

# SGF et commandes Shell

## 1 Syntaxe de description d'une commande Unix

### Exercice 1

Si la syntaxe de la commande `cut` est décrite par les deux lignes suivantes :

```
cut -c list [file ...]
```

```
cut -f list [-d delim] [-s] [file ...]
```

les appels suivants sont-ils corrects, et s'ils ne le sont pas, expliquez pourquoi.

1. `cut -d : -f 1,7 fichier` Vrai car l'ordre n'importe pas
2. `cut -c 1-16,26-38` vrai car file est
3. `cut -s fich1 fich2` faux car il n'y a
4. `cut -f 2 -c 1,3,5 fichier` -f et -c ne peuvent pas être dans la

### Exercice 2

Considérons une commande `truc` :

- qui possède 3 options : `f`, `c` et `d`. Les options `c` et `d` sont facultatives mais ne peuvent pas être utilisées simultanément. L'option `f` est obligatoire ;
- elle doit être appelée avec une liste non vide de fichiers.

Écrire la syntaxe d'appel de cette commande en utilisant les conventions vues en cours.

```
truc -f [c | d] file [...]
```

## 2 Désignation d'un fichier

### Exercice 3

Dans une désignation, tous les caractères `' / '` ont-ils la même signification ?

Non

### Exercice 4

1. La désignation relative et la désignation absolue d'un fichier sont-elles uniques ? relative non, absolue oui
2. On définit la longueur d'une désignation comme le nombre de répertoires qu'elle contient. Pour désigner un même élément, quelle est, parmi la désignation relative et la désignation absolue, celle qui est la plus courte ? relative si on est plus proche du fichier voulu sinon absolue

### Exercice 5

Si le répertoire courant est : `/users/linfg/l2inf201/REP1`

donner la désignation relative du fichier : `/users/linfg/l2inf201/REP1/f1`  
`./f1`

## 3 Métacaractères du shell

### Exercice 6

1. Décrire le modèle de noms de fichiers suivant : `REP/*.*???` tous les fichiers ayant une extension de 3 caractères dans le dossier REP
2. En utilisant les métacaractères du shell, écrire les modèles suivants :
  - le caractère `a` ou le caractère `b` ou le caractère `c` ou le caractère `d` ou le caractère `e` ; `[a-e]`
  - le caractère `a` suivi d'un chiffre ou du caractère `b` ; `a[0-9b]`
  - une minuscule ou une majuscule ; `[a-zA-Z]`
  - une minuscule suivie d'un chiffre. `[A-Z][0-9]`

👁 `ls`, `echo`, `cat`, `grep`, `wc` (cf. cours).

### 📎 Exercice 7

Nous supposons qu'il n'y a pas de sous-répertoire dans le répertoire courant. Écrire, en une seule commande à chaque fois, les commandes suivantes :

1. Afficher les noms de fichiers non cachés dont le nom contient le caractère `a` `ls *a*`
2. Afficher la liste détaillée des fichiers non cachés dont le nom comporte quatre caractères exactement. `ls -l ????`
3. Afficher le contenu des fichiers non cachés dont le nom comporte au moins cinq caractères. `cat ?????*`
4. Afficher le nombre de lignes que contient chacun des fichiers non cachés dont le nom commence par une majuscule. `wc -l [A-Z]*`

### 📎 Exercice 8

Nous supposons que les répertoires `REP1` et `REP1/SREP1` sont dans le répertoire courant.

1. En utilisant une seule fois la commande `ls`, afficher le contenu du sous-répertoire `REP1` et du sous-sous-répertoire `REP1/SREP1`. `ls REP1 REP1/SREP1`
2. Même question mais en utilisant une seule fois la commande `echo`. `echo REP1/* REP1/SREP1/*`
3. Dans les lignes de commande précédentes, les solutions avec `ls` et `echo` produisent-elles le même résultat ?

`echo` affiche aussi le chemin d'accès

## 4 Redirections

### 📎 Exercice 9

1. Donner trois manières différentes pour afficher les lignes du fichier `truc.txt` qui contiennent au moins un caractère `e`.
2. Qu'est-ce qui change lors de l'exécution ?

1 - <code>cat truc.txt   grep 'e'</code>
2 - <code>grep 'e' truc.txt</code>
3 - <code>grep 'e' &lt; truc.txt</code>

### 📎 Exercice 10

1. Écrire la commande qui enregistre dans le fichier `liste.txt` les noms de tous les fichiers et sous-répertoires du répertoire courant.
2. Écrire la commande permettant d'afficher les lignes des fichiers dont les noms sont dans `liste.txt` et qui contiennent au moins un caractère `e`.