



Systemes d'exploitation

Utilisateurs, Groupes,
et Droits

L3 Info JFC ISM Dakar

Nicolas GARRIC



CENTRE
UNIVERSITAIRE
JEAN-FRANÇOIS
CHAMPOLLION



Sous UNIX

- UNIX est depuis le début multi-utilisateurs
- Sont donc définis :
 - La notion d'**utilisateur**
 - La notion de **groupe**
- Sont associés aux groupes et utilisateurs des droits :
 - Pour les fichiers
 - Pour les processus



Notion d'utilisateur

- Chaque utilisateur Linux est enregistré avec un **UID** (User Identification) unique (entier sur 16 bits).
- Le fichier **/etc/passwd** contient la liste des utilisateurs du système

```
tom:x:100:12:tom,,,:/home/tom:/bin/bash
```

```
paul:x:101:14:paul,,,:/home/paul:/bin/csh
```



Les champs dans /etc/passwd

- Username.
- Mot de passe encrypté ou « x ».
- UID
- GID : ID du groupe principal
- Nom complet ou autre description du compte.
- Répertoire d'accueil (home directory).
- Shell de login (programme lancé au login).



Créer/modifier des utilisateurs

- pour créer modifier des utilisateurs :
 - On modifie à la main **/etc/passwd** (root)
 - On utilise une interface d'administration comme **users-admin** (root)
- Un utilisateur peut :
 - modifier son mot de passe : commande **passwd**
 - modifier son shell de login : commande **chsh**



Notion de groupe

- Les utilisateurs sont organisés en groupes.
- Chaque groupe possède un **GID** (Group IDentification)
- Le fichier **/etc/group** contient la liste des groupes, leurs GID et la liste des utilisateurs associés à chaque groupe

`g1:x:100:tom,phlippe,henri,juliette`

- On utilise **groups** pour connaître la liste des groupes d'un utilisateur



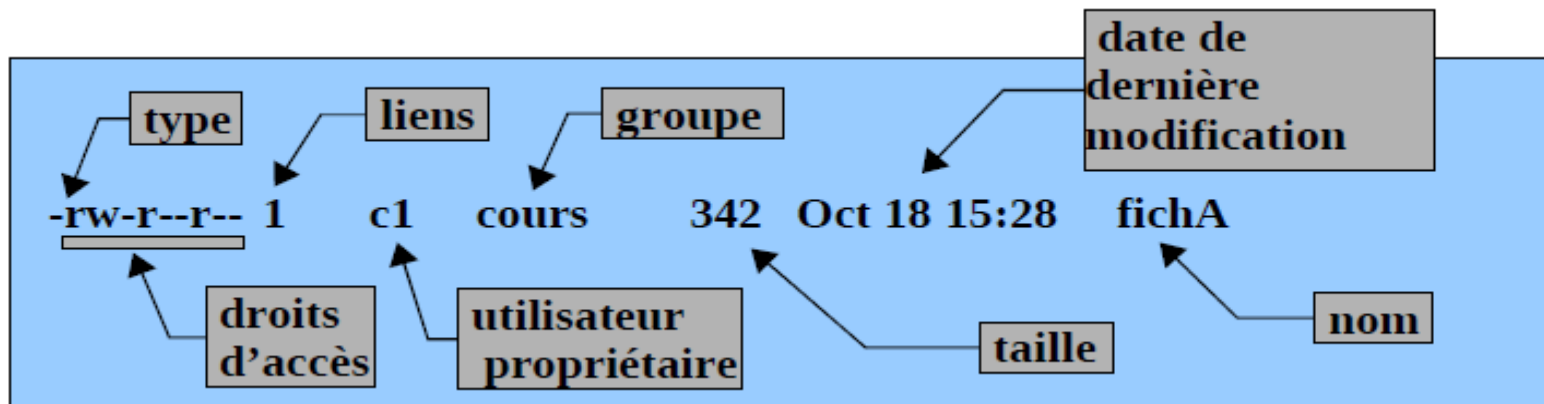
Propriétaire et groupe propriétaire

- Tout fichier possède un (utilisateur) **propriétaire** et un **groupe propriétaire**
- Quand un utilisateur crée un fichier :
 - Il est le **propriétaire** du fichier créé (on utilise **chown** pour changer le propriétaire... si on en a le droit)
 - Le **groupe propriétaire** du fichier créé est le groupe principal de l'utilisateur qui a créé le fichier (on utilise **chgrp** pour changer le groupe propriétaire d'un fichier)



Droits sur un fichier

- La commande **ls -l** donne :



Type d'un fichier

- Les différents type de fichiers sont :
 - 'd' pour **répertoire**,
 - '-' pour **fichier ordinaire**,
 - 'b' pour périphérique bloc,
 - 'c' pour périphérique caractère,
 - 'l' lien symbolique,
 - 'p' tube nommé (IPC),
 - 's' socket locale (IPC).



Droits d'accès

- Sur un fichier, il y a 3 groupes d'autorisation :
 - l'utilisateur **propriétaire**,
 - les personnes appartenant au **groupe propriétaire**
 - et les **autres utilisateurs**.
- Pour chaque groupe 3 caractères indiquent les autorisations : r, w et x (lecture, écriture et exécution)
 - rwxrw-r-- signifie : rwx pour le propriétaire, rw- pour le groupe propriétaire et r-- pour les autres





Droits d'accès **r** et **x**

- **r** = lecture : Autorisation de lire le contenu
 - Autorisation de lire un fichier.
 - Autorisation de lister le contenu d'un répertoire
- **x** = exécution/traverse :
 - Autorisation d'exécuter un fichier
 - Autorisation de traverser un répertoire



Droit d'accès **w**

- **w** = écriture : Autorisation de modifier le contenu ;
 - On peut écrire dans un fichier avec un éditeur ou une commande (cat).
 - Il faut ce droit pour le fichier destination d'une copie si le fichier existe déjà.
 - Pour un répertoire on peut créer et détruire des fichiers et des répertoires. Lors d'une destruction, seule cette permission prévaut.



Exemples

- `-rwx-----` : seul le propriétaire peut tout faire
- `-rw-r—r--` : tout le monde peut lire, seul le propriétaire peut écrire
- `drwxrw-rw-` : tout le monde peut lire et écrire, seul le propriétaire peut traverser le répertoire
- `-----rwx` : seuls les utilisateurs autres peuvent tout faire
bizarre !!!!



Modifier les droits : **chmod**

- **chmod** modifie les droits d'accès sur un fichier
- Personne concernée :
 - **u** (user), **g** (group), **o** (other), **a**(all)
- Action
 - **+** (ajouter), **-** (enlever), **=** (initialiser)
- Type d'accès
 - **r** (lecture), **w** (écriture), **x** (exécution/traverse)
- Exemple : **chmod go+x fichier**



Représentation octale

- Les permissions accordées peuvent être représentées en octal :

- | lettres | binaire | octal |
|---------|---------|-------|
| --- | 000 | 0 |
| --X | 001 | 1 |
| -W- | 010 | 2 |
| -WX | 011 | 3 |
| r-- | 100 | 4 |
| r-X | 101 | 5 |
| rw- | 110 | 6 |
| rwX | 111 | 7 |



Représentation octale

Exemples :

- `rwX --- ---` est représenté par `700`
- `r-X -w- --x` est représenté par ...
- `--- r-- rw-` est représenté par ...



/etc/shadow

-rw-r—r-- root root 1721 /etc/passwd

- Tout le monde a le droit de lecture sur /etc/passwd
- Donc on met les mots de passe chiffrés dans un autre fichier accessible uniquement par root

-rw-r----- root shadow 1721 /etc/shadow

- En fait il s'agit de hachage et pas de chiffrement





Changer d'identité

- La commande **su** permet de changer d'identité...
 - ...mais en héritant des attributs de l'utilisateur initial
- La commande **su -** permet de changer d'identité
 - En exécutant le `.bashrc` du nouvel utilisateur





root

- **root** est le compte administrateur du système
- Les permissions sont sans effet sur lui : root peut tout faire
- Sous ubuntu root n'a pas de mot de passe : on ne peut pas se connecter sous root.
- **sudo** est une commande permettant à un utilisateur d'exécuter des commandes nécessitant les droits root
- `/var/log/auth.log` contient les traces des commandes passées en sudo

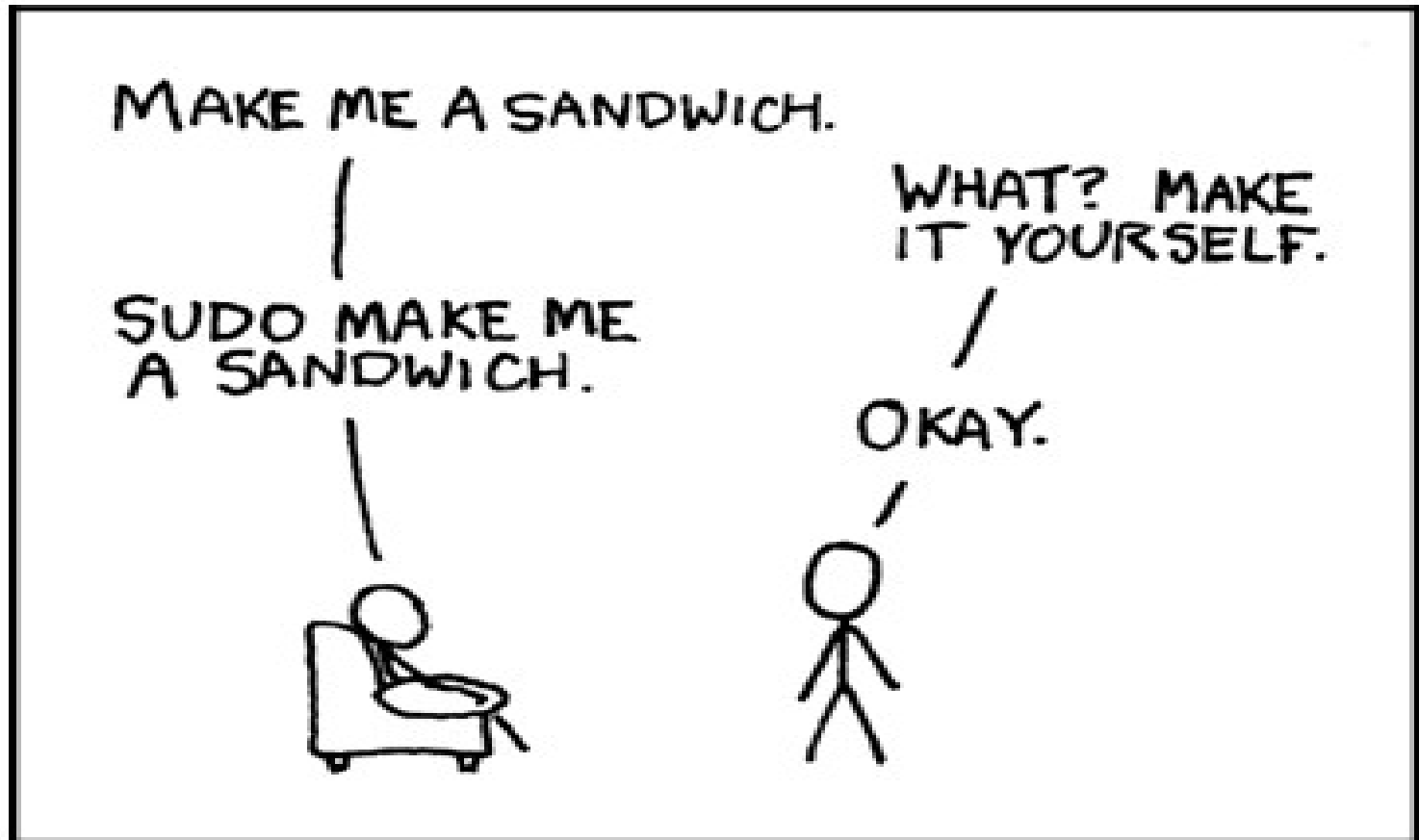


Pourquoi sudo ?

- Pour pirater un système ubuntu, il faut :
 - Connaître le nom des comptes permettant d'exécuter la commande **sudo** (disponible dans /etc/sudoers)
 - Puis trouver leur mot de passe
- Si root a un mot de passe, pour pirater le système, il suffit de :
 - Connaître le mot de passe root



Superutilisateur !



Source XKCD

