Chapitre 4 - Utilisateurs et mécanismes de droits sur UNIX

2.0

Système d'exploitation GNU/Linux

Gorgoumack SAMBE
Université Assane Seck de Ziguinchor

1 Avril 2022

Table des matières

Objectifs

A l'issue de ce chapitre, l'apprenant doit être capable de :

1. distinguer les types de **comptes d'utilisateurs** sur un système UNIX ;
2. manipuler des comptes et personnaliser un **profil d'utilisateur** ;
3. manipuler les **groupes d'utilisateurs** ;
4. distinguer et manipuler les **droits** liés à un fichier;
5. configurer les **droits par défaut** des fichiers.

Introduction

Pour pouvoir utiliser un système UNIX, il faut disposer d'un **compte d'utilisateur** sur le système. UNIX étant multi-utilisateurs et multitâches, il est nécessaire que le système assure la sécurité des données et des processus des différents utilisateurs et aussi sa propre sécurité. Cela passe dans les système UNIX par la séparation des espaces des **comptes** et l'usage d'un système de **mécanisme de droits** sur les fichiers et processus.

Nous verrons dans ce chapitre :

1. les comptes et groupes d'utilisateurs sur UNIX ;
2. et les mécanismes de droit sur les fichiers.

I.Utilisateurs et groupes

La gestion de la sécurité du système et des données sur un système UNIX s'appuie sur la notion de **compte d'utilisateurs**, plusieurs types de compte existent sur un système UNIX. Les système UNIX intègrent aussi la notion de **groupe d'utilisateurs** qui permet à un ensemble d'utilisateurs d'avoir les **mêmes droits** sur un fichier. Des commandes systèmes sont à disposition pour effectuer l'ensemble des tâches relatives à cet aspect. La "base de données" des comptes, groupes et éléments de configurations est constituée de fichiers de configuration. Nous aborderons ces différents aspects dans cette section

1. Les types de comptes

Il existe **trois types de comptes** sur un système UNIX :

1. le compte d'**utilisateur racine (root)** ou de **super administrateur**;
2. les comptes d'**utilisateurs réguliers ;**
3. les **comptes de services**.

Un compte d'utilisateur porte un **nom (login)** et est identifié de manière unique sur le système par l'**identifiant utilisateur (User identifier -UID).**

Utilisateur racine (root)

L'**utilisateur racine**, **root** ou **super-administrateur** d'un système UNIX est l'utilisateur le plus puissant et est généralement créé lors du processus d'**installation.** Il détient un **pouvoir absolu** sur le système . Il peut accéder à toutes les commandes, fichiers et répertoires et modifier le système selon ses préférences. Il peut mettre à jour le système, installer et désinstaller des packages, ajouter ou supprimer d'autres utilisateurs, accorder ou révoquer des autorisations et effectuer tout autre tâche d'administration système sans aucune restriction.

Pour cette raison, exécuter des commandes en tant qu'utilisateur root est **fortement déconseillé**.

Les systèmes UNIX utilisent une **séparation des droits stricts** entre le compte “administrateur” (“root”) et les comptes d'utilisateurs. Le **répertoire personnel** du super-administrateur n'est pas dans le répertoire /home, c'est le répertoire **/root** qui peut être sur sa propre **partition** pour encore plus de sécurité.

Le **login** du super-administrateur est par défaut **root** et il lui est conseillé d'avoir un **mot de passe robuste** et une politique de changement régulier de mot de passe.

Utilisateur régulier

Un **utilisateur régulier** est un utilisateur de connexion normal qui peut être créé par un administrateur système. Certaines distribution comme ubuntu propose d'en créer un pendant le **processus d'installation**. Cependant, vous pouvez toujours créer **autant d'utilisateurs réguliers** que vous souhaitez après l'installation. Un utilisateur régulier ne peut effectuer que des tâches et accéder aux fichiers et répertoires pour lesquels il est autorisé. Le compte d'un utilisateur est représenté par le login et un mot de passe associé et il dispose d'un répertoire personnel (home directory) situé généralement dans le répertoire /home. Les utilisateurs réguliers peuvent également être supprimés ou désactivés en cas de besoin.

Si nécessaire, un utilisateur régulier peut se voir accorder des privilèges élevés pour effectuer des tâches de niveau administratif.

Compte de service

Il s'agit d'un **compte sans connexion** qui est créé lors de l'**installation d'un progiciel**. Ces comptes sont utilisés par les **services** pour **exécuter des processus** dans le système. Ils ne sont ni conçus ni destinés à effectuer des tâches de routine ou administratives dans le système. Dès que vous avez installé votre machine Linux, plusieurs comptes de service sont créés, ces comptes sont nécessaires pour sa gestion interne. Cette démarche participe à la sécurité du système.

Vous trouverez par exemple un compte mysql dans le fichier /etc/passwd lorsque vous installez un Système de Gestion de Bases de Données (SGBD) Mysql.

2. Groupes d'utilisateurs

Principes des groupes

Les systèmes UNIX intègrent aussi la notion de **groupe d'utilisateurs**, ce qui permet à un ensemble d'utilisateurs d'avoir les mêmes droits sur un fichier ou un répertoire. Un **groupe** possède un **nom** et est identifié de manière unique sur le système par son **identifiant de groupe (GID)**, il peut être **éventuellement vide**.

A la création d'un utilisateur (par la commande useradd), le système crée par défaut un groupe de même nom. **Chaque utilisateur** ainsi créé sur GNU/Linux appartient par défaut à **un groupe primaire** qui est par défaut le groupe portant son nom, il peut faire partie de **plusieurs groupes secondaires**. Son groupe primaire peut être modifié.

Au démarrage d'une **session**, l'utilisateur est connecté avec son compte d'utilisateur mais aussi avec un groupe appelé **groupe actif** ou groupe de connexion. Le **groupe actif** (ou groupe de connexion) de l'utilisateur est son groupe primaire, il peut activer un autre groupe (groupes secondaires) s'il le souhaite, les objets (fichiers et répertoires) créé dans une session sont la **propriété de l'utilisateur** mais aussi du **groupe actif**.

Le groupe sudo

Dans **Debian** et les **distributions dérivées**, il existe un groupe d'utilisateur nommé **sudo** (on les appelle les sudoers) qui possèdent des privilèges élevés pour effectuer des **tâches d'administration**. ils doivent faire précéder la commande d'administration par la commande sudo. Par exemple :

$ sudo adduser moussa

pour ajouter un utilisateur moussa.

Reportez vous à la section 5 des manpages pour plus d'information sur le plugin sudoers et le fichier de configuration /etc/sudoers.

3. Les commandes de gestion des comptes et groupes

Les commandes de manipulation des comptes sont les suivantes :

1. **useradd** :ajout d'un nouvel utilisateur/modification des informations par défaut pour les nouveaux utilisateurs
2. **userdel** :suppression d'un utilisateur et des fichiers associés
3. **usermod** : modifications d'un utilisateur
4. **lslogins** : affichage des informations sur les comptes (liste des comptes ou informations d'un compte).
5. **passwd** : modifier le mot de passe d'un utilisateur

Les commandes de manipulation des groupes sont les suivantes :

1. **groupadd** : création d'un groupe
2. **groupdel** : suppression d'un groupe
3. **groupmod** : modification d'un groupe
4. **groups** :affichage des groupes d'appartenance d'un utilisateur
5. **id** : affichage des identifiants d'utilisateur et de groupes d'un utilisateur.

Beaucoup d'autres commandes sont disponibles sur le système :

- la commande **whoami** affiche l'ientifiant de l'utilisateur, la commande **who** montre qui sont les utilisateurs connectés au système.
- La commande **newgrp** permet à un utilisateur appartenant à plusieurs groupes de se connecter avec un autre groupe.

- La commande **su** permet de changer de profil ou d'exécuter une commande avec un UID ou GID de substitution.
- La commande **sg** permet d'exécuter une commande avec un autre identifiant de groupe (GID).
- Sur **Debian** et certaines distributions dérivées, les commandes **adduser** et **addgroup** sont des variantes plus conviviales respectives des commandes **useradd** et **groupadd**.
- Sur **Debian** et les distributions dérivées, la commande **sudo** permet d'exécuter une commande avec des privilèges d'administrateur.
- Les **références croisées** vous renverront vers d'autres commandes intéressantes comme **chage**, **chpasswd**,..., Reportez vous aux **manpages** de ces commandes, elles offrent des options intéressantes.

4. Les fichiers relatifs aux comptes

Fichiers de profil

A la **connexion sur un terminal**, plusieurs fichiers sont lus par le système pour définir l'**environnement de travail** de l'utilisateur (shell utilisé, prompt, couleur du terminal, couleur du texte,...), ils sont appelés fichiers de profil.

L'**administrateur** du système peut définir des éléments de configuration de profil qui s'applique à tous les utilisateurs dans le fichier **/etc/profile**. C'est le premier fichier lu à la connection.

Les **utilisateurs** peuvent définir des paramètres de configurations relatifs à **leur environnement** à partir d'un ensemble de **fichiers cachés** situés dans leur **répertoire personnel** : **.bash_profile** et **.profile** pour un shell de connection (avec saisie du login et du mot de passe) , et **.bashrc** pour un non-login shell qui ne demande pas de login et de mot de passe.

D'autres fichiers sont utilisés dans ce contexte, reportez vous aux manpages ou à la documentation GNU sur ce point [1].

Les fichiers de gestion des comptes

Les **informations de gestion des comptes** sont dans plusieurs fichiers du système : /etc/passwd, /etc/shadow, /etc/login.defs, /etc/group,...

La **modification** de ces fichiers par un administrateur **impacte** sur les comptes (ajout/suppression de compte, de groupe...), il est toutefois **conseillé** d'**utiliser les commandes systèmes** dédiées à ces tâches (useradd,userdel,...) pour la stabilité du système.

Le fichier /etc/passwd

Les **informations** sur les **comptes utilisateurs** disponibles sur une machine Unix sont regroupées dans le fichier **/etc/passwd**. Ce fichier appartient à root:root mais peut être lu par tous les utilisateurs. Chaque ligne de ce fichier correspond à un compte. Une ligne est composée de 7 champs séparés par des :. Les champs sont les suivants :

1. le **nom du compte** de l'utilisateur (login) ;
2. le **mot de passe crypté** de l'utilisateur ;
3. l'identifiant de l'utilisateur appelé **User IDentifier (UID) ;**
4. l'identifiant du groupe primaire de l'utilisateur appelé **Group identifier (GID) ;**
5. les **informations** sur l'utilisateur (nom, ...)
6. le **répertoire de connexion** qui est celui dans lequel il se trouve après s'être connecté au système
7. l'**interpréteur de commande (shell)** par défaut de l'utilisateur

1 - https://www.gnu.org/software/bash/manual/html_node/Bash-Startup-Files.html

La **page de manuel** de ce fichier est en **section 5** manpages. Vous avez ici des exemples de lignes du fichier /etc/passwd :

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/bin/bash

daemon:x:2:2:daemon:/sbin:/bin/bash

Si le champs de mot de passe contient un **"x" minuscule**, le mot de passe crypté est encodé dans le fichier **/etc/shadow**. si le champs est **vide**, c'est un **compte sans mot de passe**. passwd possède une copie de sauvegarde passwd~ qui est utilisée par certains outils.

Le fichier /etc/shadow

/etc/shadow est un fichier qui contient les **informations cachées** concernant les mots de passe des utilisateurs et leurs dates de validité. Ce fichier ne doit pas être accessible en lecture par les utilisateurs normaux afin de maintenir la sécurité des mots de passe, en particuliers pour prévenir les attaques par dictionnaires. Il appartient par défaut à root:shadow. Chaque ligne de ce fichier contient **9 champs**, séparés par des deux-points (« : »), dans l'ordre suivant : login, mot de passe chiffré, date du dernier changement de mot de passe, âge minimum du mot de passe, âge maximum du mot de passe, période d'avertissement d'expiration du mot de passe, période d'inactivité du mot de passe, date de fin de validité du compte, champ réservé pour le futur. (Ce texte sur shadow est tiré des manpages, vous pouvez vous y reporter pour compléter (section 5)).

Le fichier **/etc/login. defs** définit la configuration de la suite **shadow password** (mots de passe cachés) pour le système. Ce fichier est **indispensable**. Ses directives contrôlent la longueur du mot de passe, le délai dans lequel l'utilisateur doit changer le mot de passe du compte, la création par défaut d'un répertoire personnel, etc. Le fichier est généralement rempli de commentaires et de **directives commentées** (ce qui rend les directives inactives).

le fichier /etc/group

Les informations sur les **groupes d'utilisateurs** disponibles sur une machine Unix sont regroupées dans le fichier **/etc/group**. Chaque ligne de ce fichier correspond à un groupe. Une ligne est composée de 4 champs séparés par des :. Les champs sont les suivants :

1. le **nom du groupe** ;
2. le **Mot de passe** du groupe qui est rarement utilisé ;
3. l'**identificateur du groupe** : le GID ;
4. la **liste des utilisateurs membres** (login) du groupe. Les utilisateurs dont c'est le groupe principal n'ont pas besoin d'apparaître dans cette liste (leur groupe principal est indiqué dans /etc/passwd).

Vous avez ici des exemples de lignes du fichier /etc/group :

root:x:0:root

bin:x:1:root,bin,daemon

daemon:x:2:

Exercice : Que contient le fichier /etc/gshadow.

II.Mécanismes de droits

La **sécurité** et la **confidentialité** des **ressources** (processus et fichiers) sur GNU/Linux est géré par un système de **mécanismes de droits**. La sécurité des processus découle de celle des fichiers, un processus est un programme en cours d'exécution qui hérite des propriétés de sécurité du fichier (programme de lancement) avec quelques règles spéciales. Nous nous limiterons donc dans cette section au cas des **fichiers et répertoires** dont nous pourrons déduire le cas des processus.

Tout **fichier** ou **répertoire** sur un système GNU/Linux a un **utilisateur propriétaire** qui définit les droits qu'il accorde sur cette ressource. Le **partage de ressources** entre comptes d'utilisateurs qui collaborent est facilité par le concept de groupe, tout **fichier** ou **répertoire** appartient aussi a un **groupe propriétaire** dont les droits sur la ressource sont définis par l'utilisateur propriétaire.

1. Propriétaires de fichier

Identité d'utilisateur

Les **droits** sur un système GNU/Linux sont propres aux fichiers et répertoires et sont définis par le propriétaire de la ressource selon l'identité de l'utilisateur qui accède à la ressource. **Trois identités** sont considérées pour **chaque fichier** et **répertoire** dans les systèmes UNIX :

1. le **propriétaire** du fichier/répertoire identifié par son nom, il sera désigné par la lettre u (user);
2. le **groupe propriétaire** identifié par le nom de groupe, il sera désigné par la lettre g (group)
3. les **autres utilisateurs** du système qui ne sont ni le propriétaire ni membre du groupe propriétaire du fichier/répertoire, ils seront désignés par la lettre o (other - les autres).

L'ensemble des trois entités sera désigné par la lettre a (pour all).

Détermination du propriétaire du fichier

A la **création d'un fichier** ou d'un répertoire (par une commande, par téléchargement,...), l'**utilisateur créateur** du fichier/répertoire en est l'**utilisateur propriétaire** et son **groupe actif** en est le **groupe propriétaire**.

Le système autorise de **modifier les propriétaires** (utilisateur et groupe) d'un fichier, c'est un droit détenu seulement par l'**administrateur** sur certaines distributions.

Les commandes

Les **commandes** suivantes permettent de **modifier les propriétaires** d'un fichier/répertoire :

1. **chown** : Modifier le propriétaire et le groupe d'un fichier
2. **chgrp** : Changer le groupe propriétaire d'un fichier

2. Les droits standards

Droits standards sur les fichiers et répertoires

Trois types de **droits standards** s'appliquent aux **fichiers** et **répertoires** et pour **chacune des identités** (utilisateur, groupe, autres) :

1. le droit de **lecture**, désigné par la lettre **r** (**read**) ;
2. le droit d'**écriture**, désigné par la lettre **w** (**write**)
3. et le droit d'**exécution**, désigné par la lettre **x** (**eXecute**)

Les **droits sur un fichier**/**répertoire** sont donc représentés par **trois séries de trois lettres** (**9 caractères**) :

1. La **première série** définit les droits de l'**utilisateur propriétaire** (**u**) ;
2. la **deuxième série** ceux du **groupe propriétaire** (**g**)
3. et la **troisième série** ceux des **autres utilisateurs**(**o**).

Pour **chacune des trois séries**, les droits sont indiquées comme suit :

1. Le **premier caractère** est utilisé pour le **droit de lecture** sur le fichier/répertoire. La **détention du droit** est indiquée par la lettre **r** (pour read) et l'**absence du droit** est indiquée par un **"-"** (**tiret du 6**) ;
2. Le **deuxième caractère** est utilisé pour le **droit d'écriture** indiqué par **w** (write) ou par **"-"** (absence du droit) ;
3. le **troisième caractère** est utilisé pour le droit d'exécution indiqué par **x** (eXecute) ou par **"-"** (absence du droit).

all (a)								
utilisateur propriétaire (u)			groupe propriétaire (g)			autres (o)		
r	w	x	r	w	x	r	w	x

Sémantique des droits standards

Les droits ont une sémantique différente selon qu'elle portent sur un fichier ou un répertoire :

1. cas d'un **fichier** :
   a. lecture (r): permet de lire (commande cat par exemple), visualiser (cas d'une image) ;
   b. écriture (w) : permet de modifier le fichier (usage des redirections, enregistrement sur un éditeur,...) ;
   c. exécution : permet d'exécuter le fichier (scripts shell, programmes,...) ;
2. cas d'un **répertoire**
   a. lecture (r): permet de lire le contenu du répertoire (ls par exemple) ;
   b. écriture (w) : permet de modifier le contenu du répertoire (mkdir, touch, cp, rm,...)
   c. exécution : permet de traverser le répertoire (cd par exemple)

La notation numérique

Les **droits** sont positionnés en réalité sur **1 bit**, si le bit est à 1, le droit est accordé et si le bit est à zéro, le droit est absent. Le **triplet de droit** pour une entité peut ainsi être **représenté** par une **notation binaire sur trois bit** dont on déduit une **notation octale**. Cela est illustré dans le tableau suivant

Triplet	Droits	Nombre binaire	Nombre octal
- - -	aucun	000	0
- - x	exécution	001	1
- w -	écriture	010	2
- w x	écriture et exécution	011	3
r - -	lecture	100	4
r - x	lecture et exécution	101	5
r w -	lecture et écriture	110	6
r w x	lecture, écriture et exécution	111	7

Cette notation permet de représenter les droits standards sur un fichier par un nombre à 3 chiffres. Chaque chiffre ayant une valeur octale (entre 0 et 7) : le premier chiffre à gauche représentera les droits de l'utilisateur propriétaire, le deuxième chiffre celui du groupe propriétaire et le dernier chiffre celui des autres.

all (a)								
utilisateur propriétaire (u)			groupe propriétaire (g)			autres (o)		
r	w	x	r	w	x	r	w	x
0 - 7			0-7			0-7		

3. Les droits spéciaux

Le Sticky bit

Le **Sticky bit** est un droit spécial qui s'applique aux **fichiers exécutables** et **répertoires** mais avec un effet différent :

1. pour les **fichiers exécutables** - le programme **restera en mémoire** pour une exécution ultérieure
2. pour les **répertoires** - il autorise que **seul le propriétaire** d'un fichier puisse **modifier/supprimer** ce dernier, l'exemple le plus fréquent est le répertoire /tmp dans lequel tout utilisateur peut créer un fichier qu'un autre utilisateur ne devrait pas pouvoir modifier/supprimer.

En **mode symbolique**, il est **positionné** à la place du droit **x des autres** (other), il vaut **t si le droit x est positionné** et **T sinon**. En mode **numériquement**, il correspond à **1000**.

Les droits d'endossement SetUID (SUID) et le SetGID (SGID)

Les **droits d'endossement** dans GNU/Linux permettent à un utilisateur de lancer une commande (exécutable) avec l'**identité de l'utilisateur propriétaire** (SUID) ou du **groupe propriétaire** (SGID).

En mode symbolique, ils sont **positionnés** à la place **du droit x** de l'**utilisateur propriétaire (SUID)** et du **groupe propriétaire (SGID).** Ils sont représentés par la lettre **s**, si le droit x est positionné, et par la lettre **S** sinon. En **mode numérique**, le **SUID** correspond à **4000** et le **SGID** à **2000**.

La commande de modification de mot de passe **passwd** et l'outil de planification de tâches **crontab** sont de bons exemples :

- La commande **passwd** appartient à root:root mais avec des droits d'exécution pour les other. Cela permettrait à tout utilisateur de pouvoir changer son mot de passe mais vu que la commande accède à des fichiers appartenant au root (/etc/passwd,/etc/shadow), il devra être lancé avec les droits du root (**SUID**) et non de l'utilisateur qui veut modifier son mot de passe. On dira que l'utilisateur qui lance la commande endosse l'identifiant du root (utilisateur propriétaire).
- La commande **crontab** appartient à root:crontab et crée un fichier dans /var/spool/cron/crontabs pour l'utilisateur qui a exécuté la commande crontab. L'écriture dans le répertoire /var/spool/cron/crontabs est accordée à root:crontab mais pas aux autres, l'utilisateur qui exécute la commande devra endosser l'identité du groupe crontab (SGID).

Exercice : afficher les droits sur les fichiers passwd, crontab et sur les répertoires/etc/passwd, /etc/shadow et /var/spool/cron/crontabs

Remarque : mode numérique

Vous noterez que l'**ensemble des droits (droits standards et spéciaux)** sont représentés en mode numérique par des chiffres octaux (entre 0 et 7) sur **4 positions** : le premier chiffre à gauche pour les **droits spéciaux** (sticky bit =1000 , suid= 2000,sgid = 4000) et les **droits standards** occupent les **trois positions restantes** : le deuxième chiffre pour les droits de l'utilisateur propriétaire, le troisième pour les droits du groupe propriétaire et le dernier pour les autres.

4. Les commandes

Visualisation des droits

La commande **ls -l** permet de **visualiser** le propriétaire, le groupe propriétaire et les droits sur les **fichiers** et **répertoires**

affichage de ls -l

Les modes de gestion des droits

La **gestion des droits d'accès** peut se faire de **deux manières** complémentaires : **a priori** ou **a posteriori** :

1. L'utilisateur peut **fixer les (futur) droits** des fichiers et répertoires qui seront créés à l'aide du **umask** ;
2. Une fois les fichiers et répertoires créés, le système lui laisse également la possibilité de **modifier les droits** sur ces fichiers et répertoires de manière ponctuelle avec la commande **chmod**.

La gestion des droits a posteriori : chmod

la commande chmod permet de modifier les droits sur un fichier ou un répertoire existant. il a **deux modes d'usage** : le **mode symbolique** et le **mode numérique** (**octal**).

1. En **mode symbolique**, la syntaxe est :
   chmod [OPTION] [u g o a] [+ - =] [r w x X s t] FICHIER ...
   u, g, o et a peuvent être utilisés seul ou en combinaison pour respectivement l'utilisateur propriétaire (u), le groupe propiétaire (g), les autres (o) ou tous (a équivaut à ugo). +,- et = indiquent l'ajout, le retrait ou l'affectation de droit et les symboles r w x s t les droits de lecture (r),d'écriture(w), d'exécution (x), les suid/sgid (s) et le sticky bit (t).X est utilisé pour
2. En **mode numérique**,la syntaxe est :
   chmod [OPTION] mode-octal FICHIER ...
   mode-octal sera une valeur numérique composée de 4 chiffres en octal.

chmod possède une option intéressante **- -reference** qui permet de modifier le mode de chaque fichier en celui d'un **fichier de réference**.

La gestion des droits apriori avec le masque de protection des fichiers : umask

La commande **umask** ne change pas les droits sur les fichiers existants mais les fixe pour les (futur) **fichiers/répertoire à créer** (par une commande, par téléchargement, à travers un logiciel, ...).

Elle utilise la **notation numérique** et fonctionne comme un **filtre** qui déduit des droits maxima les droits indiqués par le masque.

Pour fixer les droits globaux d'un fichier ou d'un répertoire le système procède ainsi en deux étapes

1. il alloue au fichier les droits maxima de 666 (rw-rw-rw-)et au répertoire les droits maxima de 777 (rwxrwxrwx) ;
2. et ensuite déduit le masque.

Par exemple pour un masque fixé à 022 (----w--w-), c'est les droits d'écriture qui seront déduits pour le groupe et les autres (o) :

- Les **nouveaux fichiers** qui vont être créés recevrons les **droits maxima 666** (rw-rw-rw-) auxquels on **déduit le masque 022**, les droits des fichiers seront alors de **644** (rw-r--r--).
- Les nouveaux répertoire recevrons les **droits maxima 777** (rwxrwxrwx) auxquels on **déduit le masque 022**, les droits des répertoires seront alors de **755** (rwxr-xr-x).

Remarque : Les **droits maximas** sont de 666 pour un **fichier** par mesure de sécurité, les fichiers exécutables par défaut sont une des failles de certains systèmes. un fichier n'est pas par défaut exécutable sur les systèmes GNU/Linux, il revient alors à l'utilisateur de rendre exécutable ses scripts et fichiers exécutables (par exemple les fichiers C compilés) manuellement par la commande chmod.

La **commande umask** sans paramètre affiche la valeur du masque actuel de l'utilisateur. Il suffit de lui passé en paramètre la valeur du masque souhaité pour que ce dernier soit fixé à cette valeur de masque. Exemple

$ umask 002

fixe le masque à 002

Attention

Le masque est définit pour tous les utilisateurs par l'administrateur du système dans le fichier /etc/profile qui est exécute à la connexion de l'utilisateur.

Pour que le changement du masque par l'utilisateur soit permanent, il est recommandé de mettre la commande de changement dans l'un des fichiers de profil de l'utilisateur lu par le système à la connexion tel que .bashrc.

Conclusion

La gestion des comptes et le système de mécanismes de droits que nous avons introduit dans ce chapitre est une architecture robuste pour assurer les fondements de la sécurité du système. Cela permet d'avoir une séparation des espaces des utilisateurs mais également d'assurer que les processus qui s’exécutent ne puisse accéder qu'aux ressources auxquels ils ont droits.

La sécurité d'un système est toutefois un processus qui s'appuie sur l'architecture du système d'une part mais aussi sur la responsabilité des utilisateurs d'autre part.

L'administrateur du système doit définir une politique de sécurité forte :

- mettre en place un système d'authentification robuste (longueur de mot de passe/durée de validité) ;
- mise en place d'une structuration d'utilisateurs et de groupes claire ;
- une politique d'accès aux ressources et de partage claire ;
- ...