

Module GNU/Linux CEFIPA : TP n°1 - Initiation au shell

Corrigé

Nicolas Burrus

14 Avril 2005

Question 1

Lire le manuel de "ls" pour expliquer le résultat de la commande suivante :

```
$ ls -srF /etc
```

Réponse :

D'après `man ls` :

- [-s] : affiche en plus la taille des fichiers en "blocks". Les "blocks" sont les unités élémentaires de stockage sur le disque dur, et sont définis par le système de fichier. Pour ext2, chaque bloc vaut 1Ko par défaut.
- [-r] : ls liste les fichiers par ordre alphabétique. Cette option trie dans l'ordre inverse.
- [-F] : affiche à la fin des noms de fichier un indicateur sur leur type, par exemple '/' pour un répertoire, '@' pour un lien.

Question 2

Donner une commande qui affiche :

```
Hello <user>, welcome on <host>
```

avec <user> et <host> remplacés par le nom de l'utilisateur actif et par le nom de la machine.

Réponse :

`export` liste les variables d'environnement définies. Sur une Knoppix, `$HOME` et `$HOSTNAME` contiennent les informations demandées.

`echo "$USER, welcome on $HOSTNAME"` affiche donc le résultat demandé. Note : les guillemets sont optionnels ici, `echo` affiche le contenu de tous ses arguments de toute façon.

Question 3

Lister les fichiers présents dans tous les répertoires du PATH. Noter les 3 premiers de chaque répertoire. Trouver l'emplacement du programme `info`. Pouvez vous trouver l'emplacement de `cd` et `echo` ? Pourquoi ?

Réponse :

`echo $PATH` affiche le contenu du PATH. Les répertoires dans lesquels seront recherchés les binaires sont séparés par des `:`. Par exemple, si le PATH vaut `/bin :/sbin :/usr/bin :/usr/X11R6/bin`, le shell cherchera les binaires dans `/bin`, puis `/sbin`, puis `/usr/bin` et enfin `/usr/X11R6/bin`.

Ici, il suffit de faire un `ls` dans chaque répertoire du `PATH` et de noter les 3 premiers. Note : `ls -r` permet d'afficher les 3 premiers fichiers à la fin.

`cd` n'est pas un binaire, vous ne le trouverez donc pas sur le disque dur. Il s'agit d'une commande "builtin" interne au shell, qu'il comprend d'une manière particulière. Une recherche dans `man sh` permet de trouver la liste des commandes builtins.

`echo` est à la fois une commande builtin et un binaire sur le système.

Les commandes `where` ou `which` permettent de trouver l'emplacement d'une commande directement.

Question 4

Donner une commande permettant de créer l'arborescence `$HOME/a/b/c/d/e/f/` en une seule ligne (regarder la page `info` de `mkdir`).

Réponse :

La page `info` nous apprend que l'option `[-p]` permet de créer les répertoires parents si ils n'existent pas. La commande : `mkdir -p $HOME/a/b/c/d/e/f` crée donc l'arborescence d'un coup. Autres commandes équivalentes : `mkdir -p ~/a/b/c/d/e/f` ou `cd ; mkdir -p a/b/c/d/e/f`.

Question 5

Créer le fichier vide `test1` dans `$HOME/a/b/c/d/e/f/` et les fichiers vides `test2` et `test3` dans `$HOME/a/b/c/d/`. Supprimer le répertoire `$HOME/a/b/c/d/e`. Qu'est devenu le fichier `test1` ?

Réponse :

Exemple de séquence de commandes :

```
$ cd ~/a/b/c/d
$ touch e/f/test1 test2 test3
$ rm -r e
```

Le fichier `test1` a été supprimé lors de la suppression de `e`.

Note : par défaut sous Knoppix, `rm` demande confirmation pour chaque fichier concerné par la suppression d'un répertoire. Ce n'est pas le comportement par défaut de `rm`, mais un alias `rm='rm -i'` a été défini. Pour contrer cette option, il faut soit supprimer l'alias avec `unalias rm`, soit ajouter l'option `[-f]:rm -rf e`.

Question 6

Modifier l'arborescence pour obtenir `$HOME/a/c/d/` sans perdre les fichiers `test2` et `test3` dans `d/` (`b` doit être supprimé).

Réponse :

Exemple de séquence de commandes :

```
$ cd ~/a
$ mv b/c .
$ rm -rf b
```

La 2ème commande déplace le répertoire `c` dans le répertoire courant (`.`), qui est `~/a` suite au premier `cd`.

Question 7

Donner la taille de `/bin/cat` de façon "humaine", c'est à dire en Ko, en Mo ou en Go selon la taille du fichier (info `ls`).

Réponse :

Exemple de commande :

```
$ ls -lh /bin/cat
-rwxr-xr-x 1 root root 17K Jul 16 2004 /bin/cat
```

La sortie du `ls -lh` nous donne les détails du fichier, avec notamment la taille, affichée de façon "humaine" grâce à l'option `[-h]`. La taille de `/bin/cat` est donc 17Ko ici.

Question 8

Faire en sorte que :

```
$ cd
$ ls doc
```

liste le contenu le `/usr/share/doc`.

Réponse :

Il faut créer ici un lien symbolique, plus "souple" que les liens durs, qui ne s'appliquent pas aux répertoires :

```
$ cd
$ ln -s /usr/share/doc doc
```

Note : `ln -s /usr/share/doc .` donne également le bon résultat, `ln` prend le nom simple du fichier destination par défaut, et place ici le lien dans '.', à savoir le répertoire courant.

Question 9

Télécharger <http://nestechn.free.fr/pub/cefipa/tp1-files.tar.gz>. Décompressez l'archive. Les extensions des fichiers à l'intérieur ne correspondent pas au type de leur véritable contenu. Remettre les bonnes extensions.

Pour télécharger l'archive, taper les commandes suivantes :

```
$ cd
$ wget http://nestechn.free.fr/pub/cefipa/tp1-files.tar.gz
```

Réponse :

La commande `file` permet de donner le type d'un fichier à partir de son contenu :

```
$ cd
$ wget http://nestechn.free.fr/pub/cefipa/tp1-files.tar.gz
$ tar xvfz tp1-files.tar.gz
tp1-files/
tp1-files/redhat.jpg
tp1-files/sujet.doc
tp1-files/sujet.pdf
tp1-files/readme.txt
$ cd tp1-files
$ file *
readme.txt: gzip compressed data, was "README", from Unix, max compression
```

```
redhat.jpg: PNG image data, 261 x 300, 8-bit/color RGB, non-interlaced
sujet.doc:  PDF document, version 1.4
sujet.pdf:  Microsoft Office Document
$ mv readme.txt readme.gz
$ mv redhat.jpg redhat.png
$ mv sujet.doc sujet-tmp.pdf
$ mv sujet.pdf sujet.doc
$ mv sujet-tmp.pdf sujet.pdf
```

L'option [-v] de tar lui demande d'afficher le nom des fichiers qu'il décompresse.

Pour les fichiers sujet.*, il faut passer par un fichier intermédiaire, sinon le premier mv écraserait le sujet.pdf existant.

Question 10

Créer une archive modules.tar.gz contenant le répertoire /lib/modules, qui en se décompressant crée un répertoire modules/ avec le contenu de /lib/modules dedans. Exemple de décompression :

```
$ tar xfz modules.tar.gz
$ cd modules
$ ls
```

La sortie du dernier ls doit être équivalente à :

```
$ cd /lib/modules
$ ls
```

Réponse :

La commande tar inclut le chemin complet des fichiers dans l'archive, par exemple, avec :

```
$ cd /
$ tar cf /tmp/archive.tar usr/X11R6/bin
$ cd
$ tar xvf /tmp/archive.tar
```

les fichiers de /usr/X11R6/bin seront décompressés dans le répertoire \$HOME/usr/X11R6/bin.

Ici, il faut donc faire attention au chemin utilisé pour créer l'archive, en se plaçant dans le bon répertoire :

```
$ cd /lib
$ tar cfz ~/modules.tar.gz modules
$ cd
$ tar xfz ~/modules.tar.gz
$ ls -F
modules/
```

Question 11

Supprimer tous les répertoires créés dans \$HOME.

Réponse :

```
$ cd
$ rm -rf *
```

Attention avec la commande rm -rf *, elle supprime tout ce qui est dans le répertoire courant, ça peut être très dangereux.

Question 12

Monter la partition Windows du disque dur de la machine dans le répertoire /mnt/windows.
Donner la ligne contenant "execstd" dans le fichier C:\windows\system\setup.inf.
Attention, ces commandes doivent être exécutés sous l'identité de l'administrateur "root". Pour changer d'utilisateur, utiliser la commande `su`. Penser à retourner en utilisateur normal après ces commandes (commande `exit`).

Réponse :

```
$ su
# cd /mnt
# mkdir windows
# mount /dev/hda1 windows
# cd windows/WINDOWS/SYSTEM/
# less SETUP.INF
# exit
```

`less` peut être utilisé pour rechercher une chaîne de caractère, avec la touche `('/')`. Il suffit de taper `/execstd` ici pour trouver la bonne ligne.

Question 13

Etape 1

Donner une commande qui liste les fichiers/répertoires de `/usr/share/doc` en excluant :

- Les fichiers commençant par "lib"
- Les fichiers commençant par les lettres `a, b, c, d, e, s, x`

Réponse :

```
$ cd /usr/share/doc
$ ls --ignore 'lib*' --ignore '[a-esx]*'
```

L'option `-ignore` permet de spécifier un motif à exclure. Les motifs compris par `ls` sont les mêmes que ceux compris par le shell. Seulement, ici, c'est `ls` qui doit traiter les motifs, pas le shell, il faut donc mettre les motifs entre quotes pour éviter l'expansion du shell.

`[a-esx]` permet de spécifier toutes les lettres de 'a' à 'e', puis les lettres 's' et 'x'.

Etape 2

Comment peut-on combiner la commande de l'étape 1 avec `tar` pour créer une archive `$HOME/mydoc.tar.gz` contenant tous les fichiers/répertoires de `/usr/share/doc` à l'exception de ceux cités à l'étape 1 ?

Réponse :

Les backquotes permettent de remplacer un argument par le résultat de son exécution. Il suffit donc de faire :

```
$ cd /usr/share/doc
$ tar cfz ~/mydoc.tar.gz `ls --ignore 'lib*' --ignore '[a-esx]*'`
```