

# Gestion des utilisateurs sous Linux

[S.berraho@emsi.ma](mailto:S.berraho@emsi.ma)  
[Berraho.sanae@gmail.com](mailto:Berraho.sanae@gmail.com)

- Introduction
- Comptes utilisateurs
- Fichiers d'enregistrement
- Gestion des utilisateurs et groupes à travers les fichiers
- Gestion à travers les commandes
- Gestion à travers les GUI
- Droits d'accès
- Changement de propriété

## Gestion des utilisateurs sous Linux

### Introduction (1)

Linux est un système d'exploitation **multi-utilisateurs**.

- ⇒ il doit être possible d'**identifier** et/ou d'authentifier les utilisateurs du système
- ⇒ il doit être possible d'**exécuter des processus** appartenant à plusieurs utilisateurs **simultanément** et de déterminer quel utilisateur est responsable de chaque opération
- ⇒ le système d'exploitation doit fournir des mécanismes simples qui permettent de **contrôler l'accès** aux données des différents utilisateurs

## Gestion des utilisateurs sous Linux

### Introduction (2)

- **Gérer:**
  - ⇒ **Ajouter** des utilisateurs et des groupes
  - ⇒ **Modifier** les attributs des utilisateurs et groupes
  - ⇒ **Supprimer** les utilisateurs et groupes
- **Outils utilisés:**
  - ⇒ Les **commandes** du système : useradd, userdel, groupadd, groupdel, ...
  - ⇒ Agir directement sur les **fichiers de configuration**
  - ⇒ L'**interface graphique**

## Gestion des utilisateurs sous Linux

### Comptes utilisateurs

- Chaque utilisateur a un **nom** (login) et un **UID** (User Identifier)
- Les comptes utilisateur **ne sont pas tous égaux** sur Linux. On distingue trois types :
  - ⇒ **Super-utilisateur (root)**
  - ⇒ **Comptes systèmes**
  - ⇒ **Comptes ordinaires**

# Gestion des utilisateurs sous Linux

## Types de comptes utilisateurs (1)

- **Super-utilisateur (root):** c'est l'utilisateur le plus important du système du point de vue de l'administration.
  - ⇒ Compte unique qui dispose de **toutes les autorisations**
  - ⇒ Son identifiant est **root**,
  - ⇒ Il n'est **pas concerné par les droits d'accès aux fichiers**.
  - ⇒ Son **UID** égal à **0** (zéro) lui confère sa spécificité.
  - ⇒ Il a à sa charge les tâches **d'administration du système**.
- **Comptes ordinaires:** Tous les autres comptes utilisateur sont **associés à des personnes**;
  - ⇒ leur vocation est de permettre à **des utilisateurs standard de se connecter**.
  - ⇒ L'**UID** d'un utilisateur sera un nombre supérieur ou égal à **1000**

# Gestion des utilisateurs sous Linux

## Types de comptes utilisateurs (2)

- **Comptes système:** On trouve sur le système toute une série de comptes qui **ne sont pas affectés à des personnes** (bin, daemon, sync, apache...).
- Les comptes système sont des comptes **créés par le système** a des **services installés** dans votre système.
  - ⇒ Ils servent à faire **fonctionner certains services** de votre système comme l'impression, le son, la numérisation de documents
  - ⇒ Ceux-ci servent à faciliter la **gestion des droits d'accès de certaines application** et démons.
  - ⇒ Les **UID** compris entre **1 et 999** sont généralement utilisés pour ces comptes

# Gestion des utilisateurs sous Linux

## Principes (1)

- Chaque utilisateur est décrit par:
  - ⇒ Nom de connexion (Username)
  - ⇒ Mot de passe (Password)
  - ⇒ Numéro identificateur (User ID) unique entre 0 et 65535
  - ⇒ Groupe primaire (Group ID)
  - ⇒ Groupes secondaires
  - ⇒ Nom complet et autres informations
  - ⇒ Répertoire de travail (Home Directory)
  - ⇒ Shell de connexion (Login Shell)
  - ⇒ Date d'expiration du compte
  - ⇒ Durée de validité du mot de passe

# Gestion des utilisateurs sous Linux

## Principes (2)

- Sous Linux, **un groupe** est un ensemble d'utilisateurs qui **partagent les mêmes fichiers et répertoires**.
- Chaque groupe est caractérisé par:
  - ⇒ Un Nom (Group Name)
  - ⇒ Un numéro identificateur (Group ID)
  - ⇒ La liste de ses membres
  - ⇒ Éventuellement un mot de passe
  - ⇒ Éventuellement un administrateur du groupe

# Gestion des utilisateurs sous Linux

## Principes (3)

- Les **deux types de groupes** auxquels un utilisateur peut appartenir sont les suivants :
  - ⇒ **Groupe principal** : groupe assigné par le système d'exploitation aux fichiers créés par l'utilisateur. Chaque utilisateur **doit appartenir à un groupe principal** (par défaut le nom de l'utilisateur est aussi le nom de son groupe principal lors de sa création).
  - ⇒ **Groupes secondaires** : groupes auxquels un utilisateur peut appartenir. Les utilisateurs peuvent appartenir à plusieurs groupes secondaires.
- Pour cela, il existe **un fichier** qui comporte les noms des groupes existants dans votre système.

# Gestion des utilisateurs sous Linux

## Fichiers d'enregistrement

- La **gestion des utilisateurs** est assurée à l'aide de **4 fichiers de base**:
  - ⇒ **/etc/passwd** : La liste des utilisateurs avec leurs descriptions
  - ⇒ **/etc/shadow** : Les mots de passe stockés d'une manière cryptée + leurs durées de validité
  - ⇒ **/etc/group** : Définitions des groupes
  - ⇒ **/etc/gshadow** : Les mots de passe des groupes (rarement utilisé)

# Gestion des utilisateurs sous Linux

## Fichier de gestion des utilisateurs (1)

- L'ensemble des informations qui caractérisent les **paramètres fondamentaux de la connexion d'un utilisateur**, y compris de l'utilisateur root, sont définies dans le fichier **/etc/passwd**.
- C'est ce **fichier que le système consulte** lorsque vous vous connectez à votre compte en tapant votre identifiant et mot de passe.
- Accessible **en lecture seule** à **tout le monde**
- Accessible **en lecture/écriture** à l'utilisateur **root**
- Le fichier **/etc/passwd** est un fichier de type texte dont chaque ligne définit un compte utilisateur.
- La ligne est composée de **champs**. Le séparateur de champs est le caractère **":"**.

# Gestion des utilisateurs sous Linux

## Fichier de gestion des utilisateurs (2)

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

- login:** Il s'agit du nom de connexion de l'utilisateur.
- passwd:** contenait le mot de passe codé de l'utilisateur en question. Actuellement ce champ contient la seule lettre **x** pour signaler la présence du fichier **/etc/shadow**
- uid:** Il s'agit d'un numéro unique de l'utilisateur.
- gid:** Numéro de groupe. Chaque utilisateur appartient à un groupe primaire. Il pourra également appartenir à des groupes secondaires.
- comment:** Nom complet de l'utilisateur.
- home:** Chemin complet du répertoire représentant l'espace de travail privé attribué à l'utilisateur
- shell:** Chemin complet du shell de connexion.

## Gestion des utilisateurs sous Linux

### Extrait du fichier /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
.....
.....
samir:x:501:501:Samir
Amiss:/home/depinfo/samir:/bin/bash
alia:x:502:501:alia hamza:/home/alia:/bin/bash
```

## Gestion des utilisateurs sous Linux

### Fichier de gestion des mots de passes (1)

- Le fichier des mots de passe cryptés **/etc/shadow** est exclusivement utilisé lors de l'étape d'authentification.
- Il **n'est pas lisible par tous** et aucun utilisateur normal n'a le droit de voir le champ **``mot de passe chiffré``**
- Chaque utilisateur possède une ligne dans ce fichier

## Gestion des utilisateurs sous Linux

### Fichier de gestion des mots de passes (2)

**login : passwd : last : may : must : warn : inactif : disable :**

- **login :** Le nom de connexion de l'utilisateur.
- **passwd :** Le mot de passe encodé
- **last:** Nombre de jours depuis la modification du mot de passe (Depuis 1/1/1970).
- **may:** Le nombre minimum de jours pendant lesquels le mot de passe ne peut pas être changé.
- **must:** C'est le nombre maximum de jours avant que le mot de passe n'expire (Le mot de passe doit être changé après ce nombre de jours)
- **warn:** Nombre de jours durant lesquels l'utilisateur est prévenu de l'expiration de son mot de passe.
- **inactif:** la période d'inactivité maximale autorisée, après laquelle le compte est désactivé.
- **disable :** Date de la fermeture du compte (en nombre de jours depuis 1/1/1970).

## Gestion des utilisateurs sous Linux

### Extrait du fichier /etc/shadow

```
root:$1$ryiKUDlj$Quh56ZLg15C0ZCt1:12668:0:99999:7:::
bin:*:12668:0:99999:7:::
daemon:*:12668:0:99999:7:::
adm:*:12668:0:99999:7:::
lp:*:12668:0:99999:7:::
shutdown:*:12668:0:99999:7:::
mail:*:12668:0:99999:7:::
....
samir:$1$CdC9mrQu$OPB7u9D0ig1IHT0:12674:0:99999:7:::
alia:$1$Fr47ytUy$O.997i6anig1Igi9:12683:0:99999:7:::
```

## Gestion des utilisateurs sous Linux

### Fichier de gestion des groupes

- **/etc/group** contient la description de l'ensemble des **groupes** définis dans le système
- Chaque groupe est décrit par les champs suivant:

**group : passwd : gid : membres**

- **group :** Le nom du groupe en question.
- **passwd :** Le mot de passe du groupe.
- **gid :** Le numéro identificateur du groupe
- **membres:** Liste les membres séparés par des virgules.

## Gestion des utilisateurs sous Linux

### Extrait du fichier /etc/group

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
lp:x:7:daemon,lp
sshd:x:74:
users:x:100:samir, ali, bouchra
...
ginfo:x:501:
musiq:x:601:samir,alia
```

## Gestion des utilisateurs sous Linux

### Fichier des mots de passes de groupes

- Le fichier **/etc/gshadow** contient principalement les mots de passe des groupes
- Chaque ligne contient les champs suivant :

**group : passwd : administrateurs : membres**

- **group :** Le nom du groupe en question.
- **passwd :** Le mot de passe du groupe sous format crypté.
- **administrateurs :** Liste des administrateurs du groupe
- **membres :** Liste les membres séparés par des virgules

## Gestion des utilisateurs sous Linux

### Extrait du fichier /etc/gshadow

```
rpcuser:!:
nfsnobody:!:
gnome-initial-setup:!:
sshd:!:
slocate:!:
avahi:!:
postdrop:!:
postfix:!:
tcpdump:!:
bsanae:!:bsanae
vboxsf:!:
yumex:!:
wireshark:!:
IT:$6$sDXnqDSnIx/$t6PD00Wza5GtfunIDB86/
CheKJUx2nP7AyZCHSaS1KYAZ4cvk5sWL.XdMvge
RJTstNstquge0HGON3/5Dpeoe0: :
```

# Gestion des utilisateurs sous Linux

## Gestion à travers les fichiers

- Les 4 fichiers précédents peuvent être **édités manuellement** pour créer ou modifier les **utilisateurs** et les **groupes**.
- Dans ce cas les liens entre ces fichiers devront être respectés
- Les commandes **vipw** et **vigr** (sans arguments) doivent être utilisées (à la place de votre éditeur de texte préféré) pour éditer respectivement les 2 couples de fichiers : (**/etc/passwd**, **/etc/shadow**) et (**/etc/group**, **/etc/gshadow**)
- vipw** et **vigr** verrouillent les fichiers correspondant pour interdire aux utilisateurs connectés de modifier les attributs de leurs comptes (mot de passe, shell, ...) au moment de l'édition.

# Gestion des utilisateurs sous Linux

## Commande de création des utilisateurs (1)

**Useradd** [**<options>**] **<nom d'utilisateur>**

### [<options>]

- u** : Spécifier manuellement le UID du compte
- c** : Commentaire (généralement le nom complet de l'utilisateur)
- d** : Répertoire personnel
- D** : Afficher ou modifier les valeurs par défaut
- e** : Date d'expiration (AAAA-MMM-JJ)
- g** : Groupe primaire (par défaut un nouveau groupe est créée avec le même nom)
- G** : Groupes secondaires (séparés par des virgules)
- m** : Le répertoire personnel sera crée s'il n'existe pas
- M** : Le répertoire personnel ne sera pas crée (même s'il existe dans le fichier des valeurs par défaut)
- s** : Spécifier le type de shell
- r** : Créer un compte système

# Gestion des utilisateurs sous Linux

## Commande de création des utilisateurs (2)

```
# useradd user1 -c "utilisateur 1" -d /home/user1 -m -g guser -s /bin/bash
```

- Après sa création en utilisant la commande **useradd**, le nouveau compte n'est **pas encore accessible**
- L'administrateur système doit tout d'abord lui **assigner un mot de passe**

```
$ sudo passwd user1
```

# Gestion des utilisateurs sous Linux

## Options par défaut de la commande useradd (1)

- Lors de la création d'un utilisateur, si les options ne sont pas précisées, le système utilise les valeurs par défaut définies dans:

⇒ **/etc/login.defs**

⇒ **/etc/default/useradd**

- Pour consulter les options par défauts:

```
# useradd -D
```

```
[root@localhost bsanae]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

- Les options par défaut peuvent être changées de deux manières:
  - ⇒ modifier directement les fichiers
  - ⇒ Utiliser la commande **useradd** avec l'option **-D**

## Gestion des utilisateurs sous Linux

### Options par défaut de la commande useradd

⇒ Fichier **/etc/default/useradd**

Valeur	Commentaire
<b>GROUP</b>	Groupe par défaut.
<b>HOME</b>	Chemin dans lequel le répertoire de connexion du nom de l'utilisateur sera créé.
<b>INACTIVE</b>	Nombre de jours suivant l'expiration du mot de passe avant que le compte ne soit désactivé.
<b>EXPIRE</b>	Date d'expiration du compte.
<b>SHELL</b>	Interpréteur de commandes.
<b>SKEL</b>	Répertoire squelette du répertoire de connexion.
<b>CREATE_MAIL_SPOOL</b>	Création de la boîte aux lettres dans /var/spool/mail.

## Gestion des utilisateurs sous Linux

### Options par défaut de la commande useradd

⇒ Fichier **/etc/login.defs**

Ce fichier contient de nombreux paramètres par défaut utiles aux commandes de création ou de modification d'utilisateurs.

- **UID\_MIN** : Indique le numéro UID le plus bas autorisé pour les comptes
- **UID\_MAX** : indique le nombre maximal que le système peut utiliser pour l'UID
- **SYS\_UID\_MIN** : UID minimum pour les comptes système
- **SYS\_UID\_MAX** : UID maximum pour les comptes système
- **PASS\_MAX\_DAYS** : Nombre maximum de jours de validité d'un mot de passe. Après cette durée, une modification du mot de passe est obligatoire
- **PASS\_MIN\_DAYS** : Nombre minimum de jours autorisé avant la modification d'un mot de passe. Toute tentative de modification du mot de passe avant cette durée est rejetée.
- **PASS\_MIN\_LENGTH** : nombre minimum de caractères requis dans le mot de passe
- **CREATE\_HOME** : Si défini sur oui, le répertoire personnel de l'utilisateur sera créé
- **ENCRYPT\_METHOD** : méthode utilisée pour hacher les mots de passe des comptes
- etc

## Gestion des utilisateurs sous Linux

### Dossier **/etc/skel**

- Lors de la création d'un utilisateur, son **répertoire personnel est créé**.
- Par défaut, tous les fichiers et répertoires placés dans **/etc/skel** seront copiés dans les répertoires personnels des utilisateurs lors de leur création.
- Par défaut, le répertoire **"**/etc/skel/**"** est vide. Ce qui se justifie par le fait que lorsqu'on crée un nouvel utilisateur, son « **répertoire personnel** » est vide.

## Gestion des utilisateurs sous Linux

### Modifier un utilisateur

- Pour modifier les propriétés d'un compte déjà créé, on peut utiliser la commande **usermod** dont le fonctionnement est très similaire à **useradd**.

Option	Description
<b>-m</b>	Associé à l'option -d, déplace le contenu de l'ancien répertoire de connexion vers le nouveau.
<b>-l login</b>	Nouveau nom.
<b>-e AAAA-MM-JJ</b>	Date d'expiration du compte.
<b>-L</b>	Verrouille le compte.
<b>-U</b>	Déverrouille le compte.
<b>-a</b>	Empêche la suppression de l'utilisateur d'un groupe secondaire lors de l'ajout dans un autre groupe secondaire.
<b>-G</b>	Précise plusieurs groupes secondaires lors de l'ajout

## Gestion des utilisateurs sous Linux

### Supprimer un utilisateur

- La commande qui permet de supprimer un utilisateur :

```
# userdel user1
```

- Si on désire aussi supprimer son dossier personnel :

```
# userdel -r user1
```

- La commande précédente supprime l'utilisateur **user1** ainsi que **son répertoire personnel**,
- cependant un problème demeure: **les fichiers appartenant à user1 et qui se trouvent en dehors du répertoire personnel ne sont pas supprimés.**
- La commande suivante permet de les trouver et de les supprimer à partir du **UID** de l'utilisateur (on suppose que le **UID** de l'exemple **user1** est **1002**)

```
# find / -type f -uid 1002 -print -exec rm {} \;
```

## Gestion des utilisateurs sous Linux

### Commande passwd

- La commande **passwd** permet de gérer un mot de passe (gérer divers paramètres dans le fichier **/etc/shadow**)

Option	Description
-d	Supprime le mot de passe.
-l	Verrouille le compte.
-S	Affiche le statut du compte.
-u	Déverrouille le compte.
-e	Fait expirer le mot de passe.
-n jours	Nbr de jours avant que le mot de passe puisse être changé
-x jours	Nbr de jours avant que le mot de passe doit être changé
-w jours	Délai d'avertissement avant expiration.
-i jours	Délai avant désactivation lorsque le mot de passe expire.

## Gestion des utilisateurs sous Linux

### Commande chage

- La commande **chage** permet de modifier les informations de validité du mot de passe

Option	Description
-d LAST_DAY	Positionne la date du dernier changement du mot de passe à LAST_DAY
-E Expire_Date	Modifie la date d'expiration du compte
-l jours	Délai avant désactivation lorsque le mot de passe expire (même effet que <b>-i</b> de la commande <b>passwd</b> )
-m jours	Nbr de jours avant que le mot de passe puisse être changé (même effet que <b>-n</b> de la commande <b>passwd</b> )
-M jours	Nbr de jours avant que le mot de passe doit être changé . (même effet que <b>-x</b> de la commande <b>passwd</b> )
-W jours	Délai d'avertissement avant expiration.
-l	Liste les informations liées aux dates et durées de validité

## Gestion des utilisateurs sous Linux

### Créer un groupe

- La commande **groupadd** permet d'ajouter un groupe au système.

```
groupadd [<options>] <nom du groupe>
```

Option	Description
-g GID	GID du groupe à créer.
-f	Le système choisit un GID si celui précisé par l'option -g existe déjà.
-r	Crée un groupe système avec un GID compris entre SYS_GID_MIN et SYS_GID_MAX. Ces deux variables sont définies dans <b>/etc/login.defs</b>



## Gestion des utilisateurs sous Linux

### Modifier un groupe

- La commande **groupmod** permet de modifier un groupe existant.

```
groupmod [<options>] <nom du groupe>
```

Option	Description
<b>-g GID</b>	GID du groupe à créer.
<b>-f</b>	Le système choisit un GID si celui précisé par l'option -g existe déjà.
<b>-r</b>	Crée un groupe système avec un GID compris entre SYS_GID_MIN et SYS_GID_MAX. Ces deux variables sont définies dans /etc/login.defs

## Gestion des utilisateurs sous Linux

### Supprimer un groupe

- La commande **groupdel** permet de supprimer un groupe existant.

```
groupdel <nom du groupe>
```

- Pour être supprimé, un groupe ne doit plus contenir d'utilisateurs

## Gestion des utilisateurs sous Linux

### Gérer un groupe

- La commande **gpasswd** permet de gérer un groupe

```
gpasswd [-a login] [-A login] [-d login] [-M login] groupe
```

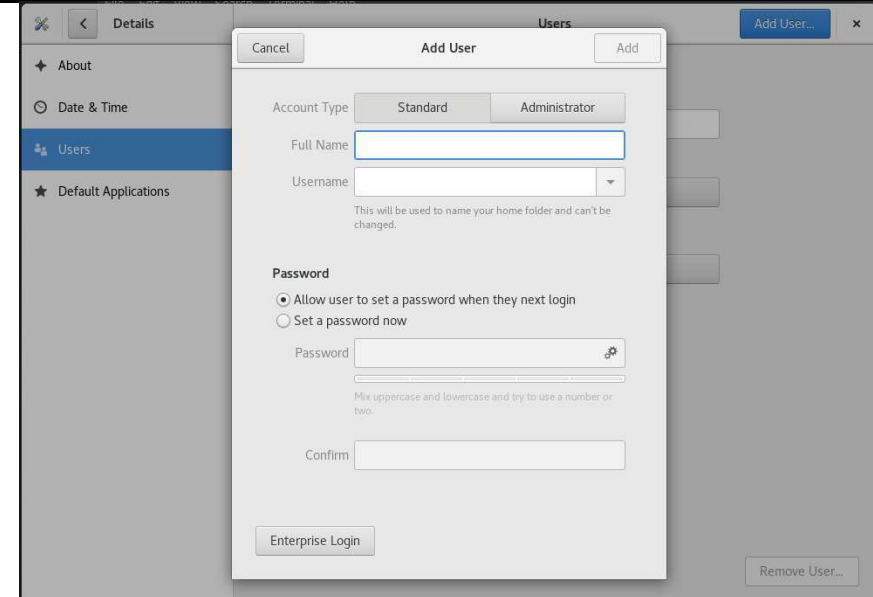
Option	Description
<b>-a login</b>	Ajoute l'utilisateur au groupe.
<b>-A login</b>	Définit l'administrateur du groupe.
<b>-d login</b>	Retire l'utilisateur du groupe.
<b>-M login</b>	Définit la liste de membres

- Définir le mot de passe pour le groupe

```
# gpasswd Groupe
```

## Gestion des utilisateurs sous Linux

### Les GUI (1)



# Gestion des utilisateurs sous Linux

## Les GUI (2)

- Permettent la majorité des **opérations habituelles** (ajout, suppression, modification)
- Simples et conviviales mais **limitées**
- Dépendent des distributions
- Pratiques si :
  - ⇒ On travaille avec une distribution en **interface graphique**
  - ⇒ On désire faire **une manipulation classique**

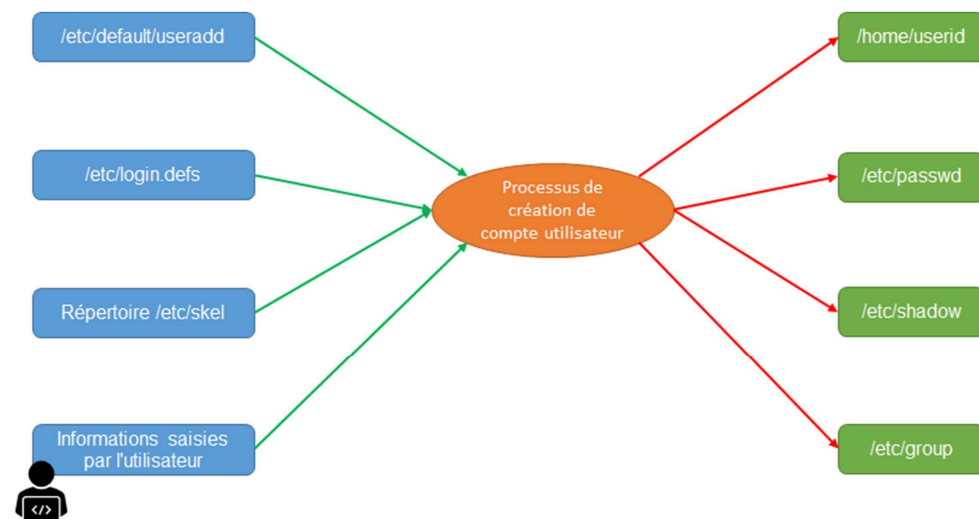
# Gestion des utilisateurs sous Linux

## Afficher les informations sur les utilisateurs et les groupes

- La commande **whoami** affiche le nom du compte dans lequel on travaille
- La commande **who** liste les utilisateurs connectés sur le même système
- La commande **groups** permet d'afficher les groupes auxquels appartient un utilisateur
- La commande **id** affiche les noms des groupes d'un utilisateur.

# Gestion des utilisateurs sous Linux

## Création des utilisateurs: Résumé des fichiers modifiés



# Gestion des utilisateurs sous Linux

## Les droits d'accès (1)

Droits d'accès	Sur les répertoires	Sur les fichiers
Lire (read ) (r)	Autorisation de voir le contenu d'un répertoire ou les sous-répertoires	Autorisation de voir le contenu du fichier.
Écrire (write) (w)	Autorisation de créer, modifier, supprimer les fichiers ou les sous répertoires.	Autorisation aux entités d'ajouter, de modifier, de supprimer le contenu d'un fichier
Exécuter (execute)(x)	Autorisation d'accéder au répertoire	Permettre d'exécuter le fichier
(-)	Pas d'autorisation	Pas d'autorisation

## Gestion des utilisateurs sous Linux

### Les droits d'accès (2)

- Ces permissions sont consultables complètement par la commande:

```
# ls -l
```

```
root@server1:/home/berraho/Bureau# ls -l
total 8
drwxr-xr-x 2 root root 4096 00:18 mon-repertoire
-rw-r--r-- 1 root root 7 00:18 test.txt
```

-	r	w	x	r	w	-	r	-	-	Avec:	-	Fichier ordinaire
d	r	w	x	r	w	-	r	-	-		d	Fichier répertoire
Type de fichier	Droits pour le propriétaire	Droits pour le groupe	Droits pour les autres								r	Read (lecture)
											w	Write (Ecriture)
											x	Exécution

## Gestion des utilisateurs sous Linux

### Changement des droits d'accès (1)

- On utilise **chmod** pour modifier les droits accordés aux utilisateurs en tant que **root** ou en tant que **propriétaire** d'un fichier/répertoire.
- Il existe deux façons d'utiliser **chmod**:
  - ⇒ **mode symbolique**
  - ⇒ **mode absolu**

## Gestion des utilisateurs sous Linux

### Changement des droits d'accès (2)

#### CHMOD en mode symbolique

Opérateur chmod	Signification	Exemple	Résultat
+	Ajouter les droits désignés à un fichier ou répertoire	<b>chmod o+wx mon_fichier.txt</b>	Ajout des droits de modification et d'exécution au autres utilisateurs
-	Supprimer les droits désignés à un fichier ou répertoire	<b>chmod u-x mon_fichier.txt</b>	Supprime le droit d'exécuter ce fichier pour le propriétaire
=	Attribuer exactement ces droits	<b>chmod g=r-x mon_fichier.txt</b>	Donne exactement les droits de lire et d'exécuter pour les utilisateurs du groupe

## Gestion des utilisateurs sous Linux

### Changement des droits d'accès (3)

#### CHMOD en mode absolu

##### Exemples:

```
-  r w x  r w -  r - -
   1 1 1  1 1 0  1 0 0
   7   6   4
```

- Pour affecter ces droits en mode absolu: **Chmod 764 fichier**

## Gestion des utilisateurs sous Linux

### Changement du groupe d'un fichier

#### La commande chgrp

- **chgrp** est utilisée pour **changer le groupe du fichier ou du répertoire**. Le changement de groupe peut être effectué par :
  - ⇒ **Le root**
  - ⇒ Le propriétaire du fichier si ce dernier est **parmi les membres du groupe en question**.
- **Syntaxe : chgrp [OPTIONS] nouveau\_groupe fichier/répertoire**
  - ⇒ **L'option -R** : Changer l'autorisation sur les fichiers qui sont dans les sous répertoires du répertoire en question.
- **Exemple:**

```
# chgrp informatique fichier1
```

## Gestion des utilisateurs sous Linux

### Changement du propriétaire d'un fichier (1)

#### La commande chown

- **chown** est utilisée pour changer **le propriétaire et/ou le groupe** propriétaire du fichier ou du répertoire.
- Le syntaxe de cette commande :  
**chown [-option] [utilisateur][:groupe] fichier [fichier1 fichier2 ..]**
- Elle peut être utilisée pour changer :
  - ⇒ Le propriétaire et le groupe,
  - ⇒ Seulement le propriétaire,
  - ⇒ Seulement le groupe (devient alors similaire à chgrp)
- Les options intéressantes :
  - ⇒ **-R** : Modifie tous ses sous-répertoires et ses sous-fichiers d'une manière récursive.

## Gestion des utilisateurs sous Linux

### Changement du propriétaire d'un fichier (2)

#### La commande chown

#### Exemples:

- Changer seulement le propriétaire

```
# chown user1 fichier1
```

- Changer seulement le groupe

```
# chown :informatique fichier1
```

- Changer le groupe et le propriétaire

```
# chown user1:informatique fichier1
```