Unix (2)

1 Alias

Il est possible de définir des **alias** sous Unix permettant de personnaliser vos appels de commandes les plus utilisées. Les commandes à utiliser ici sont `alias` et `unalias`. Lancer `man alias`.

1.1 Exercice Afficher la liste des alias définis dans le shell courant.

1.2 Exercice Définir un alias `date` sur la commande `ls`.

1.3 Exercice Détruire l'alias précédent et rappeler la commande `date` afin d'en vérifier son bon fonctionnement.

1.4 Exercice Ouvrir le fichier `.bashrc` se trouvant dans votre répertoire d'accueil. Rajouter les alias suivants à l'intérieur de ce fichier, à la suite des alias déjà existant :

```
alias rm='rm -i'
alias ll='ls -la'
```

2 Substitution de chemin

2.1 Exercice Essayer `echo ~`. Comment peut-on faire afficher le caractère ~ ? Commenter rapidement le mécanisme de **quotation** au moyen du caractère `\`.

Les caractères suivants ont une signification particulière pour bash : `;`, `&`, `(`, `)`, `!`, `<`, `>`, `*`, `?`, `[`, `]`, `~`, `+`, `-`, `@`, `"`, `.`, `'`, `` ` ``, `\`, `!`, `$`.

Pour utiliser ces caractères dans des noms de fichiers ou des répertoires :
- mettre le caractère `\` devant un caractère particulier ou
- entourer tout le nom par `'` et `'` (le caractère `'` lui-même ne doit pas apparaître dans le nom).

2.2 Exercice Essayer successivement :

```
$ touch '*'
$ ls
$ rm \*
$ ls
```

2.3 Exercice Essayer les commandes suivantes :

```
$ cd ~/staroffice6.0
$ echo *.html
$ cd ..
$ echo staroffice6.0/*.html
$ echo */*
$ echo */*a*
```

Le caractère spécial `*` remplace n'importe quelle suite de caractères dans le chemin.

IMPORTANT : le résultat de la substitution peut être une liste de mots. Dans les deux cas suivants :

```
echo */*
ls -F */*
```

c'est *bash* qui construit la liste des chemins.

3 Création et suppression de répertoires

3.1 Exercice Créer un répertoire `tmp` et un répertoire `Projets` dans son répertoire d'accueil. Plusieurs possibilités, par exemple :

```
$ mkdir ~/tmp
$ cd
$ mkdir Projets
```

Si le chemin est celui d'un fichier (ou répertoire) déjà présent, cela provoque une erreur : le nom de base doit être nouveau dans le répertoire de référence.

```
$ mkdir eclair
$ ls -F
$ cd eclair
$ rmdir ../eclair
$ cd ..
$ rmdir eclair
$ ls -F
```

3.2 Exercice Essayer `rmdir ~` et commenter le résultat obtenu.

4 Manipulation de fichiers

4.1 Copies

4.1 Exercice Que font les commandes :

```
$ cd staroffice6.0
$ cp README.html LICENCE.html ~/tmp
$ cp *.html ~/tmp
$ cd ~/tmp; cp ~/staroffice6.0/* .
```

Y a-t-il une différence du point de vue de `cp` entre les deux premières commandes ?

4.2 Exercice Enchaîner et commenter, en redessinant l'arborescence à chaque étape, la suite de commandes :

```
$ cd tmp
$ cp ../staroffice6.0/README.html README.cp
$ cp README.cp README.html
$ cp README.html ../staroffice6.0
$ cd ../staroffice6.0
$ cp *html ../tmp
```

4.2 Suppressions

4.3 Exercice Essayer la suite de commandes :

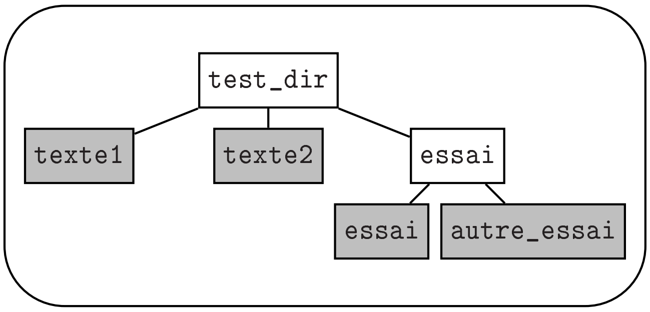
```
$ rm ~/tmp/README.html
$ cd ~/tmp
$ rm * # attention danger!
```

Pour la suppression interactive essayer : `rm -i ≺`*fichier*`≻ ...` (répondre `y` pour confirmer la suppression et `n` pour infirmer).

4.3 Déplacements et renommages

```
$ cd ~/staroffice6.0
$ mv README.html README.HTML
$ ls -F
$ mv README.HTML ../README.html
$ ls -F
$ ls -F ..
$ mv ../README.html .
$ ls -F ..
$ ls -F
$ mv *.html  /tmp
$ ls -F
$ mv  /tmp/*.html .
```

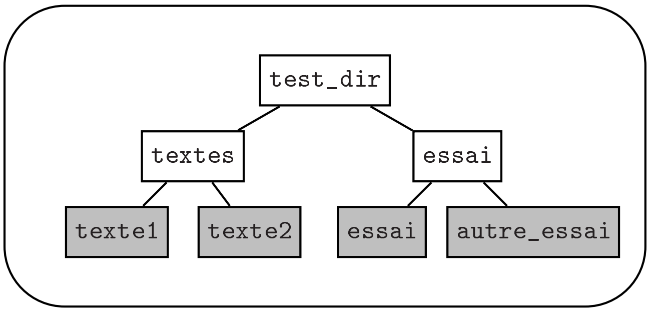
4.4 Exercice Créer à partir de son répertoire d'accueil l'arborescence



Les cases blanches sont des répertoires, tandis que les cases grisées sont des fichiers texte à créer. La marche à suivre pour cete exercice est la suivante :

– créer les répertoires ;
– lancer emacs à partir du répertoire d'accueil ;
– éditer les fichiers et les sauvegarder.

4.5 Exercice Se placer dans le répertoire `test_dir`, et se ramener en deux commandes à :



4.6 Exercice Depuis `test_dir` essayer `cp essai textes` ; commenter le résultat. Que fait dans ce même cas `cp *` ? Essayer avec l'option `-R`. Commenter les erreurs possibles lors de l'exercice précédent :

```
$ mv t* textes
$ cp t* textes
```

4.4 Liens

4.7 Exercice Dans un répertoire vide, essayer le commandes suivantes :

```
$ touch fichier
$ ln fichier hardlink
$ ln -s fichier symblink
$ ls -F
$ rm fichier
$ ls -l
$ more hardlink
$ more symblink
```

5 Protections

5.1 Exercice Donner des exemples de droit d'accès en lecture et écriture dans un répertoire.

5.2 Exercice Essayer `ls -l`; commenter l'affichage.

5.3 Exercice Regarder les différents accès de ses fichiers.

5.4 Exercice Enlever l'accès en lecture pour l'utilisateur et essayer de lire au moyen de `cat`. Vérifier qu'un autre utilisateur peut encore le lire (utiliser le fichier de son voisin). Réveiller **`emacs` (de quelle façon ?)** et charger le fichier protégé en lecture. Stopper emacs et remettre l'accès en lecture. Tenter de lire le contenu du fichier au moyen de `cat`

5.5 Exercice Enlever l'accès en écriture pour l'utilisateur, puis réveiller emacs et tenter de modifier le fichier : emacs a placé ce buffer en mode `read only` repérable par `-%%-` en début de ligne de mode. Ce mode est désactivable au moyen de la commande `toggle-read-only` liée à la clé `C-x C-q`. Désactiver ce mode, modifier le fichier et tenter de sauver. Stopper emacs, redonner l'accès en écriture et recommencer.

5.6 Exercice Essayer ces manipulations sur un fichier d'un autre utilisateur.

5.7 Exercice Expérimenter les modes d'accès d'un répertoire au moyen de `chmod`, `ls`, `cp` et `rm`. Vérifier que l'on peut avoir le droit de détruire un fichier sans avoir le droit de le lire. Dans quel cas peut-on détruire un fichier qui ne nous appartient pas ?

5.8 Exercice Pourquoi `ls -l` ne permet pas de lister le mode d'accès du répertoire courant (rechercher au moyen de `man` la façon de faire).

6 Visite de l'arborescence

6.1 Exercice Présenter les principaux répertoires du système Unix.

- `cd / ; ls -l` (on ignore les liens symboliques)
- observer les droits d'accès
- visiter `/bin` ; commenter `cp`, `mv`, `chmod`, ...
- visiter `/lib` ; commenter `libc.a`, `libm.a`, ...
- rôle de `/tmp`, `/dev`, `/dev/null`
- visiter et commenter `usr`, en particulier `include`, `man` et, `bin` (`gcc, java, javac`).

6.2 Exercice Essayer la commande `find /lib -name \*.a -print`.

6.3 Exercice Utiliser l'option `-type` de `find` pour lister les répertoires au dessous de `/dev`.

6.4 Exercice Trouver des fichiers chez vous modifiés depuis hier.

6.5 Exercice Trouver chez vous les fichiers modifiés il y a plus de deux jours mais moins de 10 jours.