Table des matières

Utiliser conjointement SED & CUT pour filtrer

sed et cut permettent de modifier ou de supprimer une partie d'une chaîne de caractères, par exemple pour remplacer un caractère par un autre dans un fichier, ou encore supprimer des chaînes de caractères inutiles. Ce sont des outils très puissants.

Les possibilités de ces commandes en terme de syntaxe étant tellement vastes que nous n'aborderons que la partie émergée de l'iceberg (cette page étant bien évidemment perfectible dans le temps)

Utilisation

Pour utiliser sed ou cut, vous devez lui fournir une chaîne à traiter.
Cette chaîne peut provenir :

d'un fichier
d'une variable
En règle générale la syntaxe est de la forme

```
"s/[occurrence_cherchée]/[occurrence_de_substitution]/[comportement]"
```

sed - cut et les fichiers

Pour pouvoir traiter un fichier, il est nécessaire de lier le fichier à sed. Nous utiliserons la commande grep sous la forme : grep occurrence /fichier/a/parcourir | sed ...

```
grep occurrence /fichier/a/parcourir | cut ... \\
```

ou encore même conjointement grep occurrence /fichier/a/parcourir | cut ... | sed ...

sed - cut et les variables

Pour nos exemples nous allons définir une variable à l'aide de la commande export

export chaine="ceci est une chaine de caractères"
Ainsi nous définissons la valeur de la variable $chaine. Nous pouvons à présent appeler cette variable au besoin.

```
echo $chaine | sed -e "s/ /_/g"
```

Donnera en réponse

```
ceci_est_une_chaine_de_caractères
```

Remplacer les espaces par des caractères est une des étapes importantes visant à substituer les caractères pouvant être mal interprétés par la suite.
mplayer mon fichier.avi ne fonctionne pas tandis que mplayer mon_fichier.avi fonctionnera Le g positionné après l'occurrence de substitution (“s/ /_/g”) indique à sed de se comporter de façon globale (“récursive”)

Exemple:

Nous voulons récupérer l'uid ainsi que le gid d'un utilisateur en cours. Sous linux, l'utilisateur courant est défini dans la variable $USER
La preuve en est :

```
echo $USER
```

retourne l'utilisateur en cours
l'uid et gid de l'utilisateur est stocké dans le fichier /etc/passwd.
Dans un premier temps récupérons la ligne concernant notre utilisateur (ici Florent).

```
grep $USER /etc/passwd
```

parcourir le fichier /etc/passwd et retourner la ligne concernant le nom de l'utilisateur en cours
Donnera en réponse

```
florent:x:1000:1000::/home/florent:/bin/bash
```

vous me direz : parfait ! Et bien nous avons notre uid et gid. Certes mais le résultat est difficilement exploitable vous en conviendrez.
Nous allons donc séparer de façon distincte les deux valeurs.
Il nous faut pour cela analyser le résultat de la commande précédente pour définir des règles de traitement :

```
uid et gid sont sous la forme de chiffres
```

les informations sont séparées par :
la liste des paths est introduite par ::
ces différentes remarques vont nous permettre de fixer des délimiteurs, isolons donc la chaîne précédant les :: du reste de la chaîne

```
grep $USER /etc/passwd | sed "s/::/%/" |cut -d'%' -f1
```

Parcourir le fichier /etc/passwd et retourner la ligne concernant le nom de l'utilisateur en cours | remplacer (s) les :: par % dans le résultat | dans le résultat, supprimer (-d) l'occurrence '%' et tout ce qui la suit la première occurrence vérifiant (-f1) donnera

```
florent:x:1000:1000
```

Le principe est donc maintenant posé. Effectuons une dernière action afin de mettre en forme le résultat :

```
grep $USER /etc/passwd | sed "s/::/%/g" |cut -d'%' -f1 |cut -d'x' -f2 |sed -
e "s/:\([0-9][0-9]*\)/UID=\1\n/"  -e "s/:\([0-9][0-9]*\)/GID=\1/"
```

Donnera

```
UID=1000
GID=1000
```

Décortiquons les différentes étapes du traitement :

```
Commande    Sortie
origine grep $USER /etc/passwd
florent:x:1000:1000::/home/florent:/bin/bash
Etape 1 sed "s/::/%/"      florent:x:1000:1000%/home/florent:/bin/bash
Etape 2 cut -d'%' -f1      florent:x:1000:1000
Etape 3 cut -d'x' -f2      :1000:1000
Etape 4 sed -e "s/:\([0-9][0-9][0-9][0-9]\)/UID=\1\n/" -e
"s/:\([0-9][0-9][0-9][0-9]\)/GID=\1/"     UID=1000
GID=1000
```

plus simple :

```
sed -n -e "/`echo
$USER`/s/[^:]*:[^:]*:\([^:]*\):\([^:]*\):.*/UID=\1\nGID=\2/p" /etc/passwd
```

Voila les étapes du traitement d'une chaîne de caractères avec quelques détails à ne pas oublier.

L'ordre de traitement est crucial
Le délimiteur utilisé par cut ne peut excéder 1 caractère d'où l'étape 1 visant à remplacer deux caractères consécutifs par un caractère unique remarquable utilisé dans l'étape 2
L'étape 4 introduit quelques syntaxes intéressantes :

- sed -e “[règle 1]” -e “[règle 2]”
- sed “s/\(occurrence\)/%\1%/” qui retourne %occurrence%
- sed “s/\(occurrence\)/\1\n/” qui insère un retour à la ligne à la fin de l'occurrence