

# Système d'exploitation

## TP n° 2

### Script shell

Florent Devin

## 1 De l'utilité de la première ligne

Créez un fichier qui contient le texte suivant :

```
a=" Bonjour"  
b=${ a##*o}  
c=${ a#*o}  
  
echo $a  
echo $b  
echo $c
```

Changer les droits d'exécution du fichier pour que vous puissiez l'exécuter, puis exécutez ce fichier. Une fois ceci fait, rajoutez au début du fichier la ligne suivante :

```
#!/ bin / bash -v
```

Puis réexécutez le fichier. Que constatez vous ?

Faites la même chose avec :

```
#!/ bin / tcsh
```

Puis avec :

```
#!/ bin / bash
```

Cette ligne sert donc à fixer un environnement d'exécution identique pour tous les utilisateurs. Il vaut mieux toujours mettre ce type de ligne, car elle vous assure que le script sera toujours lancé de la même façon, c'est à dire avec le même shell.

## 2 Préambule

### 2.1 Objectif

Nous allons créer un utilitaire capable de déterminer les doublons de fichiers. Pour ce faire, nous allons utiliser plusieurs programmes. Ces programmes sont : `md5sum`, et un programme de calcul de CRC. Le principe de ce programme est très simple. Pour déterminer si un fichier est en double, nous allons calculer le *md5* de tous les fichiers que nous souhaitons analyser. Une fois ce calcul fait, nous extrayons les fichiers qui sont en double, c'est à dire tous les fichiers qui possèdent le même *md5*. Sur chacun de ces fichiers, nous calculons le *CRC*. Si on détecte que les fichiers ont le même *md5*, et le même *CRC*, alors on en déduit que ces fichiers sont identiques. Il faudra alors proposer de soit supprimer les doublons, soit de les déplacer dans un répertoire prévu à cet effet.

### 2.2 Vérification de l'environnement

Tapez la commande :

```
apropos md5
```

Vérifiez que vous avez une fonction capable de calculer la somme MD5. Vous devriez avoir une ligne qui ressemble à :

```
md5sum (1)          - compute and check MD5 message digest
```

Tapez la commande :

```
apropos crc
```

Faites de même pour vérifier que vous avez une commande capable de calculer le CRC.

S'il vous manque une des ces commandes, il va falloir les installer. Pour ce faire, passez super utilisateur<sup>1</sup>. Une fois passé super utilisateur, tapez les commandes :

```
apt-get update
apt-cache search md5 | grep compute
```

Ceci vous indique les paquets qui vous permettent de réaliser les actions demandées. Installez les paquets qui vous manque en fonction du résultat de l'étude précédente. Pour installer un paquet il faut taper une commande qui ressemble à :

```
apt-get install nom_du_paquet
```

---

<sup>1</sup>Pour rappel, la commande qui vous permet de changer d'utilisateur est : `su`

## 3 Script shell

Le script que nous allons écrire deviendra une commande Unix, il faut donc traiter la ligne de commande correctement. Pour ce faire, votre script doit réagir comme suit :

```
.../Tp2_Script_Shell $ doublons -h
Usage : doublon [OPTION] [REPERTOIRE]
doublons : script de détection et de suppression des doublons
Sans REPERTOIRE, applique le script sur le répertoire courant.
-h, --help    affiche cette aide, et quitte
-d            efface les doublons détectés
-o rep        déplace les doublons détectés dans rep
--no-crc     ne calcule pas le CRC pour déterminer les doublons
```

### 3.1 Travail préliminaire

Dans un premier temps, écrire un script qui traite la ligne de commande. C'est à dire un script qui détecte les options passées.

### 3.2 Travail final

Écrire le script dans son ensemble pour qu'il réalise la tâche demandée. Vous ferez attention à ne pas détecter les liens symboliques comme doublons. Par contre, il ne vous est pas demandé de fournir une version optimal du code.