

Quelques commandes UNIX

- Les commandes Unix interprétées par le « Shell » sont sensibles à la casse : elles font la différence entre les MAJUSCULES et les minuscules.
- Unix gère les droits d'appartenance, de modification des répertoires comme ceux des fichiers : pour Unix, tout est fichier.
- Les noms des fichiers et des répertoires peuvent comporter tous les caractères alphanumériques (lettres et chiffres), des soulignés, des tirets et des points. Leur longueur est limitée à 253 caractères.

Commandes communes aux fichiers et répertoires :

ls : affichage des répertoires et fichiers

ls -l : affichage des droits, du nom du propriétaire et de son groupe sur les fichiers et répertoire locaux.

ls *.ext : affichage des fichiers et des répertoires ayant l'extension *.ext*

mv ancien_nom nouveau_nom : changement de nom d'un fichier ou d'un répertoire

find . -name nom : recherche du répertoire ou du fichier *nom*.

Gestion des répertoires :

pwd : affichage du nom du répertoire courant.

mkdir *nomdir* : création du répertoire *nomdir* dans le répertoire local

rmdir *nomdir* : destruction du répertoire *nomdir* situé dans le répertoire local s'il est vide => destruction préalable des fichiers contenus dans *nomdir*

rm -r *nomdir* : destruction récursive et irréversible de tout le contenu du répertoire *nomdir* et de *nomdir*. DANGER !

cd *nomdir* : changement de répertoire, le répertoire courant devient *nomdir*

cd : le répertoire courant devient le Home Directory : répertoire courant lors de l'entrée sur votre compte utilisateur.

cd .. : pour remonter d'un cran dans l'arborescence.

cd . : aucun changement

cd /res/an_1/toussaint/seance1 : positionnement du répertoire courant dans /res/an_1/toussaint/seance1, le caractère '/' est le séparateur de noms de répertoires.

cp -r *nomdir1 nomdir2* : copie de manière récursive tous les fichiers et répertoires de *nomdir1* dans *nomdir2* en préservant l'arborescence. Si le répertoire *nomdir2* existe avant copie, son contenu est préservé et une copie du répertoire *nomdir1* est placée dans le répertoire *nomdir2*.

Gestion des fichiers :

more *nomfichier* : affichage du contenu du fichier *nomfichier*

cp *chemin1/nomfichier1 chemin2/nomfichier2* : copie *nomfichier1* dans *nomfichier2*, si *nomfichier2* existe alors son contenu est remplacé par celui de *nomfichier1*.

cmp *chemin1/nomfichier1 chemin2/nomfichier2* : comparaison rapide des contenus des fichiers *nomfichier1* et *nomfichier2*.

diff *chemin1/nomfichier1 chemin2/nomfichier2* : comparaison détaillée des contenus des fichiers *nomfichier1* et *nomfichier2*.

grep *''chaine'' nomfichier* : affichage des lignes du fichier *nomfichier* contenant la chaîne de caractère *''chaine''*.

grep -v *''chaine'' nomfichier* : affichage des lignes du fichier *nomfichier* ne contenant pas la chaîne de caractère *''chaine''*.

tail -n *nomfichier* : affichage des *n* dernières lignes du fichier *nomfichier*.

head -n *nomfichier* : affichage des *n* premières lignes du fichier *nomfichier*.

Autres commandes UNIX :

man *commande* : affichage du manuel d'utilisation d'une commande.

Les commandes UNIX sont souvent complétées par des options, la syntaxe générale est la suivante :

[nom de commande] [options] [nom]

Par exemple pour la commande "rm" d'effacement de fichier, on utilise les options suivantes

rm -r : (récursif) une destruction récursive appliquée à un fichier qui est un répertoire implique la destruction de tous les fichiers du répertoire et du répertoire lui-même.

rm -i : (interactif) confirmation par yes/no avant la destruction du fichier. Une destruction interactive évite des dégâts irrémediables.

La commande "ls" possède aussi des options :

ls -l : affiche le contenu d'un répertoire avec les informations suivantes

nature du fichier : un - pour un fichier ordinaire

un d (directory) pour un répertoire

un l (link) pour un lien symbolique

droits d'accès : pour le propriétaire (owner), pour le groupe (group), pour les autres utilisateurs (others) ; les droits sont affichés par une lettre (**r** pour lecture, **w** pour écriture, **x** pour exécution) nom du propriétaire; taille du fichier en caractères; date de dernière modification ; nom du fichier.

Exemples:

Droits	Propriétaire	Groupe	Taille	Date de création	Nom
drwxr-x---	utilisateur1	ens	64	Apr 1 11:43	an_1

"an_1 " est un répertoire (d) appartenant à " utilisateur1" du groupe "ens" avec les droits suivants

propriétaire : lecture (r) écriture (w) exécution (x)

groupe : lecture (r) exécution (x)

les autres : aucun droit

Droits	Propriétaire	Groupe	Taille	Date de création	Nom
-rw-r -----	g3xl	g3	1024	Dec 25 00:00	pgm.c

"**pgm.c**" est un fichier (-) appartenant à "g3xl" du groupe "g3" avec les droits suivants :

propriétaire : lecture (r) écriture (w) groupe : lecture (r)

les autres aucun droit

Les droits d'accès sous UNIX :

Pour assurer des protections sur un fichier, UNIX classe les utilisateurs potentiels d'un fichier en trois catégories :

son propriétaire, c'est-à-dire le créateur du fichier;

les usagers qui font partie du même groupe que le propriétaire

les autres, excluant le propriétaire et les usagers du groupe

A chacune de ces trois classes d'utilisateurs, trois types de droits d'accès sont définis

r (read) droit de lire le fichier;

w (write) droit de modifier le contenu du fichier;

x (execute) droit d'exécuter le fichier.

Changement des protections : **chmod**

Les droits établis à la création d'un fichier peuvent être modifiés par la suite par la commande :
`chmod mode nomdir ou nomfichier`

L'option "mode" est composé de trois champs collés les uns aux autres, de la façon suivante :
[qui] position [droit]

qui est l'un des caractères suivants

u (user) pour le propriétaire;

g (group) pour le groupe;

o (other) pour les autres;

a (all) pour tous.

Position est l'un des caractères suivants

+ pour ajouter un droit

- pour enlever un droit

= pour mettre un droit et supprimer les autres

droit est l'un des caractères suivants :

r pour le droit de lecture

w pour le droit d'écriture

x pour le droit d'exécution

Exemples de chmod

Commande	Permission avant	Permission après
chmod u+x ficl.txt	rw-r-----	rwxr-----
chmod u-w ficl.exe	rw-r-----	r--r-----
chmod a=r ficl.exe	rw-rw-rw-	r--r--r--

Pour changer les droits sur un répertoire ainsi que sur tous les répertoires et fichiers contenus, on complète la commande chmod avec l'option **-R** (récursivité), comme par exemple :

chmod -R a-w *nomdir* suppression des droits en écriture de manière récursive dans tout le répertoire *nomdir*.

Compression de fichiers :

La commande "**compress**" permet de réduire (autrement dit compresser) la place occupée par un fichier. Le fichier porte alors le même nom avec l'extension **.Z** permettant de reconnaître le type de commande utilisée lors de la compression. La syntaxe est la suivante :

compress *nomdufichier* => *nomdufichier.Z*

L'opération inverse est réalisée par la commande **uncompress**

uncompress *nomdufichier.Z* => *nomdufichier*

La tendance actuelle est d'utiliser de nouveaux utilitaires de compression - décompression :

gzip *nomdufichier* => *nomdufichier.gz* ; **gzip -d** *nomdufichier.gz* => *nomdufichier*

bzip2 *nomdufichier* => *nomdufichier.bz2* ; **bzip2 -d** *nomdufichier.bz2* => *nomdufichier*

Archivage de données :

La commande "**tar**" permet d'archiver (ou extraire) des fichiers ou des répertoires en conservant leur arborescence sur supports amovibles, cartouches et sur disques.

La commande utilisée pour archiver est :	tar cvf <i>nom_archive.tar nomdir</i>
pour desarchiver	tar xvf <i>nom_archive.tar</i>
pour visualiser le contenu d'une archive :	tar tf <i>nom_archive.tar</i>

Les options possibles de tar sont :

- t** liste les noms des fichiers de l'archive
- x** extraire les fichiers de l'archive
- c** créer l'archive
- v** mode verbeux (verbose)
- f** nom de l'archive

Notons qu'il est possible de compresser une archive pour économiser en place disque.

Intérêt des commandes en lignes : script shell

Dans le cadre de petits programmes interprétables qu'on appelle des scripts, il est possible d'automatiser certaines procédures comme les installations de logiciels. Taper l'exemple suivant dans un éditeur de texte et sauver le fichier sous le nom *install*

```
#!/bin/sh
echo ''copie de seance1''
cp -r /res/an_1/toussaint/seance1 .
cd seance1
echo ''repertoire courant : '' `pwd`
echo ''compilation de seance.cc ''
g++ seance.cc -o exe -lm
echo ''execution''
./exe
```

Pour exécuter ce script, taper dans une fenêtre terminal : sh install.

Ou encore, rendre le script exécutable avec la commande : chmod +x install puis taper ./install.