

Commandes UNIX utiles en TP

Résumé

Ce document peut constituer un aide mémoire pour les commandes shell les plus utiles en TP. *Il n'est pas complet*. En cas de problèmes, la seule référence correcte est le manuel UNIX (voir la partie 2).

Table des matières

1 Introduction	1
2 Le manuel UNIX	2
3 Gestion et déplacement des arborescences	2
4 Gestion et Manipulation des fichiers	3
5 Archivage et compression	7
6 Gestion des programmes en cours d'exécution	7
7 Travailler à distance	7
8 Commandes diverses	8

1 Introduction

Chaque série de caractères tapée et se terminant par un retour chariot (entrée, **Enter**) est interprétée par le shell comme un commande (première série de caractères jusqu'à un séparateur) suivie de ses arguments. Les séparateurs sont l'espace et la tabulation.

Syntaxe générale des commandes UNIX :

commande [argument(s)]

En général, une commande lit ses données sur le clavier (on parle l'entrée standard) et écrit les données en sortie sur l'écran (la sortie standard). Les erreurs sont écrites sur l'écran (la sortie en erreur standard.

ATTENTION : sous UNIX, les minuscules et les majuscules sont différenciées.

Par exemple, **LICENCE** et **licence** sont deux chaînes de caractères différentes.

2 Le manuel UNIX

La commande `man nom` permet de visualiser l'aide de la commande ou de la fonction spécifiée.

Exemple : `man ls` : Affiche l'aide de la commande `ls`.

3 Gestion et déplacement des arborescences

Visualisation du contenu d'un répertoire

`ls Options`

La commande peut prendre en argument :

- un ou plusieurs répertoires
- une liste de fichiers

Options :

-a : Affiche également les fichiers cachés

-l : Affiche toutes les informations sur les fichiers (droits d'accès, propriétaire, taille, dernière date de mise à jour)

Exemples :

`ls -l`

Affichage à l'écran :

```
-rw-r-r- 1 hamon hamon 902 Nov 25 13:33 exo1.c
drwxrwx-- 2 hamon hamon 512 Dec 30 10:43 Repertoire1
```

`ls -al`

Affichage à l'écran :

```
drwxr-xr-x 2 hamon hamon 512 Nov 25 13:33 .
drwxr-xr-x 4 hamon hamon 512 Nov 25 13:33 ..
-rw-r-r- 1 hamon hamon 902 Nov 25 13:33 .cshrc
-rw-r-r- 1 hamon hamon 902 Nov 25 13:33 exo1.c
drwxrwx-- 2 hamon hamon 512 Dec 30 10:43 Repertoire1
```

Déplacement dans un répertoire

`cd nom du répertoire`

Exemples :

- `cd Repertoire1` (Déplacement dans le répertoire `Repertoire1`)
- `cd /u/HOME/h/hamon/Repertoire1` (Déplacement dans le répertoire `/u/HOME/h/hamon/Repertoire1`)
- `cd` (Retour au répertoire de connexion)
- `cd ..` (Retour au répertoire père)

Affichage du répertoire courant

`pwd`

Exemple : pwd
Affichage à l'écran :
/u/HOME/h/hamon/Repertoire1

Création d'un répertoire

mkdir nom du répertoire
Exemple : mkdir Repertoire1 (création du répertoire Repertoire1)

Destruction d'un répertoire

rmdir nom du répertoire
Exemple : rmdir Repertoire1 (destruction du répertoire Repertoire1)
Note : le répertoire doit être vide.

4 Gestion et Manipulation des fichiers

Copie de fichiers

cp options fichier1 fichier2
(Copie du contenu de *fichier1* dans *fichier2*)
ou
cp options fichier1 [fichier2 ...] répertoire
(Copie du contenu des fichiers *fichier1* et *fichier2* dans *répertoire*)
Options :
– *-i* : mode interactif (confirmation avant écrasement) ;
– *-r* : copie récursive de répertoires ;
– *-p* : conserve la date du fichier source.
Exemples :
cp /etc/fstab .
cp -r rep1 rep2

Déplacement de fichiers

mv options fichier1 [fichier2 ...] répertoire
Déplace les fichiers *fichier1* et *fichier2* dans *répertoire*.
Options :
– *-i* : mode interactif (confirmation avant écrasement).
– *-f* : force la commande
Exemple :
mv fichier1 rep1

Destruction de fichiers

```
rm options fichier1 [fichier2 ...]
```

Suppression des fichiers spécifiés.

Options :

- *-i* : mode interactif (confirmation avant écrasement).
- *-f* : force la commande
- *-r* : supprime récursivement les répertoires.

Exemples :

```
rm fichier1
rm -r repertoire1
```

Visualisation ou concaténation de fichiers

```
cat fichier...
```

Affiche à l'écran (la sortie standard) le fichier donné en argument ou les concatène (ajout les uns à la suite des autres) s'il y en a plusieurs.

Par défaut, la lecture s'effectue sur l'entrée standard et l'affichage sur la sortie standard.

Exemple :

```
cat /etc/fstab
```

Affichage à l'écran du contenu du fichier `/etc/fstab`.

Visualisation du début/de la fin d'un fichier

```
head -n fichier : affichage des n premières lignes d'un fichier
tail -n fichier : affichage des n dernières lignes d'un fichier
```

Par défaut, *n* vaut 10.

Visualisation d'un fichier page par page

```
more fichier...
```

Affichage à l'écran, page par page, des fichiers spécifiés.

Tri des lignes d'un fichier

```
sort -ufnr -o fic fichier...
```

Par défaut, tri par ordre alphabétique.

Options :

-u : n'affiche qu'une fois les lignes multiples.
-f : ne différencie pas les minuscules des majuscules
-n : effectue un tri numérique
-r : ordre décroissant
-o *fic* : spécifie un fichier de sortie.

Autre option :

-k <premier champ><type>,<dernier champ><type>

Par exemple, `sort -k 2n,2n` (Tri sur le deuxième champ).

Décompte des nombres mots/lignes/caractères d'un fichier

`wc -lwc fichier...`

Options :

-l : Affiche que le nombre de lignes
-w : Affiche que le nombre de mots
-c : Affiche que le nombre de colonnes

Si aucune option n'est spécifiée, la commande affiche le nombre de lignes, de mots et de colonnes.

Substitution de caractères

`tr caracteres caracteres < fichier`

Substitue un caractère par un autre (Suppression des caractères invisibles).

On peut utiliser le code hexadécimal du caractère.

Exemple :

`tr '\011' '@' < fichier1`

Affichage de certaines colonnes d'un fichier

`cut -d delim -f champs fichier`

Options :

-d *delim* : spécifie le délimiteur (entre double quote, ");
-f *champs* : spécifie les champs à afficher (séparés par une virgule).

Le délimiteur par défaut est la tabulation.

Exemple :

`cut -f1 /etc/fstab`

Affichage de la première colonne du fichier `/etc/fstab`.

Modification des droits d'un fichier/répertoire

`chmod [-R] <classe d'utilisateur>< nature de la modification>< droit fichiers>`

- [-R] : Opération récursive sur les fichiers et les sous-répertoires
- classe d'utilisateur :
 - u** : le propriétaire;
 - g** : le groupe;
 - o** : les autres;
 - a** : les trois classes.
- nature de la modification :
 - + ajout du droit
 - retrait du droit
 - = affectation du droit
- droit :
 - r** droit en lecture
 - w** droit en écriture
 - x** droit en exécution

Exemples :

```
chmod g-rw fichier1
chmod -x fichier2
```

Recherche de fichiers ou de répertoires

`find répertoire options`

Recherche dans toute l'arborescence à partir du point spécifié.

Options courantes :

- name** *fichier* : recherche sur le nom *fichier*
- print** : affiche le résultat de la recherche
- !** : négation de la recherche

Exemple :

```
find / -name "fstab" -print
```

Recherche des fichiers dont le nom est `fstab`, dans tous les répertoires.

Recherche d'une chaîne de caractères dans un fichier

`grep options expression fichier`

5 Archivage et compression

Les systèmes UNIX offrent de nombreux outils d'archivage et de compression des fichiers. Les commandes les plus couramment utilisés sont `gzip` et `tar` :

- `gzip fichier`

Cette commande permet la compression de fichiers donnés en argument. Le fichier produit possède une extension supplémentaire : `gz`
L'utilisation de l'option `-d` permet la décompression du fichier préalablement. La commande `gunzip fichier` est équivalente à l'utilisation de cette option.

Exemples :

- Compression du fichier `Fichier1.txt` : `gzip Fichier1.txt`
Le fichier compressé s'appelle `Fichier1.txt.gz`
- Décompression du fichier `Fichier1.txt.gz` : `gzip Fichier1.txt.gz`
Le fichier décompressé s'appelle `Fichier1.txt`
- `tar Options FichierArchive Fichiers`
Cette commande permet l'archive (et la compression) d'un ensemble de fichiers et de répertoires mais aussi le désarchivage (et la décompression) d'archives créées précédemment.
 - Archivage (et compression) : `tar czvf Archive.tar.gz Fichiers`
 - Désarchivage (et décompression) : `tar xzvf Archive.tar.gz`
(les fichiers contenus dans l'archive sont créés dans le répertoire courant)

6 Gestion des programmes en cours d'exécution

Les commandes suivantes permettent d'interagir avec les programmes en cours d'exécution sur la machine.

- `ps -aux` : Visualisation de programmes en cours d'exécution avec leur identifiant (PID)
- `kill PID` : Terminaison immédiate du processus de pid `PID`

7 Travailler à distance

`ssh -X machine.e.ujf-grenoble.fr`

Permet d'ouvrir un shell sur une autre machine. Ceci est utile si la machine locale est mal installée ou si elle ne possède pas un logiciel qui est présent sur l'autre machine, etc.

Options courantes :

`-X` : permet de lancer des applications graphiques sur la machine distante

Exemple :

`ssh -X jpp.e.ujf-grenoble.fr`

Ouvre un shell dans le terminal courant pour lancer des commandes sur la machine `jpp`

8 Commandes diverses

Affichage de texte sur la sortie standard

`echo -n 'message'`
Option : `-n` Pas de retour chariot final

Effacement de l'écran

`clear`
(Pas d'option)