

## TD n°6 - Bash

Le but du TD est d'étudier les concepts complexes des scripts : chaînes, structures de contrôle... sur un script qui est construit progressivement.

### 1) Objectif du script renommext

On veut renommer tous les fichiers portant une certaine extension (ex : .lst) en une autre extension (ex : .txt). On veut donc pouvoir faire `renommext .lst .txt *.lst` : on fournit la première extension en premier paramètre (celle qu'il faut enlever), la deuxième extension en deuxième paramètre (celle qu'il faut mettre à la place) et la liste des fichiers concernés comme paramètres suivants (en nombre variable).

Sur un exemple concret, voici une liste de fichiers :

```
p1.lst    p2.lst    prog.lst  prog.c    fich.lst
```

On tape `renommext .lst .txt p*.lst`

Ça produit ce résultat :

```
p1.txt    p2.txt    prog.txt  prog.c    fich.lst
```

Explications : seuls les fichiers `p...lst` ont été renommés conformément au troisième paramètre du script, `fich.lst` n'était pas concerné. Le script a donc exécuté les commandes suivantes :

```
mv p1.lst p1.txt
mv p2.lst p2.txt
mv prog.lst prog.txt
```

### 2) Analyse et réalisation du script

Le cœur du script lance plusieurs fois la commande `mv` avec les bons paramètres. Notez qu'un seul appel à `mv` ne peut pas faire tout le travail : quand il y a trois fichiers à renommer, il faut appeler trois fois `mv`. Il faut donc :

- obtenir la liste des fichiers à renommer
- pour chaque fichier de cette liste :
- construire les paramètres de `mv` (changement d'extension) et appeler `mv`

#### a) Gestion des extensions pour un seul fichier

Admettons qu'il n'y ait qu'un seul fichier à traiter : `toto.lst`, que l'ancienne extension soit `.lst` et que la nouvelle soit `.txt`, il faut construire la commande : `mv toto.lst toto.txt` c'est-à-dire en généralisant :

```
mv nomsansextension.ancienneextension nomsansextension.nouvelleextension
```

Donc on a comme premier problème d'obtenir le nom sans extension d'un fichier.

- Le cours mentionne la commande `basename`. Comment faut-il l'utiliser pour obtenir à l'écran le nom sans l'extension ? (relire le cours)
- Sauriez-vous l'écrire avec une expression de bash (substitution de variable) permettant d'obtenir la valeur d'une variable tout en la tronquant par une expression régulière ?
- Si on place `toto.lst`, `.lst` et `.txt` dans des variables `$nomfichier`, `$ancext` et `$nouvert` : quelle instruction Bash permet de calculer dans `$nomseul` le nom sans extension du fichier (`toto`) ? Comment fonctionnent les `$(...)` ?
- Comment faut-il écrire le lancement de `mv` avec ces variables ?

**b) Paramètres du script - cas simple**

Admettons toujours qu'il n'y ait qu'un seul fichier à traiter car on a lancé le script par :

```
renommext .lst .txt toto.lst
```

- i. Dans quelles variables se trouvent ces paramètres ?
- ii. Compléter le script pour qu'il les mette dans \$ancext, \$nouvext et \$nomfichier.
- iii. Peut-on vérifier qu'au moins trois paramètres ont été fournis au script ?
- iv. Peut-on vérifier que le nom de fichier est vraiment un nom de fichier qui existe ?

**c) Paramètres du script - cas complet**

On se place maintenant dans le cas d'une liste de fichiers fournie à la suite des extensions. Par exemple :

```
renommext .lst .txt toto.lst titi.lst tutu.lst
```

ou bien :

```
renommext .lst .txt t*.lst
```

- i. Comment le joker \* est-il traité : que voit le script ?
- ii. Dans quelle variable et de quelle façon récupère-t-on cette liste de fichiers ?
- iii. Comment fait-on pour traiter chacun de ces noms de fichiers ? Programmer cet aspect.

**d) Amélioration du script : éviter certaines erreurs**

On a fait l'hypothèse qu'on pouvait renommer les fichiers sans conflit. Pourtant, si on renomme sans précaution, on peut écraser un fichier existant. Exemple : ls montre

```
p1.lst p2.lst p1.txt prog.c fich.lst
```

Si on fait `renommext .lst .txt *.lst`, on écrase `p1.txt` (vérifiez).

- i. Comment faire pour empêcher d'écraser l'ancien `p1.txt` ? Programmer ce qu'il faut pour afficher un message d'erreur dans ce cas.
- ii. On a également fait l'hypothèse qu'on fournit une liste de fichiers portant la bonne extension. Que se passerait-il si on faisait : `renommext .c .txt fich.lst` ?  
Peut-on rajouter quelque chose pour avertir l'utilisateur de l'erreur qu'il commet et pour empêcher que ça arrive ?