Droits Linux serveur Debian

# Rappels

Lors de la création d’un fichier ou répertoire, l’utilisateur qui le crée devient propriétaire ainsi que son groupe (Les droits par défaut sont gérés par UMASK), il conviendra de modifier ceux-ci si besoin.

La commande ***ls -l***(list -long) permet de voir le contenu détaillé d’un répertoire et fichier*.*

Ex :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Droits Utilisateur** | **Droits Groupe** | **Droits Autres (Others)** | **Nbre de liens**  **Ou**  **Nbre de répertoires contenus** | **Nom utilisateur**  **(Propriétaire)** | **Nom du groupe** | **Taille en Octets** | **Date et heure de modification** | **Nom du fichier ou du répertoire** |
| d | rwx | r-x | r-x | 3 | tech | Root | 4096 | 21juil. 16 :36 | Test4 |

Utilisateurs

* **Utilisateur :** Utilisateur connecté au système.**/etc/passwd**
* **Groupe :** Groupe appartenant au système. **/etc/group**
* **Utilisateur Propriétaire** (noté **u** comme user) : Utilisateur qui est en possession du fichier
* **Groupe Propriétaire** (noté **g** comme group): Groupe d'utilisateurs qui est en possession du fichier
* **Autres Utilisateurs** (noté **o** comme other): Utilisateurs qui ne sont ni propriétaire du fichier, ni faisant partie du groupe propriétaire.
* **Tous** (noté **a** comme all): Utilisateur propriétaire + Groupe propriétaire + Autres utilisateurs.

# Type (7 types)

### Base

**-** : File/Fichier

**d**: Directory /Répertoire/Dossier

**l**: Link/Lien symbolique (=shortcut, permet d’atteindre un fichier sans passer par le chemin entier)

### Avancé

**c**: fichier spécial de caractère - Le fichier spécial en mode caractère : il permet de transférer les données vers le périphérique sous formes des flux, ex : /dev/tty\* (console), clavier, souris, port série…

\*Le contenu des fichiers en mode bloc sont généralement gardé en mémoire, ce qui n’est pas le cas des fichiers en mode caractère.

**b** : Bloc de fichier spécial - Le fichier spécial en mode bloc : il permet de transférer les données vers le périphérique en utilisant les buffers du système, ex : /dev/sd\* (gère les données par blocs)

\*Les périphériques que l’on peut monter, comme par exemple les périphériques de stockage, peuvent être accédés en mode bloc et en mode caractère. ***Les périphériques que l’on ne peut pas monter, ne sont accessibles qu’en mode caractère****.*

**p** : pipe/canal de communication FIFO (First In First Out) ex : spool d’impression, buffers…

**s**: socket / les sockets (en français « interfaces de connexion » mais aussi « ports, points de connexion, connecteurs logiciels ») par lesquelles une application peut se brancher à un réseau et communiquer ainsi avec une autre application branchée depuis un autre ordinateur. (Wikipédia)

UMASK

ls -l : affiche la liste longue des détails ; permet de connaitre les accès au dossier

# Droits

## Base

|  |  |  |
| --- | --- | --- |
| Qui | Peut faire quoi | Sur quoi |
| U G O | * R W X | Dossier / fichier |

Qui (U G O)

U : propriétaire (User)

G : groupe (Group) = groupe propriétaire

O : Autres (Others)

A : Tous (all)

Peut faire quoi sur quoi

### Fichier

**-** : pas de droit là où il se trouve (interdit)

R : Lecture (Read) : Permet de lire, ouvrir, visualiser (cas d'une image) -ex :cat

W : écriture (Write) : on peut modifier le fichier et le vider de son contenu.

X : eXécutable (eXecute) : possibilité d'exécuter le fichier (permet l'exécution d'un programme, d'un exécutable, et permet d'accéder aux informations de gestions des fichiers du répertoire

### Dossier

**-** : pas de droit là où il se trouve (interdit)

R : lecture (Read) : il autorise l'affichage du contenu du répertoire (la liste des fichiers présents à la racine ***de ce répertoire***) -ex : ls

W : écriture (Write) : il autorise la création***, la suppression et le changement de nom des fichiers qu'il contient***, quels que soient les droits d'accès des fichiers de ce répertoire (même s'ils ne possèdent pas eux-mêmes le droit en écriture) ex : mkdir, touch, cp, rm

Néanmoins le droit spécial *sticky bit* permet de passer outre ce comportement.

X : il autorise l'accès (ouvir) au répertoire ex : cd

Chmod : modifier les droits

Octal 8 bits

### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Binaire** | **Octal** | **User** | **Group** | **Other** | **Spécial** |
| **000** | **0** | **- - -** | **- - -** | **- - -** | **- - -** |
| **001** | **1** | **- - x** | **- - x** | **- - x** | **- - t** |
| **010** | **2** | **- w -** | **- w -** | **- w -** | **- s -** |
| **011** | **3** | **- w x** | **- w x** | **- w x** | **- s t** |
| **100** | **4** | **r - -** | **r - -** | **r - -** | **s - -** |
| **101** | **5** | **r – x** | **r – x** | **r – x** | **s – t** |
| **110** | **6** | **r w -** | **r w -** | **r w -** | **s s -** |
| **111** | **7** | **r w x** | **r w x** | **r w x** | **s s t** |

Prendre la valeur octal pour chaque autorisation et l’appliquer pour U, pour G et pour O

UGO

Ex : chmod 7 6 0 *Nom\_Du\_Fichier/Dossier*  = *User* **rwx** *Group* **rw-** *Other* **- - -**

### Octal spécifique

1000 : Stickybit (t)

2000 : Sgid (Set Group ID) (s/S)

4000 : Suid (Set User ID) (s/S)

Ils peuvent être additionnés (Voir colonne Spécial du tableau ci-dessus)

### Umask (user file-creation mode)

Umask est un fichier qui sera lu lors de la création d’un fichier ou d’un dossier, peu importe qui le crée.

Umask est un fichier modifiable et personnalisable

---

**Par défaut lors de la création d’un fichier ou d’un dossier linux donne des droits**

Pour un **fichier** 666 = rw- rw- rw-

Pour un **dossier** 777 = rwx rwx rwx

Umask est utilisé lors de la création d’un fichier ou d’un dossier pour rendre plus restrictif les droits donner par défaut par Linux.

Le principe de fonctionnement est le suivant, pour protéger au mieux les fichiers et dossiers Umask va supprimer les droits d’écriture pour les groupes et les autres (other).

La valeur d’écriture en octal est de 2 (voir le tableau), ce qui donne pour le groupe la valeur de 2 et pour other 2

Qui donne pour user 0 (on ne change rien) 2 pour le groupe et 2 pour other = 022, cette valeur (022) sera soustraite de la valeur fournie par linux lors de la création d’un fichier ou d’un dossier.

666 – 022 = 644 — pour les fichiers

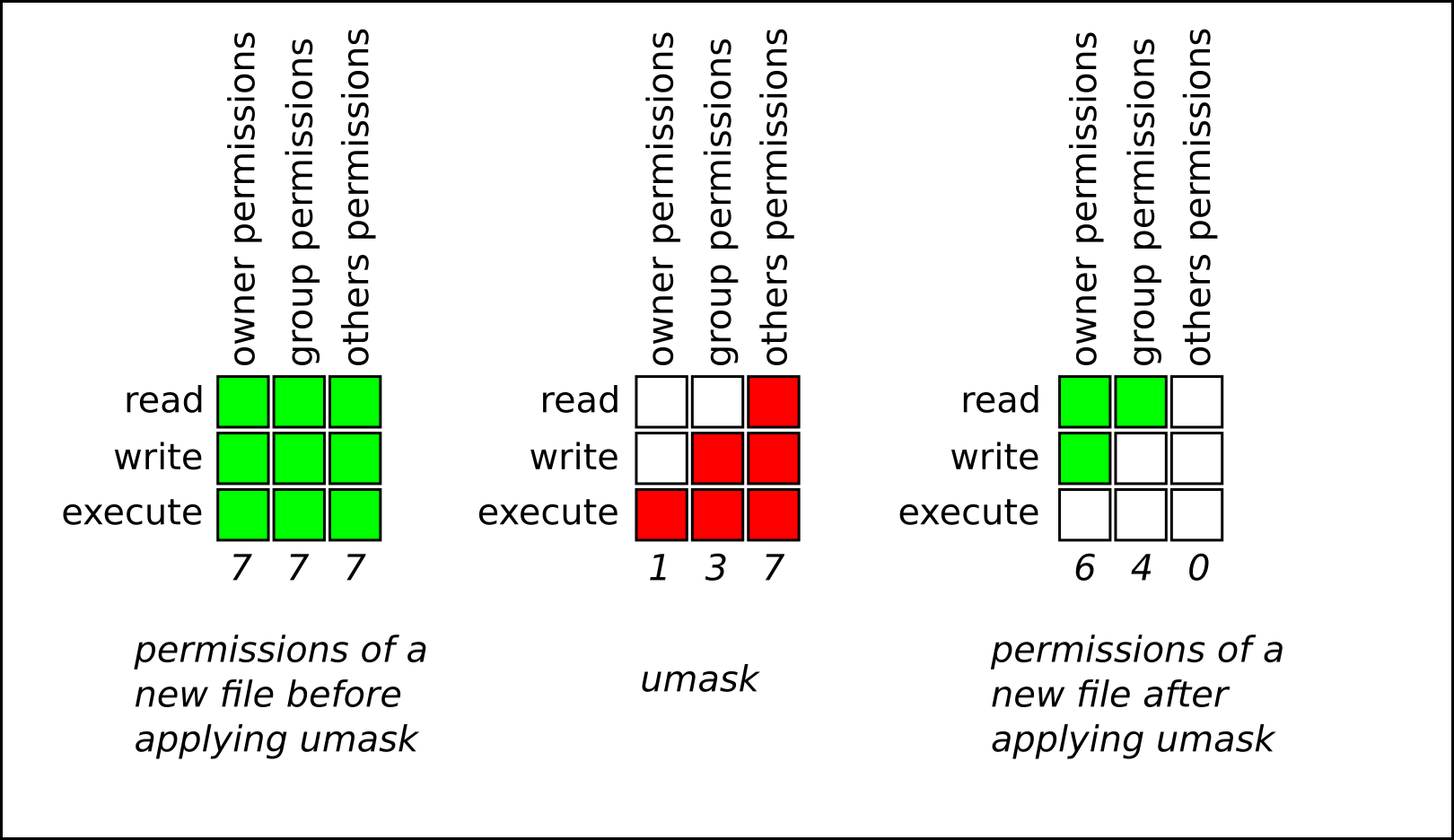
777 – 022 = 755 — pour les répertoires

Ce qui a pour résultat pour un fichier

644= rw- r—r—

Et pour un dossier

755= rwx r-x r-x



## Avancé

Résumé : Le S**U**ID est une élévation de droits pour l’utilisateur (USER), le S**G**ID est une élévation de droits pour les groupes (GROUP), le STIKY BIT est une limitation pour les autres (OTHER).

Le SU/GID est considéré comme une faille de sécurité, il est préférable d’utiliser SUDO (bien sûr configurer comme il se doit, donc pas ALL (ALL:ALL) ALL ☹)

### StikyBit (uniquement sur les dossiers pour autoriser le propriétaire de les renommer, supprimer, modifier…) -t

Un sticky bit, est **un bit d'autorisation sur un fichier ou un répertoire** qui permet ***uniquement au propriétaire*** du fichier/répertoire ou à l'utilisateur ***root*** de ***supprimer ou de renommer*** le fichier



Le stiky bit sur un fichier n’est plus d’actualité, il servait pour garder un programme ouvert en mémoire pour une ré utilisation plus rapide.

### SUID : élevation de privilège user

rw s rw-rx - Le bit SUID est défini et le binaire est exécutable

rw S rw-rx - bit SUID défini et le binaire n'est pas exécutable

### SGID : élevation de privilège groupe

rwxrw s r-x - bit SGID défini et le binaire est exécutable

rwxrw S r-x - Bit SGID défini et le binaire n'est pas exécutable

### 