



Les commandes de base UNIX

TP 5 : Droits, Processus

Aller sur moodle

[Travaux Pratiques numéro 6](#)

NOTION DE PROCESSUS sous UNIX

- [Afficher l'état des processus actifs : ps](#)
- [Arrêter un processus actif : kill](#)
- [Différentes façons de lancer un processus](#)
 - [Continuité d'exécution d'un processus : nohup](#)
 - [Exécution d'un processus en différé : at time, batch](#)
 - [Exécution d'un processus avec priorité basse : nice](#)

Exercices de contrôle

Les processus

1. Affichez la liste des processus associés à votre terminal. Affichez la liste des processus dont vous êtes propriétaire. Recommencez en utilisant les options `-l` et `-f`. À quoi correspondent les colonnes `PID` et `PPID` ?
2. Lancez une commande longue en arrière plan. Quel est le comportement du processus associé lorsqu'il reçoit les signaux suivants :
 - `sigkill` (9)
 - `sigstop` (23)
 - `sigcont` (25)
3. Utilisez la commande `nice` pour lancer des commandes ayant une faible priorité.
4. Interprétez la hiérarchie des processus qui vous appartiennent.
5. La commande `ps | wc` compte deux processus en plus de ceux qui existent réellement lorsqu'on lance la commande. Pourquoi ?
6. Donner deux commandes pour reprendre l'exécution d'une instruction interrompue par un `^Z`. conseil voir les commandes `fg` et `bg` avec `man`

NOTION D'UTILISATEUR

Fichiers systèmes : `/etc/passwd`, `/etc/group`

- [Permissions et contrôle d'accès](#)
 - [Modification des droits d'accès : chmod](#)
 - [Changement de propriétaire : chown](#)
 - [Changement de groupe : chgrp](#)
 - [Notion de masque : umask](#)
 - [Notion de Super Utilisateur](#)

Rappel : créer un fichier avec `touch`

1. Quel jour sommes-nous ?
2. Quel jour de la semaine êtes-vous né ?
3. Votre nom de login

Les droits

- Changez les droits d'un fichier `fic1` pour que tous ceux de votre groupe puissent écrire dedans.
 - Donnez en une seule ligne le droit d'exécution à tous les utilisateurs d'un fichier `script` qui n'a jusqu'alors que des droits standards (`-rw-r--r--`).
 - Le fichier `toto` a les droits suivants : `-rwxr--r--`. Modifiez-en les droits en une ligne de commande de sorte que le propriétaire n'ait plus que le droit de lecture.
 - Modifier les droits du fichier `toto` (`-rwxr--r--`) de sorte que le groupe et les autres utilisateurs aient les mêmes droits que le propriétaire.
 - Quelle option permet de modifier récursivement les droits d'un répertoire et des fichiers qu'il contient ?
 - Quelle option de `mkdir` permet de créer un répertoire en spécifiant les droits sur ce répertoire ?
 - Affichez et interprétez les droits de `/usr/sbin/mount`.
 - si vous tapez : **1) umask 022** **2) umask 244**
- Quels sont les droits pour un fichier, pour un répertoire pourquoi ?

Aller à [Travaux Pratiques numéro 7](#)



Cours

Exercices

 Exercices de Synthèses : faire cet exercice

EXERCICE DE SYNTHÈSE

- 1) Créer un fichier vide de nom `essai.txt`
 - ◆ Sans connaître les droits positionnés, mettez le fichier en lecture seule pour tous de deux manières (symbolique et numérique)
 - ◆ Que donne la commande `rm essai.txt` ?
 - ◆ Quels sont les droits du fichier `essai.txt` après l'exécution de la commande suivante :
`$ chmod u+wx,g+w,o-r essai.txt` ?
- 2) Créer un répertoire de nom `perso`, et protégez le contre tout accès des autres utilisateurs
 - ◆ Rendez ce répertoire accessible en lecture aux utilisateurs membres de votre groupe
 - ◆ Pouvez vous renommer votre répertoire de connexion, pourquoi ?
 - ◆ Quelle valeur de « `umask` » utiliser pour que lors d'une création les droits soient :
`drwxr-x---` pour un répertoire

Complément : pour ceux qui ne sont pas habitué encore au binaire et octal

chmod utilise le codage binaire, fondé sur l'association de valeurs numériques aux différentes permissions :

droits	valeur octale	valeur binaire
---	0	000
--x	1	001
-w-	2	010
-wx	3	011
r--	4	100
r-x	5	101
rw-	6	110
rwX	7	111

La commande **chmod** permettant de positionner **rwXr-x---** sur **fic1** à la syntaxe suivante : **chmod 750 fic1**

Il existe aussi un autre moyen de positionner les droits d'un fichier avec cette même commande. Vous pouvez utiliser des +, - ou = pour ajouter, supprimer ou fixer des droits à l'une ou l'autre des catégories u (user), g (group), o (other) ou a tous (a) en précisant le type de droit d'accès, r (read), w (write) ou x (execute).

Cas d'un répertoire

*Pour un répertoire le **x** n'est pas un droit en exécution, mais un droit d'accès au répertoire, sans ce droit, on ne peut pas accéder au répertoire et voir ce qu'il y a dedans.*

En tapant **ls -l** sur un répertoire, vous obtenez :

```
drwxr-x--- 1 truc users 13242 Dec 2 13 :14 mon-répertoire
```

d signifie qu'on a affaire à un répertoire, rwx sont les droits du propriétaire olivier qui est autorisé en lecture, écriture et droit d'accès au répertoire-r-x droits du groupe users, autorisé en lecture, droit d'accès au répertoire, pas de droit en écriture --- droits des autres utilisateurs, aucun droit dans le cas présent

Changer les droits par défaut : umask

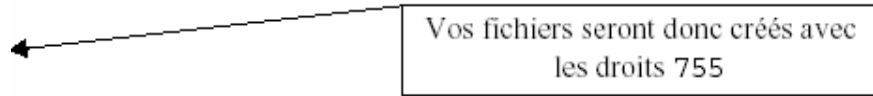
Quand vous créer un fichier, par exemple avec la commande **touch**, ce fichier par défaut possède certains droits, par exemple **666** pour un fichier (**-rw-rw-rw-**) et **777** pour un répertoire (**-rwxrwxrwx**). **umask** permet de changer les paramètres par défaut.

umask xyz : permet de définir le masque pour tous les fichiers nouvellement créés.

La valeur octale xyz associée à umask est soustraite de la valeur existante.

Chaque x, y, z représente une valeur entre 0 et 7 qui est le **complément à 7** des droits à affecter aux fichiers. Si l'on veut avoir des fichiers avec 751 (rwxr-x--x) comme droits, il faudra définir comme masque 026 (on remarque que $7+0 = 5+2 = 1+6 = 7$).

Exemple :



Vos fichiers seront donc créés avec
les droits 755

\$ **umask** 022

Après cette commande tous les fichiers nouvellement créés auront les droits suivants : -rw-r--r-- et les nouveaux répertoires les droits : -rwxr-xr-x, donc créés avec les droits 755

umask : permet de connaître le masque actif.