

# ADMINISTRATION UNIX



# ADMINISTRATION DES UTILISATEURS ET DES GROUPES



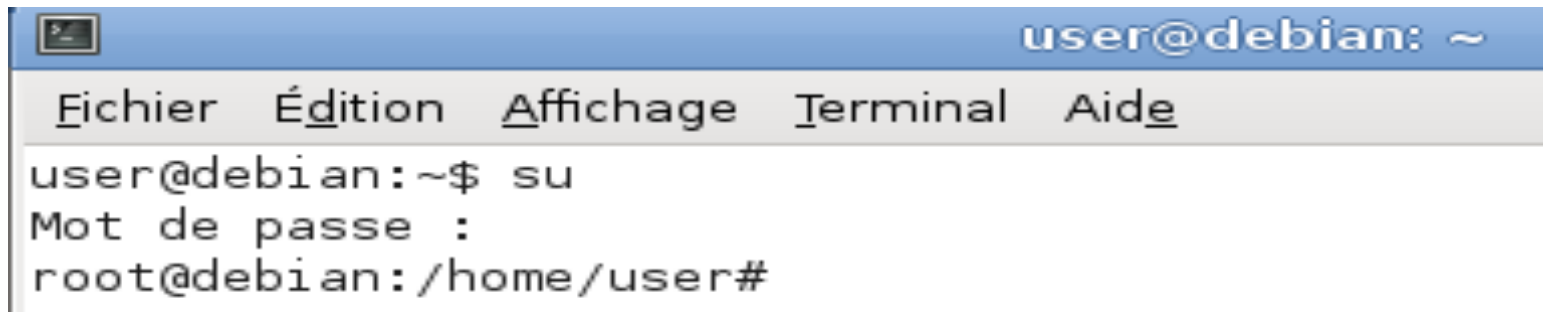
# GESTION DES UTILISATEURS

- Les comptes d'utilisateurs
  - Root:
    - Super utilisateur
    - Les permissions d'accès ne sont pas testées
    - Administrateur du système
  - ftp, bin, news...
    - Comptes utilisateurs utilisés par les applications
    - Ne peuvent pas (ne doivent pas) être utilisés comme login
  - Comptes ordinaires



# GESTION DES UTILISATEURS

- Comment être un root?
  - Le nom de l'administrateur du système linux est « root »
  - Pour avoir les privilèges de l'administrateur il faut se connecter en tant que « root »
    - Via le terminal administrateur en mentionnant le mot de passe de l'utilisateur « root »
    - En étant connecté déjà via un utilisateur simple et en utilisant la commande « **su** »
  - Le compte « root » est désactivé dans certaines distributions, le passage aux privilèges d super utilisateur se fait à travers la commande « **sudo** »



```
user@debian: ~  
Fichier  Édition  Affichage  Terminal  Aide  
user@debian:~$ su  
Mot de passe :  
root@debian:/home/user#
```



# GESTION DES UTILISATEURS

- Les comptes d'utilisateurs
  - Chaque utilisateur est reconnu par son login et son mot de passe
  - Un utilisateur peut appartenir à un ou plusieurs groupes
  - Les utilisateurs et les groupes sont repérés dans le système par des numéros : uid pour le numéro d'utilisateur (User IDentifier) et gid pour le numéro de groupe (Group IDentifier).
  - L'identification des utilisateurs se fait par le fichier `/etc/passwd`
  - L'identification des groupes se fait par le fichier `/etc/group`
  - Les mots de passes cryptés sont accessibles uniquement à l'administrateur via le fichier `/etc/shadow`



# GESTION DES UTILISATEURS

- Chaque utilisateur est une ligne dans le fichier /etc/passwd
- le fichier /etc/passwd contient la liste des utilisateurs système. Ce fichier est accessible en lecture à tous les utilisateurs et est de la forme suivante:

**login:passwd:uid:gid:comment:home:shell**

- Il contient:
  - Login: Nom de connexion (login) de l'utilisateur,
  - Passwd: Un caractère x: correspond au mot de passe de l'utilisateur
  - Uid: Le numéro de l'utilisateur (UID = user identifier),
  - Gid: Le numéro de groupe (GID = group identifier),
  - Comment: Commentaire textuel sur l'utilisateur qui peut contenir son nom réel, numéro de téléphone, etc.
  - Home: Le répertoire personnel de l'utilisateur,
  - L'interpréteur Shell par défaut de l'utilisateur (csh, sh, bash, etc.).



# GESTION DES UTILISATEURS

- Les commandes de gestion des utilisateurs
  - Changer l'identité de l'utilisateur connecté : **su**
    - **Syntaxe:** su [options] [nom\_utilisateur]
    - nom\_utilisateur est l'utilisateur dont on veut prendre l'identité
    - Si aucun utilisateur n'est spécifié le changement se fait vers l'utilisateur root
    - La commande demande un mot de passe avant de s'exécuter sauf pour le root
    - Exemple: su - test



# GESTION DES UTILISATEURS

- Les commandes de gestion des utilisateurs

- Créer un nouvel utilisateur: **useradd**

- **Syntaxe:** **useradd** [-c "commentaire"] [-d répertoire] [-e date\_expiration]  
[-g groupe\_primaire] [-G groupes\_secondaires]  
[-p mot\_de\_passe] [-s shell] [-u uid] **login**

- **Exemple:**

- `useradd -c "tril01" -d /home/stage -g reseau -G admin -p P@ssw0rd -s /bin/bash -u 501 etudiant`

- Modifier un utilisateur existant: **usermod**

- **Syntaxe:** **usermod** [-c "commentaire"] [-d répertoire] [-e date\_expiration]  
[-g groupe\_primaire] [-G groupes\_secondaires] [-p mot\_de\_passe] [-s shell] [-u uid]  
[-l nouveau\_login] **login**

- **Exemple:**

- `usermod -l student etudiant`

- `usermod -c "employé en essai" -d /home/employe -e 20141230 student`





# GESTION DES UTILISATEURS

- Les commandes de gestion des utilisateurs
  - Supprimer un utilisateur: **userdel**
    - **Syntaxe:** **userdel** [-r] login
    - -r: pour supprimer un utilisateur et aussi son répertoire personnel.
    - La commande **userdel** supprime toute trace de l'utilisateur dans le fichier de configuration : /etc/passwd y compris dans les groupes d'utilisateurs.
    - **Exemple:** **userdel mohamed** (supprime l'utilisateur mohamed et non pas son répertoire personnel).  
**userdel -r ali** (supprime l'utilisateur ali ainsi que son répertoire personnel).
  - Modifier le mot de passe d'un utilisateur: **passwd**
    - **Syntaxe:** **passwd** login
    - **Exemple:** **passwd etudiant**
      - passwd -d etudiant** désactive le mot de passe de l'utilisateur
      - passwd -u etudiant** supprime le mot de passe de l'utilisateur
      - passwd -l etudiant** active le mot de passe de l'utilisateur



# GESTION DES UTILISATEURS

- Les commandes de gestion des utilisateurs
  - Commande **id**
    - Affiche l'uid et le gid de l'utilisateur courant
  - Commande « **who** »
    - Affiche les utilisateurs connectés



# GESTION DES UTILISATEURS

- Le fichier `/etc/passwd` est public (toute personne qui a un compte sur la machine peut le lire).
- Pour contrecarrer cette faille, certains systèmes ont introduit le fichier `/etc/shadow`
  - lisible uniquement par root
  - contient les mots de passe des utilisateurs, qui disparaissent alors de `/etc/passwd`.
  - Si on ajoute un utilisateur à la main, cela implique d'éditer les 2 fichiers.



# GESTION DES UTILISATEURS

- Une ligne /etc/shadow est composée des champs suivants:
  - **Login.**
  - **Mot de passe** crypté. Une \* dans ce champ indique le compte ne peut être connecté (cas du compte **bin** par exemple). Un mot de passe commençant par !! indique que le compte est verrouillé (**désactivé**).
  - **Date de dernière modification** du mot de passe (en nombre de jour depuis le **1er janvier 1970** ).
  - **Période de changement** : Le nombre minimum de jours entre deux changements de mots de passe. Un **0** indique que l'utilisateur peut changer le mot de passe à n'importe quel moment.
  - **Durée de validité** : Le nombre maximum de jours pendant lesquels le mots de passe est valide (depuis 1/1/1970). La valeur **99999** indique que le mot de passe est toujours valide.
  - **Durée d'alerte** : nombre de jours pour lequel un utilisateur est averti que son mot de passe expirera.
  - Le nombre de jours après quoi le mot de passe expire de sorte que le compte est désactivé. -1 est utilisé pour indiquer un nombre infini de jours
  - **Date d'expiration** : Exprimée en nombre de jour depuis la date de référence (1/1/70)
  - **Exemple:**  
jack:Q,Jpl.or6u2e7:10795:0:99999:7:-1:-1:134537220



# GESTION DES GROUPES

- Chaque utilisateur doit faire partie de au moins un groupe
- Dans /etc/passwd chaque utilisateur possède un groupe par défaut
- Le fichier /etc/group rassemble la liste des groupes.
- Un groupe d'utilisateurs rassemble un certain nombre d'utilisateurs pouvant facilement partager des fichiers.
- Le fichier /etc/group est de la forme suivante:

**nom\_du\_groupe:mot\_de\_passe:GID:liste\_utilisateurs**

- Il contient:
  - Nom du groupe,
  - Un mot de passe du groupe (vide ou X)
  - Un numéro du groupe (GID = group identifier),
  - [ liste des utilisateurs membres du groupe ].



# GESTION DES GROUPES

- Les commandes de gestion des groupes
  - Pour lister tous les groupes d'un utilisateur :
    - **groups** *nom\_utilisateur*
  - Pour créer un nouveau groupe
    - **Syntaxe groupadd** [-g gid] *nom\_groupe*
    - **Exemple:** groupadd -g 1220 ensias  
groupadd ensias
  - Supprimer un groupe :
    - **Syntaxe groupdel** *nom\_groupe*
    - Le groupe est supprimé du fichier /etc/group.
  - modifier un groupe
    - **groupmod** [-g gid] [-n nouveau\_nom\_groupe] *nom\_groupe*
  - Ajouter ou supprimer un utilisateur d'un groupe: **gpasswd**
    - gpasswd -a etudiant ensias (Ajoute l'utilisateur stagiaire au groupe reseau).
    - gpasswd -d etudiant ensias (Supprime l'utilisateur stagiaire du groupe reseau).



# GESTION DES GROUPES

- Les commandes de gestion des groupes
  - Il y'a trois possibilités pour appartenir à un groupe :
    - Le groupe est votre groupe initial (principal) défini dans /etc/passwd.
    - Le groupe est un de vos groupes supplémentaires définis dans /etc/group.
    - Vous connaissez le mot de passe associé au groupe et vous vous connectez à ce groupe grâce à la commande **newgrp**.



# GESTION DES DROITS





# PROTECTION DES FICHIERS

- Le système Linux est un système multi-utilisateurs où l'accès aux fichiers est contrôlé par des droits.
- La commande ***ls -l*** permet de les afficher.
- Pour contrôler l'accès à un fichier, le système UNIX divise les utilisateurs en quatre catégories:
  - **u** : le propriétaire (user)
  - **g** : le groupe (group)
  - **o** : les autres (others)
  - **a** : user, group et other (all)
- ainsi que quatre types de droits
  - **r** : lecture (read)
  - **w** : écriture (write)
  - **x** : exécution
  - **-** : aucun droit



# PROTECTION DES FICHIERS

- Pour un répertoire
  - Lecture: Afficher le contenu du répertoire (ls)
  - Ecriture: créer supprimer un fichier dans le répertoire (vi, nano, rm, etc.)
  - Exécution: traverser (cd)
- Pour un fichier
  - Lecture: Afficher le contenu (more, less, cat, etc.)
  - Ecriture: modifier le contenu
  - Exécution: exécuter le contenu



# PROTECTION DES FICHIERS

- si vous essayez de copier un fichier dans un sous-répertoire (**cp fich1 rep2/rep3**),
- il faut disposer des droits suivants:
  1. lire le fichier *fich1* : droit en lecture sur le fichier
  2. traverser les sous-répertoires *rep2* et *rep3* : droit x au moins pour vous,
  3. écrire dans *rep3* : droit W



# PROTECTION DES FICHIERS

- La modification des droits:

- Commande **chown**

- Permet de modifier le propriétaire et le group associés au fichier

- **Syntaxe** : `chown nouveau_propriétaire fichier`

- `chown nouveau_propriétaire.nouveau_group fichier`

- `chown nouveau_propriétaire. fichier` //dans le cas ou le propriétaire et le groupe ont le même nom

- Pour changer le propriétaire d'un répertoire et de ses sous-répertoires, on utilise l'option `-R`

- **Exemple**: `chown ali test.txt`

- Commande **chgrp**

- **Syntaxe** : `chgrp nouveau_group fichier`

- L'option `-R` permet d'utiliser la récursivité sur les répertoires

- **Exemple**: `chown reseau test.txt`



# PROTECTION DES FICHIERS

- La modification des droits : `chmod`
  - Pour un fichier les droits sont exprimés par une chaîne de 10 caractères :
    - `tuuugggooo`
    - `t` : type du fichier
    - `uuu` : droits du propriétaire
    - `ggg` : droits du groupe
    - `ooo` : droits des autres
    - **Exemple** : `-rwxr-x---`  
`dr-xrwxr--`
  - Pour modifier le droit d'accès d'un fichier donné, utiliser la commande `chmod` (voir man `chmod`).
    - `chmod u+r toto` donne le droit de lire le fichier `toto` à l'utilisateur (vous même)
    - `chmod g+w toto` autorise une personne du même groupe que vous à lire le fichier
    - `chmod o-x toto` les autres ne sont pas autorisés à exécuter le fichier
    - `chmod a+rx toto` autorise le propriétaire, le groupe et les autres à lire et exécuter le fichier.
    - `chmod u=rwx,g=rx,o=- toto` fixe `r,w` et `x` pour propriétaire puis `r` et `x` pour le groupe et ne donne aucun droit aux autres.



# PROTECTION DES FICHIERS

- Représentation des droits en octal

**Syntaxe** *chmod nombre\_en\_base\_8 fichier1 [fichier2 ...]*

- *chmod* utilise le codage binaire pour modifier les droits en notation numérique:
  - lecture : 4,
  - écriture : 2,
  - exécution : 1,
  - pas de permission : 0.
- Chaque triplet se code par l'addition de 4, 2, 1, ou 0.
  - Pour un rwx il faudra ajouter  $4+2+1=7$ ,
  - pour r-x  $4+0+1=5$  etc.
- La commande *chmod* permettant de positionner *rwxr-x---* sur *fich1* à la syntaxe suivante : ***chmod 750 fich1***



# PROTECTION DES FICHIERS

- Représentation des droits en octal
  - Exemple:
    - `chmod 600 fichier => donner: rw----- à fichier`
    - `chmod 644 fichier => donner: rw-r--r- à fichier`
    - `chmod 750 fichier => donner : rwxr-x--- à fichier`
    - `chmod 777 *`        `=> donner : rwxrwxrwx à tous les fichiers`



# PROTECTION DES FICHIERS

- Valeur par défaut des droits d'accès: **umask**

**Syntaxe:** `umask nombre_en_base_8`

- La commande `umask` permet de définir les **droits d'accès par défaut** qui vont être donnés à un fichier ou à un répertoire après la création.

- **Exemple: `umask 026`**

- Les droits qui seront affectés à un **fichier** crée sont:

**666-026=640**      **(rw-r-----)**

- Les droits qui seront affectés à un **répertoire** crée sont:

**777-026=751**      **(rwxr-x—x)**





# DROITS D'ACCÈS SPÉCIAUX: STICKY BIT

- Sticky bit: C'est un droit qui s'ajoute aux autres.

(-rwx r-x r-**t**)

- Pour **un fichier** (exécutable), le **sticky bit** indique que lorsqu'on exécute le fichier, signifie qu'il reste en mémoire même après la fin de son exécution, pour pouvoir être relancé plus rapidement.
- Pour **un répertoire**, seul le propriétaire a le droit de supprimer les fichiers qu'il contient. Néanmoins, il est toujours possible pour un utilisateur possédant les droits d'écriture sur un fichier de le modifier.
- **Notation numérique**: on ajoute **1000** aux droits.
- **Exemple**: `chmod 1755 fich ou rep (rwx r-x r-t )`



# DROITS D'ACCÈS SPÉCIAUX: STICKY BIT

- **Notation symbolique:** on ajoute la lettre **t** aux autres.

- **Exemple:** Prenons comme exemple le répertoire /tmp.

```
$ ls -ld /tmp
```

```
drwxrwxrwx 6 root system 16384 Aug 14 13:22 tmp
```

```
$ chmod o+t /tmp ou $ chmod 1777 /tmp
```

```
$ ls -ld /tmp
```

```
drwxrwxrwt 6 root system 16384 Aug 14 13:22 tmp
```



# DROITS D'ACCÈS SPÉCIAUX: SUID BIT

- Suid bit: C'est un droit qui s'ajoute au **propriétaire**.

(-rws r-x r-x)

- Pour **un fichier** (exécutable), le **suid bit** indique que n'importe quel utilisateur peut lancer le fichier avec les droits du propriétaire.
- Pour **un répertoire: le bit suid n'existe pas.**
- **Notation numérique:** on ajoute **4000** aux droits.
- **Exemple:** `chmod 4755 fich`

(rws r-x r-x )

- **Notation symbolique:** on ajoute **la lettre s** au propriétaire.
- **Exemple:** `chmod u+s fich`

(rws r-x r-x )



# DROITS D'ACCÈS SPÉCIAUX: GUID BIT

- Guid bit: C'est un droit qui s'ajoute au **groupe**.

(-rwx r-**s** r-x)

- Pour **un fichier** (exécutable), le **guid bit** indique que n'importe quel utilisateur peut lancer le fichier avec les droits du groupe.
- Pour **un répertoire**: tous les fichiers créés au sein de ce répertoire, et quel que soit le groupe de la personne créant ce fichier, seront du même groupe que ce répertoire.
- **Notation numérique**: on ajoute **2000** aux droits.

- **Exemple**: `chmod 2755 fich ou rep`

(rwx r-**s** r-x )

- **Notation symbolique**: on ajoute **la lettre s** au groupe.
- **Exemple**: `chmod g+s fich ou rep`

(rwx r-**s** r-x )

