

Linux en ligne de commande

I. Organisation et gestion des fichiers.

A – organisation des répertoires :

Les systèmes Unix reposent sur une arborescence de fichiers différente des systèmes Windows. En effet, vous ne trouverez pas sur Unix de périphérique en tant que tel ; la philosophie est la suivante : *tout est fichier*. Les fichiers peuvent donc représenter des données, des programmes, des périphériques, des processus ... Ils sont organisés suivant une arborescence qui peut varier selon les systèmes. En règle générale nous retrouvons la structure basique suivante :

/	=> racine des fichiers (ROOT)
/etc	=> répertoire contenant les fichiers de configuration.
/usr	=> répertoire des applications
/var	=> répertoire des données et logs
/proc	=> répertoire des processus
/dev	=> répertoire des périphériques
/home	=> répertoires privés des utilisateurs.
/tmp	=> répertoire des fichiers temporaires.

B – types des fichiers :

Les fichiers n'ont pas d'extension particulière et la notion même d'extension, bien que couramment utilisée, n'existe pas. Le type d'un fichier est identifié par son contenu. La commande *file* permet d'en connaître le type :

```
>file plugin131_04.trace  
plugin131_04.trace: ISO-8859 text  
>file x_keysrv.c  
x_keysrv.c: ISO-8859 C program text  
>file projets/mcs/HEXBIN.EXE  
projets/mcs/HEXBIN.EXE: MS-DOS executable (EXE)  
>file /dev/ttya0  
/dev/ttya0: character special (3/176)
```

C – Manipulation :

Les fichiers sont organisés en répertoires et sous-répertoires comme sur un système Windows. La navigation peut se faire au travers de commandes simples :

<i>pwd</i>	Indique le répertoire courant.
<i>mkdir rep</i>	Crée dans le répertoire courant un nouveau répertoire appelé <i>rep</i> .
<i>ls</i>	Liste les fichiers du répertoire courant. Des options peuvent être ajoutées : <i>ls -a</i> liste tous les fichiers, y compris ceux cachés (débutant par '.') <i>ls -l</i> liste les fichiers en ajoutant les informations telles que les droits et le propriétaire.
<i>cd rep</i>	Change le répertoire courant pour le répertoire <i>rep</i> . Par exemple <i>cd /home</i> vous amène directement dans le répertoire home. Il existe certains répertoires particuliers : '.' indique le répertoire courant. '..' indique le répertoire de niveau supérieur dans l'arborescence.

rm fichier	Supprime le fichier 'fichier'. L'option -f évite d'avoir à confirmer chaque effacement. L'option -r permet de supprimer un sous-répertoire et son contenu.
rmdir rep	Supprime le répertoire 'rep' s'il est vide.
cp src dst	Copie le fichier dont le chemin est <i>src</i> vers le chemin <i>dst</i> . L'option -R indique une copie récursive (incluant les sous-répertoires).
mv src dst	Déplace le fichier dont le chemin est <i>src</i> vers le chemin <i>dst</i> . L'opération réalisée est un couper-coller.

D – Utilisation des jokers :

Il est possible de remplacer un nom de fichier par un nom incluant des jokers. Un joker est un caractère auquel peut correspondre tous caractères voire toutes suites de caractères. Le joker "*" indique une suite de caractères quelconque. Le joker '?' indique un caractère unique. Ceci peut être utilisé avec les commandes vues précédemment comme l'illustrent les exemples suivants :

- Lister tous les fichiers se terminant par ".c" (eq. tous les fichiers C)

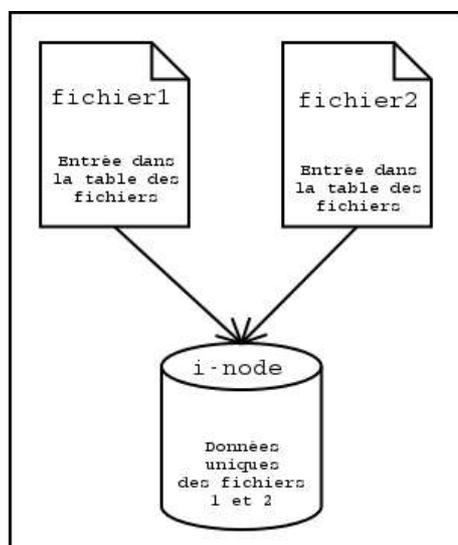
```
> ls *.c
main.c                test.c                mylib.c
```
- Lister les fichiers dont les trois dernières lettres avant l'extension sont indifférentes :

```
> ls fichiers???.dat
fichiers000.dat      fichiers001.dat      fichiers002.dat      fichiers003.dat
fichiers004.dat      fichiers005.dat      fichiers006.dat      fichiers007.dat
fichiers008.dat      fichiers009.dat      fichiers010.dat      fichiers011.dat
```
- Copier tous les fichiers du repertoire "test" vers le repertoire "/home/users/dest/"

```
> cp -R test/* /home/user/dest/
```

E – Utilisation des liens :

Les systèmes Unix permettent la création de liens. Un lien est en quelque sorte un raccourci vers un fichier. C'est est en apparence un fichier comme un autre. Il existe deux principaux sortes de liens : les liens physiques et les liens symboliques. Un fichier est identifié par un numéro d'enregistrement (i-node) qui lui est associé.

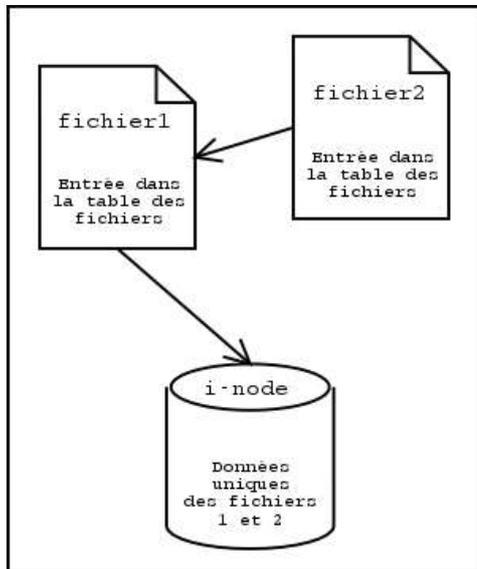


Lorsqu'un lien physique est créé sur un fichier, une nouvelle entrée est créée dans la table des fichiers, mais celle-ci pointe sur le même i-node que le fichier déjà existant. La suppression de l'un des fichier n'entraîne pas la suppression des données, celles-ci ne sont effacées que lorsque tous les fichiers pointant dessus ont été détruits.

Les données étant communes, la modification du fichier 1 entraîne automatiquement celle du fichier 2.

La commande utilisée pour créer un lien est **ln** *src* *dst* créant un fichier *dst* lié au fichier *src*.

```
> ln fichier1 fichier2
> ls -il
(n° i-node [identiques])
391144      fichier1
391144      fichier2
```



Lors de la création d'un lien symbolique, le fichier 2 pointe non pas sur les données du fichier 1 mais sur son entrée dans la table des fichiers. De ce fait, lors de la destruction du fichier1, les données sont, elles-aussi, détruites et le fichier 2 pointe alors dans le vide.

La modification de l'un des fichiers entraîne toujours la modification du second.

La commande utilisée pour créer un lien symbolique est **ln -s src dst** créant un fichier *dst* lié au fichier *src*.

```
> ln -s fichier1 fichier2
```

```
> ls -il
```

```
(n° i-node [différents])
```

```
391144      fichier1
```

```
391142      fichier2 -> fichier1
```

Le symbole '->' indique le chemin du fichier lié.

F – Edition d'un fichier :

L'éditeur de fichier que vous trouverez sur tous les système Unix est **vi**. Cet éditeur est certes peu pratique pour un novice mais connaître quelques commandes de base se révèle très souvent utile. Il prend comme paramètre le nom du fichier à éditer.

Vi est un éditeur non graphique et sans menu. Il possède deux modes : édition et commande. Le mode commande permet de sauver, quitter, copier, coller... le document. Le mode édition permet d'ajouter ou modifier du texte. A tout moment, il est possible de passer en mode commande en pressant la touche *ESC* du clavier. Le passage en mode édition, se fait en pressant la touche '*i*' pour une édition avec insertion à la position courante du curseur. '*a*' pour une insertion à la position suivant le curseur. '*R*' pour le remplacement.

Le mode commande permet de multiples commandes, certaines sont détaillées si-dessous.

- q Quitter vi (sans sauvegarde).
- q! Quitter vi y compris si le fichier a été modifié et non sauvé.(sans sauvegarde).
- w Sauvegarder le fichier.
- w! Sauvegarder le fichier même si celui-ci est en lecture seule (selon droits).
- dd Supprimer la ligne courante (couper). Plusieurs lignes si précédé du nombre. (expl 2dd)
- yy Copier la ligne courante. Plusieurs lignes si précédé du nombre.
- P Coller les lignes précédemment coupées ou copiées.
- /str Rechercher la chaîne *str*. La commande '/' seule recherche l'occurrence suivante de la même chaîne.
- :num Aller à la ligne *num*.
- x Supprimer le caractère courant.

G – Recherche d'un fichier :

La commande **find** permet de rechercher un fichier sur le système. Cette commande prend pour paramètre le répertoire de base à partir duquel la recherche est effectuée et le nom du fichier recherché. La recherche est récursive. De nombreuses autres options rendent cette commande très puissante.

Exemple d'utilisation : **find / -name "fichier1"** recherche le fichier "*fichier1*" dans tout le système.

II. Propriété et droits des fichiers

A – La propriété d'un fichier :

Lorsqu'un fichier est créé, il devient la propriété de son créateur et celle du groupe principal de l'utilisateur. Un fichier appartient toujours à la fois à un utilisateur et à un groupe.

```
> vi fichier1
> ls -l
-rw-r--r-- 1 paul users 5 2003-03-03 18:24 fichier1
```

Le fichier "fichier1" appartient à l'utilisateur qui vient de le créer : "paul" et à son groupe d'appartenance principal : "users".

Il est possible de modifier ces propriétés à l'aide de la commande *chown* .

```
> chown jean fichier1
> ls -l
-rw-r--r-- 1 jean users 5 2003-03-03 18:24 fichier1
Le nouveau propriétaire du fichier est l'utilisateur jean.
```

```
> chown :friendly fichier1
> ls -l
-rw-r--r-- 1 jean friendly 5 2003-03-03 18:24 fichier1
Le caractère ":" précédant le nom indique que l'on modifie le groupe propriétaire, il devient alors friendly.
```

```
> chown paul:users fichier1
> ls -l
-rw-r--r-- 1 paul users 5 2003-03-03 18:24 fichier1
La combinaison des deux est possible.
```

B – Les droits d'un fichier :

Un fichier possède trois types de droits : lecture, écriture et exécution. Ils sont associés à l'utilisateur propriétaire, au groupe propriétaire et aux autres utilisateurs. Ces 3 catégories sont affichées lors d'un *ls -l* :

```
> ls -l
-rw-r--r-- 1 paul users 5 2003-03-03 18:24 fichier1
L'utilisateur possède un droit en lecture et écriture (rw- à gauche).
Le groupe possède un droit en lecture seule (r-- au milieu).
Les autres utilisateurs possèdent un droit en lecture seule (r-- à droite).
Les '-' indiquent que le droit correspondant n'est pas donné.
```

La commande *chmod* permet de modifier ces droits, il existe plusieurs façons de faire : textuellement les options *u,g,o* permettent d'indiquer quelle catégorie est à modifier, les options + et - activent ou désactivent le droit, enfin *r,w,x* indiquent le droit concerné. L'option *a* indique que toutes les catégories sont concernées.

```
> chmod og+rw fichier1
> ls -l
-rw-rw-rw- 1 paul users 5 2003-03-03 18:24 fichier1
Le groupe et les autres utilisateurs ont obtenu le droit en lecture et écriture sur le fichier.
```

```
> chmod a-w fichier1
> ls -l
-r--r--r-- 1 paul users 5 2003-03-03 18:24 fichier1
Tous les utilisateurs ont perdu le droit d'écriture sur le fichier.
```

Le second moyen de modifier les droits et d'entrer une valeur numérique correspondant au droit souhaité. Cette méthode est assez courante. Chaque catégorie est représentée par un digit, la valeur de ce digit correspond à la somme des droits. Le droit de lecture vaut 4, celui d'écriture 2 et celui d'exécution 1. Alors un droit de lecture-écriture vaut $4+2 = 6$.

```
> chmod 640 fichier1
> ls -l
-rw-r----- 1 paul users 5 2003-03-03 18:24 fichier1
6 = rw pour l'utilisateur, 4 = r pour le groupe, 0 = aucun droit pour les autres.
```

Les droits d'un répertoire s'appliquent aussi aux fichiers qui sont dans ce répertoire et dans tous les sous répertoires.

III. Utilisateurs et groupes

A – Gestion des utilisateurs

Les utilisateurs sont inscrits dans le fichier */etc/passwd*. Son format est le suivant, pour chaque utilisateur du système :

```
login:password:uid:gid:nom_complet:répertoire_home:shell_par_défaut
Login est son nom court utilisé pour se connecter.
Password son mot de passe crypté selon l'algorithme RC5.
Uid est un numéro unique identifiant formellement l'utilisateur.
Gid est le numéro du groupe principal de l'utilisateur (les fichiers que l'utilisateur crée auront ce groupe comme groupe propriétaire par défaut).
Nom_complet permet d'avoir un nom long associé au login.
Répertoire_home est le chemin d'accès au répertoire personnel de l'utilisateur.
Shell_par_défaut indique quel type de ligne de commande sera lancé lors de sa connexion.
```

Un utilisateur peut être ajouté manuellement par l'administrateur ou à l'aide d'une commande dédiée *useradd*. La commande *useradd* permet de spécifier les groupes secondaires de l'utilisateur. L'appartenance à un ou plusieurs groupes secondaires permet d'obtenir des droits sur les fichiers et programmes appartenant à ces groupes.

```
> useradd -g users -G word -d /home/newuser newuser
Ajoute un utilisateur "newuser" dont le répertoire personnel sera "/home/newuser". Il aura comme groupe principal "users" et appartiendra de façon secondaire au groupe "word".
```

Le mot de passe doit aussi être configuré, la commande *passwd* suivi du login de l'utilisateur concerné permet d'effectuer cet opération.

Un compte peut être désactivé en insérant un '!' dans le fichier */etc/passwd* dans la colonne *passwd*.

La commande *groups* suivie d'un nom d'utilisateur liste les groupes auquel il appartient.

Il est possible de changer d'identité sans avoir besoin de fermer la connexion en cours au travers de la commande *su* suivie du nom de login à utiliser. Le mot de passe est alors demandé, hors mis pour l'utilisateur root qui peut ainsi faire des installations en temps qu'utilisateur sans avoir besoin de connaître le mot de passe.

Cette technique est aussi très utile pour l'administration distante car elle permet de ne pas échanger d'informations concernant l'utilisateur "root" sur le réseau.

B – Gestion des groupes

La liste des groupes d'un système est définie dans le fichier */etc/group*. Son format est le suivant pour chaque groupe :

nom_group:group_password:gid:utilisateur1, utilisateur2, ..., utilisateur n

nom_group indique le nom du groupe.

group_password indique le mot de passe pour le groupe (peu utilisé)

gid : indique le numéro unique identifiant le groupe (utilisé dans le fichier */etc/passwd*)

utilisateur : indique la liste des utilisateurs appartenant à ce groupe.

Un groupe est créé par la commande “*groupadd nom_groupe*” ou édité à la main.

Un groupe peut être administré par un utilisateur autre que *root*. Cet utilisateur aura le droit d'ajouter, supprimer des utilisateurs dans le groupe. Cette possibilité requiert l'usage de la commande *gpasswd*. L'association d'un utilisateur comme administrateur de groupe se fait par la commande *gpasswd -A login_user nom_groupe*.

Un utilisateur peut demander une appartenance temporaire à un groupe, par exemple pour l'accès à un repertoire partagé que l'on souhaite protéger par un mot de passe. La commande “*sg nom_groupe*” permet cela. Le mot de passe du groupe est alors demandé.

Un utilisateur peut aussi demander le changement de son groupe principal au travers de la commande *newgrp* suivie du nom du groupe. Le mot de passe du groupe sera demandé si l'utilisateur n'est pas inscrit dans ce groupe.

Un utilisateur est associé à un groupe (qui lui sera secondaire) en ajoutant son login dans la colonne *utilisateur* du fichier */etc/group*. Un utilisateur peut avoir plusieurs groupes secondaires, le nombre maximum de groupe par utilisateur dépend du système.

IV.Recherche d'informations

Il est très souvent nécessaire de rechercher des informations sur les commandes existantes car leur utilisation poussée passe par la connaissance de très nombreuses options. Pour simplifier cela, chaque commandes, comme de nombreux fichiers de configuration sont documentés. Cette aide est accessible au travers de plusieurs outils :

1. *man nom_commande* permet d'accéder aux pages de manuel.
2. *info nom_commande* permet d'accéder à une aide plus complète intégrant une navigation.
3. */usr/share/doc/packages* (variable selon les distributions) contient des documents textes complémentaires.
4. Les *HowTo* et les *F.A.Q.* traitent de sujets plus vastes et sont accessibles dans la plupart des distributions ou sur Internet.

V. Protection des fichiers ayant les mots de passe :

Un des points vulnérable d'un système Unix est l'accès aux mots de passe. En effet, même s'il sont cryptés et indécryptables, il reste toujours possible de les comparer à des mots sortis d'un dictionnaire. Ainsi, les mots de passe trop simples pourront être trouvés.

Les fichiers `/etc/passwd` et `/etc/group` doivent être accessibles en lecture par tous les utilisateurs pour que le système puisse fonctionner. Cette nécessité posait donc de gros problèmes de sécurité. Pour remédier à cela, les mots de passe ont été retirés de ces fichiers et placés dans d'autres. Ces nouveaux fichiers ne sont alors accessibles en lecture que par l'administrateur. Cette technique est appelée *shadow password*. La prise en compte de ce système est transparente lors de l'utilisation des commandes `useradd`, `groupadd` et `passwd`. Toutefois vous pourrez constater son effet si vous éditez les fichiers `/etc/passwd` et `/etc/shadow` par exemple. Il est à noter d'ailleurs que le nom du fichier shadow peut varier selon les distributions.

Deux commandes permettent d'activer et de désactiver le système de *shadow password* :

`pwunconv` désactive la protection et met à jour le fichier `/etc/passwd`.

`pwconv` active la protection et met à jour le fichier `/etc/shadow`.

Cette protection est aussi valable pour le fichier de gestion des groupes, `/etc/gshadow` est alors la copie protégée intégrant les mots de passe.

VI. Le réseau :

TCP/IP est nativement supporté par les systèmes UNIX. Quelques commandes servent à le configurer et connaître son status.

`ifconfig` permet de configurer le protocole TCP/IP pour chaque périphérique de sorte à associer une adresse IP et un masque de réseau, activer ou non une carte.

`route` permet d'ajouter, supprimer des routes vers d'autres réseaux. Cette commande permet donc de configurer les passerelles à utiliser pour joindre les réseaux distants.

`/etc/resolv.conf` permet de configurer l'adresse des serveurs DNS.

`ping` est un outil permettant l'échange d'un écho entre deux périphériques. Ce programme sera très utile pour vérifier le bon fonctionnement d'une liaison.

`nslookup nom_serveur` permet d'obtenir l'adresse IP d'une machine à partir de son nom, cette commande met en oeuvre la résolution de nom par DNS.

`traceroute` permet de savoir par quelles passerelles passe une trame allant de votre poste à une destination indiquée. Une version graphique permet de visualiser le résultat sur une mappemonde.

`nmap` permet d'analyser les services offerts par une machine distante. C'est un outil efficace pour vérifier qu'il n'y ait pas sur un de vos serveurs de ports ouverts superflus.

`telnet` permet de se connecter à un ordinateur distant pour l'administrer. Ce protocole n'est pas sécurisé et les mots de passe sont diffusés sans cryptage sur le réseau. De ce fait, il tend à disparaître au profit de `ssh`.

`ftp` permet la copie de fichiers entre ordinateurs dès lors qu'un serveur adapté est installé et configuré.

`ssh` permet aussi une connexion à un ordinateur distant. Cette fois, la communication est cryptée, les mots de passe ne sont plus accessibles. Ssh intègre d'autres commandes comme `scp` qui permet de copier un fichier. Ssh permet aussi d'encapsuler des informations graphiques pour l'affichage de fenêtres à distance. Ssh demande l'installation d'un serveur, opération généralement automatisée par l'installation de la distribution.