjà
```

```
yfaye@MacBookProYFAYE: /home$ mkdir toto/tata/titi L2i/info
```



```
yfaye@MacBookProYFAYE: /home$ mkdir -p /L1/L2/L3 //crée la chaine de répertoire  
L1, L2 et L3 en même temps (parent)
```

Commandes de base

- ▶ **pwd** pour *print working directory*: affiche le chemin absolu du répertoire courant
- ▶ **mv** toto yfaye : (*move*) renome toto en yfaye
- ▶ **cp** toto.txt tata.txt: (*copy*) copie le contenu du fichier toto.txt dans tata.txt
- ▶ **cp -i** toto.txt tata.txt: *itérative* (demande confirmation)
- ▶ **cp -r** toto tata: *récursive* (copie le répertoire en entier)
- ▶ **rm** toto tata titi : (*remove*) supprime les fichiers donnés
- ▶ **rm -i** toto tata : (*interactive*)
- ▶ **touch** toto.txt : permet de créer un fichier

Commandes de base

► Les caractères génériques

- * dénote 0, 1 ou plusieurs caractères
 - *.c doc*
- ? dénote un seul caractère quelconque
 - ?nix *.s??
- [] dénote un caractère appartenant à un ensemble
 - [Uv]nix *.vl[ci]
- [^] dénote un caractère n'appartenant pas à un ensemble
 - [^co]

Unix, unix, mnx,
vnx, labo.txt,
cola.txt colatxt

yfaye@MacBookProYFAYE: ~ \$ **ls *.txt** => Cola.txt, labo.txt

yfaye@MacBookProYFAYE: /home\$ **ls ?nix** => Unix, unix, vnx, mix

yfaye@MacBookProYFAYE: /home\$ **ls [Uu]nix** => Unix, unix

yfaye@MacBookProYFAYE: /home\$ **ls [^Uu]nix** => vnx, mnx

Commandes de base

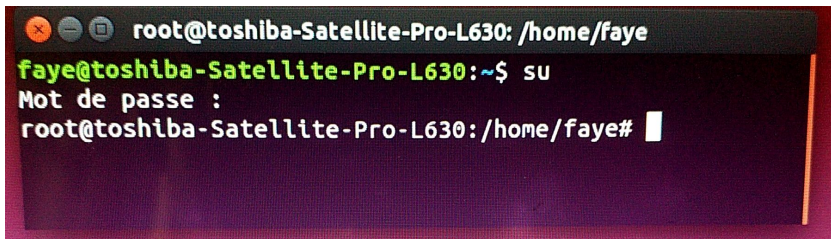
Les utilisateurs

- Pour effectuer une tâche sur un fichier ou installer un logiciel on a besoin d'obtenir des droits de super utilisateur.

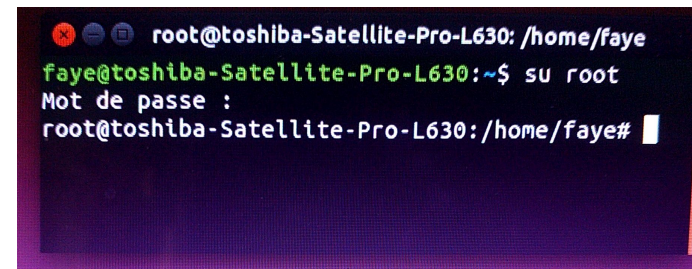
yfaye@MacBookProYFAYE: ~ \$ **whoami** // affiche l'utilisateur courant

yfaye@MacBookProYFAYE: ~ \$ **yfaye**

- **su** pour **substitute user** ou **switch user** (appelé à tort semble-t-il) **S**uper **U**tilisateur, permet de se connecter en tant **root**.
- **su** sans **paramètre**: le nom **root** est utilisé par défaut et le mot de passe de utilisateur **root** est demandé par le système
 - **su** = **su root** // mot de pass root obligatoire



```
root@toshiba-Satellite-Pro-L630: /home/faye
faye@toshiba-Satellite-Pro-L630:~$ su
Mot de passe :
root@toshiba-Satellite-Pro-L630: /home/faye#
```



```
root@toshiba-Satellite-Pro-L630: /home/faye
faye@toshiba-Satellite-Pro-L630:~$ su root
Mot de passe :
root@toshiba-Satellite-Pro-L630: /home/faye#
```

Commandes de base

Les utilisateurs

- **root** peut utiliser **su faye** pour se connecter en tant que yfaye ou passer à n'importe quel compte d'utilisateur. Il n'a pas besoin de fournir un mot de passe

```
root@toshiba-Satellite: /home/toshiba# su yfaye //pas besoin de mot de passe
yfaye@toshiba-Satellite: /home/toshiba$
```

- Un utilisateur autre que **root** peut utiliser **su**, il doit fournir le mot de passe root

```
yfaye@toshiba-Satellite: /home/yfaye$ su youssou //mot de passe root à fournir
```

Mot de passe:

```
youssou@toshiba-Satellite: /home/youssou$
```

Commandes de base

Les utilisateurs

- ▶ **su** a toujours été la manière traditionnelle d'acquérir des permissions root sur Linux
- ▶ **sudo** pour (**substitute user do**) permet à un utilisateur d'exécuter une commande en tant que *root* (sans être connecté en tant que root): **devenir root pour un instant**
- ▶ La commande sudo existe avant Ubuntu qui a été la première distribution Linux populaire à:
 - ▶ créer un utilisateur pouvant exécuter `sudo`
 - ▶ créer le compte root standard sans mot de passe.
 - ▶ ne pas pouvoir se connecter en tant que root avant d'avoir attribué un mot de passe au compte root.
 - ▶ La commande **sudo passwd root** // active le compte root et demande de créer un mot de passe pour le root
 - ▶ La commande **sudo -i** // permet d'avoir un shell root, après avoir donné son mot de passe et non celui du root.

yfaye@toshiba-Satellite: /home/yfaye\$ **sudo -i** //mot de passe de yfaye et non de root

Mot de passe [sudo] pour yfaye:

yfaye n'apparaît pas dans le fichier sudoers, cet événement sera signalé.

toshiba@toshiba-Satellite: /home/toshiba\$**sudo -i**. //mot de passe de toshiba

root@toshiba-Satellite: /root #

Commandes de base

Les utilisateurs

- ▶ Seuls certains utilisateurs appelés sudoers ont le droit d'exécuter la commande sudo.
- ▶ Pour ajouter un utilisateur dans la liste des sudoers, il faut excéuter en tant que root (ou via un autre utilisateur sudoers) :

- ▶ **adduser yfaye sudo** (en tant que *root*)

- ▶ **sudo adduser yfaye sudo** (avec un utilisateur déjà *sudoers*)

- ▶ **sudo commande**: le mot de passe de l'utilisateur lançant sudo est demandé par le système

toshiba@toshiba-Satellite: ~\$ **sudo apt-get install Firefox** //mot de passe de toshiba

Mot de passe [sudo] pour toshiba:

toshiba@toshiba-Satellite: ~\$

- ▶ La commande **sudo su** // pour se connecter comme root avec son mot de passe et non celui du root: **devenir root et le rester**

toshiba@toshiba-Satellite: ~\$ **sudo su** //mot de passe de toshiba

Mot de passe [sudo] pour yfaye:

root@toshiba-Satellite: /home/toshiba#**sudo -i**

root@toshiba-Satellite: /root#

Gestion des utilisateurs

- ▶ Sous Linux, les utilisateurs sont regroupés en groupes
- ▶ Par défaut un utilisateur est créé avec un groupe (**groupe primaire**) portant son nom
- ▶ Chaque utilisateur doit faire partie au moins d'un groupe, son **groupe primaire**, il peut être invité dans un ou plusieurs groupes secondaires.
- ▶ Les groupes et utilisateurs se gèrent par leur identifiant numérique unique **GID** et **UID**.
 - ▶ **UID** *User IDentifier*. Identifiant unique d'utilisateur.
 - ▶ **GID** *Group IDentifier*. Identifiant unique de groupe.
- ▶ Il existe trois types d'utilisateurs
 - ▶ **super-utilisateur ou super-administrateur**, son identifiant **root** (sous Ubuntu, son accès direct est bloqué) est **Administrateur** du système, le premier compte créé lors de l'installation est un compte super-administrateur.
 - ▶ **utilisateur ordinaire** : créé pour chacun des usagers de l'ordinateur, il identifie les gens qui se servent de votre ordinateur.
 - ▶ **Comptes système**: créés par le système Ubuntu, généralement invisibles aux usagers, servent à faire fonctionner certains services du système

Gestion des utilisateurs

Création d'un utilisateur

- La commande **adduser** (interactive) réservée à administrateur permet d'ajouter un utilisateur, fournir un paramètre : le nom de l'utilisateur à créer.

```
root@toshiba-Satellite: /home/toshiba#adduser Et1 // crée l'utilisateur Et1.
```

```
root@toshiba-Satellite: /home/toshiba#adduser Et2 // crée l'utilisateur Et2
```

```
root@toshiba-Satellite: /home/toshiba#deluser Et1. //supprime Et1
```

```
root@toshiba-Satellite: /home/toshiba#passwd Et1 //crée /change le mot de passe
```

Saisir 2 fois le mot de passe

```
root@toshiba-Satellite: /home/toshiba#cat /etc/passwd //liste les utilisateurs
```

► Format de /etc/passwd

- nom_utilisateur:mot_de_passe:uid:gid:commentaire:home:shell

- **nom_utilisateur**: login de l'utilisateur.
- **mot_de_passe**: mot de passe de l'utilisateur remplacer par un x pour des raisons de sécurité
- **uid**: identifiant système de l'utilisateur
- **gid**: groupe principal de l'utilisateur.
- **commentaire**: commentaire textuel sur l'utilisateur qui est souvent son nom réel (Prénom et Nom).
- **home**: répertoire home de l'utilisateur sur ce système.
- **shell**: l'interpréteur shell par défaut de l'utilisateur

```
yfaye:x:1000:1000:Youssoufaye:/home/yfaye:/bin/bash
```


Gestion des utilisateurs

Création d'un utilisateur

- ▶ La commande `useradd` (non-interactive) réservée à un administrateur permet de créer un utilisateur automatiquement.

- ▶ `useradd` permet de créer des scripts exploitant: le script `adduser` en est un exemple,

```
root@toshiba-Satellite: /home/toshiba#useradd Et1 // crée l'utilisateur Et1.
```

- ▶ Un compte créé avec `useradd` est inactif. Pour l'activer, il faut attribuer un mot de passe au compte.

```
root@toshiba-Satellite: /home/toshiba#passwd Et1 //demande de créer le mot de passe pour Et1.
```

- ▶ Pour forcer l'utilisateur à changer de mot de passe lors de sa première connexion

```
root@toshiba-Satellite: /home/toshiba# passwd -e Et1
```

Gestion des utilisateurs

Groupe: création

- ▶ Un compte d'utilisateur peut être affecté à un ou plusieurs groupes sous Linux.
- ▶ On peut configurer des autorisations de fichiers et d'autres privilèges par groupe.
 - ▶ sur Ubuntu, seuls les utilisateurs du groupe sudo peuvent utiliser la commande **sudo** pour obtenir des permissions élevées.
- ▶ La commande **groupadd** réservée à root permet d'ajouter un groupe.

root@toshiba-Satellite: /home/toshiba#**groupadd Gr1** // crée le groupe Gr1, Gr1 différent du nom d'un utilisateur ou fichier système existant

root@toshiba-Satellite: /home/toshiba#**groupadd Gr2** // crée le groupe Gr1

root@toshiba-Satellite: /home/toshiba#**cat /etc/group** //liste les utilisateurs avec leurs groupes

▶ Format de /etc/group

- ▶ **nom_de_groupe** : **champ_special** : **numero_de_groupe** : **utilisateurs**
 - **nom_utilisateur** correspond au nom du groupe.
 - **champ_special** liée aux anciennes versions d'Unix, n'est plus utilisé peut rester vide ou contenir le caractère * ou x.
 - **numero_de_groupe** identifiant unique du groupe sous la forme d'une valeur numérique
 - **utilisateurs** il s'agit de la liste des utilisateurs appartenant au groupe, ils sont séparés par des virgules

root:x:0:root

Gestion des utilisateurs

Ajouter un utilisateur à un groupe

- ▶ Par défaut un utilisateur est créé avec un groupe (**groupe primaire**) portant son nom. Chaque utilisateur doit faire partie au moins d'un groupe, son **groupe primaire**.
- ▶ On peut utiliser **useradd**, **adduser**, **usermod**

root@toshiba-Satellite: /home/toshiba#**adduser** Et1 Gr1 // ajoute l'utilisateur Et1 (**s'il existe déjà**) au Gr1 sans le supprimer des groupes auxquels il existe déjà

root@toshiba-Satellite: /home/toshiba#**adduser** --ingroup Gr2 Et1 // ajoute l'utilisateur Et1 (**s'il n'existe pas**) au Gr2

root@toshiba-Satellite: /home/toshiba#**useradd** -u 1000 -g 221 -d /home/L2i/Et1 Et1 //Ajoute l'utilisateur Et1 (**s'il n'existe pas**) d'identifiant 1000, dans le groupe 221 avec le répertoire de travail /home/L2i/Et1

root@toshiba-Satellite: /home/toshiba#**usermod** -g Gr1 Et1 // définit pour Et1 le groupe Gr1 comme groupe primaire

root@toshiba-Satellite: /home/toshiba#**usermod** -aG Gr2 Et2 // ajoute Et2 au groupe Gr2 sans supprimer Et2 de ses groupes d'origine

Droits d'accès sur les fichiers et dossiers

- ▶ Les fichiers appartiennent forcément à un utilisateur et à un groupe. Le groupe principal de l'utilisateur qui crée le fichier est, par défaut, le groupe propriétaire du fichier.
- ▶ Utiliser `ls l` pour consulter les droits d'accès
- ▶ Il y a 3 catégories d'utilisateurs d'un fichier :
 - ▶ l'utilisateur propriétaire du fichier ou user (u): c'est en principe l'utilisateur qui l'a créé.
 - ▶ Le groupe propriétaire (g): généralement le groupe principal du propriétaire du fichier mais peut être modifié par ce dernier et prendre la valeur d'un de ses groupes secondaires
 - ▶ Les **autres** ou others (o): toute personne autre que le propriétaire et qui n'est pas membre du groupe cité précédemment.
- ▶ Les « permissions » accordées à chaque utilisateur ou groupe d'utilisateurs sont de 3 types :
 - ▶ Permission en **lecture** : **R**ead (r)
 - ▶ Permission en **écriture** : **W**rite (w)
 - ▶ Permission d'**exécution** : **eX**ecute (x)

Propriétaire (u)			Groupe (g)			Autres (o)		
------------------	--	--	------------	--	--	------------	--	--

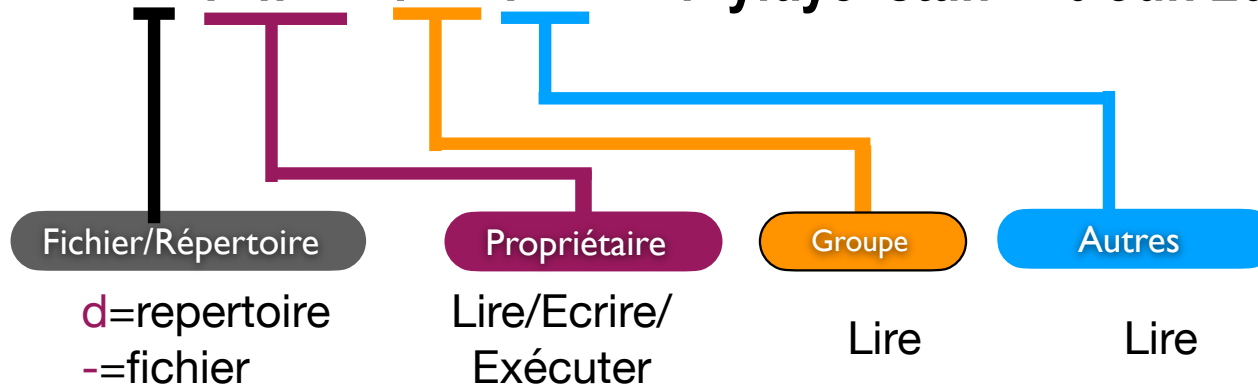
R	W	X	R	W	X	R	W	X
---	---	---	---	---	---	---	---	---

Droits d'accès sur les fichiers et dossiers

yfaye@MacBookProYFAYE: ~ \$ **ls -l**

d r w x r - x r - x 4 yfaye staff 128 Apr 30 16:05 Public

- r w - r - - r - - 1 yfaye staff 0 Jun 20 18:06 toto.txt



- Ces droits peuvent être écrits sous forme octale sur 3 bits, avec **x** le bit de poids faible et **r** le bit de poids fort.

Propriétaire (u)			Groupe (g)			Autres (o)		
R	W	X	R	W	X	R	W	X
4	2	1	4	2	1	4	2	1

$r*4+w*2+x*1$

► 664 = 1 1 0 1 0 0 1 0 0

► = RW- R- - R- - => rw pour u, r pour g et o

Droits d'accès sur les fichiers et dossiers

► Avec la commande **chmod** (change mode) permet de **changer** les permissions d'accès d'un fichier ou d'un répertoire.

► On indique l'utilisateur (u, g, o ou a), les actions "+, -, =" et le droit concerné "r, w, x".

```
yfaye@MacBookProYFAYE: ~ # chmod go +r toto //droit en lecture au groupe et aux autres
```

```
yfaye@MacBookProYFAYE: ~ # chmod u+rx toto.txt // donne les droits de lecture et d'exécution au propriétaire
```

```
yfaye@MacBookProYFAYE: ~ # chmod g+w,o-w toto.txt //donne le droit d'écriture au groupe et l'enleve aux autres
```

```
yfaye@MacBookProYFAYE: ~ # chmod +x toto.txt// donne le droit d'exécution à tout le monde
```

```
yfaye@MacBookProYFAYE: ~ # chmod u=rwx,g=r,o=- toto.txt //donne tous les droits au propriétaire, juste la lecture au groupe, rien aux autres
```

```
yfaye@MacBookProYFAYE: ~ # chmod ax toto.txt// (all ) enlever les droits d'exécution à tous
```

```
yfaye@MacBookProYFAYE: ~ # chmod uw toto.txt// enlever les droits d'écriture au propriétaire
```

```
yfaye@MacBookProYFAYE: ~ # chmod 600 toto.txt // donne les droits de lecture et écriture au propriétaire seul
```

```
yfaye@MacBookProYFAYE: ~ $ chmod 777 toto.txt // donne tous les droits à tous
```

Droits d'accès sur les fichiers et dossiers

- ▶ Un utilisateur ne disposant ni des permissions de **lecture**, ni d'**exécution** ne pourra pas accéder au contenu du dossier.
- ▶ Un utilisateur ne disposant que de la permission de **lecture** pourra lister le contenu du dossier. (Par exemple, avec la commande **ls** dans une fenêtre de terminal.) Il ne pourra pas accéder au dossier avec son navigateur de fichiers.
- ▶ Un utilisateur ne disposant que de la permission d'**exécution** peut ouvrir un dossier, mais ne peut pas en voir le contenu.
- ▶ Un utilisateur disposant des droits de **lecture** et d'**exécution** pourra lister le contenu d'un dossier et y entrer avec son navigateur de fichiers.
- ▶ Un utilisateur ne peut renommer, supprimer ou copier des fichiers dans un dossier s'il n'a pas accès en **écriture** à ce dossier
- ▶ Si un utilisateurs a accès en **écriture** à un répertoire, il peut supprimer un fichier même s'il ne dispose pas de droits d'**écriture** pour ce fichier.

Droits d'accès sur les fichiers et dossiers

- ▶ La commande **chown** (change owner) permet de **changer** le propriétaire ou le groupe
- ▶ La commande **chgrp** (change group) permet de **changer** le groupe

yfaye@MacBookProYFAYE: ~ # **chown** Et1 toto.txt // Et1 devient propriétaire du fichier toto.txt

yfaye@MacBookProYFAYE: ~ # **chgrp** Gr1 toto.txt // Gr1 devient groupe propriétaire de toto.txt

yfaye@MacBookProYFAYE: ~ # **chown** Et1:Gr1 toto.txt // Et1 et Gr1 deviennent propriétaires de toto.txt

Entrée, Sortie, Redirection

- ▶ > redirige dans un fichier la sortie standard de la commande.
 - ▶ yfaye@MacBookProYFAYE: ~ # echo bonjour L2i > toto.txt //met bonjour L2i dans le fichier toto.txt, le contenu antérieur de toto.txt est écrasé
 - ▶ yfaye@MacBookProYFAYE: ~ # ls -l > toto.txt //Le résultat **ls -l** ne s'affiche pas à l'écran, mais il est placé dans toto.txt
 - ▶ yfaye@MacBookProYFAYE: ~ # echo 2021 >> toto.txt //ajout sans écraser le contenu antérieur
- ▶ <: change l'entrée standard de la commande pour la prendre dans un fichier.
 - ▶ yfaye@MacBookProYFAYE: ~ # more < toto.txt //affiche page par page le contenu de toto.txt
- ▶ « | » branche la sortie standard de la commande de gauche sur l'entrée standard de la commande de droite
 - ▶ yfaye@MacBookProYFAYE: ~ # ls -l | more //affiche page par page la sortie de **ls -l**
- ▶ Qu'y a-t-il dans le fichier toto après l'exécution de la commande suivante: (echo "Bonjour" > toto; echo "Au revoir" > liste; ls -al >> toto) || echo "Bonjour" > toto (sur une seule ligne) ?
- ▶ Et si l'on remplace || par && ?