

SED: les bases

Comment modifier du texte en masse?

- ⇒ [SED c'est quoi ?](#)
 - ⇒ [References](#)
 - ⇒ [Cas pratique 1](#)
 - ⇒ [Remplacer des occurrences](#)
 - ⇒ [Commande](#)
 - ⇒ [Utilisation](#)
 - ⇒ [Cas pratique 2](#)
 - ⇒ [Commande *grep*](#)
 - ⇒ [Ca marche pas ...](#)

1) SED c'est quoi ?

Voici la définition de [wikipedia](#)

sed (abréviation de Stream Editor, « éditeur de flux ») est, comme awk, un programme informatique permettant d'appliquer différentes transformations prédéfinies à un flux séquentiel de données textuelles. sed lit des données d'entrée ligne par ligne, modifie chaque ligne selon des règles spécifiées dans un langage propre (appelé « script sed »), puis retourne le contenu du fichier (par défaut). Bien qu'originellement écrit pour Unix, par Lee E. McMahon (en) en 1973/1974 (Bell Labs), sed est maintenant disponible sur pratiquement tous les systèmes d'exploitation disposant d'une interface en ligne de commande.

Il faut comprendre que *SED* va permettre de modifier du texte au sein d'un fichier, ou d'un ensemble de fichiers.

SED dispose de nombreuses commandes et options, c'est un outil très puissant, et il est facile de tomber dans des pièges, on peut très rapidement se retrouver avec un ensemble de fichiers quasi inutilisable.

1.1) References

[Le grimoyre Sed - An Introduction and Tutorial by Bruce Barnett](#)

[Gnu.org sed, a stream editor](#)

1.2) Cas pratique 1

Nous allons voir 2 cas très simple, comme beaucoup on découvre *SED* lorsque l'on souhaite *substituer* un mot par un autre dans un fichier très grand (mais pas toujours) ou dans un ensemble de fichiers, voir toute une arborescence.

1.2.1) Remplacer des occurrences

Nous souhaitons remplacer toutes les occurrences trouvées dans un fichier par une autre, comme lors du passage de debian 8 *jessie* vers debian 9 *stretch*, dans le cas présent le fichier `/etc/apt/sources.list`

1.2.1.1) Commande

```
sed -i 's|jessie|stretch|' /etc/apt/sources.list
```

Le fichier `sources.list` contient la liste des sources pour les mise à jours. Dans le cas présent on souhaite modifier la source depuis *jessie* vers *stretch*.

```
deb http://deb.debian.org/debian jessie main
deb http://deb.debian.org/debian jessie-updates main
deb http://security.debian.org/ jessie/updates main
```

Pour l'occasion je crée le fichier `sed_test01` et applique la modification.

↪ `sed -i 's|jessie|stretch|' test_sed01` si il n'y a pas de message d'erreur c'est que ça a fonctionner.

Je fais l'opération inverse.

```
└─ ◎ ▶ ~ ▶                               ◀ ✕ ◀ ◎ 11:12 🗓 03.08.18 ◀ 100%
└─
└─ sed -i 's|stretch|jessie|' test_sed01
└─ ◎ ▶ ~ ▶                               ◀ ✕ ◀ ◎ 11:12 🗓 03.08.18 ◀ 100%
└─
└─ cat test_sed01
deb http://deb.debian.org/debian jessie main
deb http://deb.debian.org/debian jessie-updates main
deb http://security.debian.org/ jessie/updates main
└─ ◎ ▶ ~ ▶                               ◀ ✕ ◀ ◎ 11:12 🗓 03.08.18 ◀ 100%
└─
└─ sed 's|jessie|stretch|' test_sed01
deb http://deb.debian.org/debian stretch main
deb http://deb.debian.org/debian stretch-updates main
deb http://security.debian.org/ stretch/updates main
```

L'exemple ici se veut très simple , on remplace un simple mot par un autre dans un seul fichier.

On peut également effacer une occurrence (option `-d`), comme dans le post [ssh_partie2](#): `* sed -i '/clé exemple$/d' /root/.ssh/authorized_keys`

1.3) Utilisation

Nous voyions une première option `sed -i` , lorsqu'elle est invoqué `sed` nous montre le résultat.

Ensuite nous avons `'s|jessie|stretch|'` entre '' nous devons indiquer la commande `s` pour *substitute* , puis `|` qui nous sert de séparateur.

Je vous recommande très fortement d'utiliser `|`, ou `,`, ou bien encore `#` comme séparateur, en lieu et place de `\` qui est souvent proposé dans un premier temps comme exemple.

Pourquoi ? : * lorsque l'on souhaite substituer une adresse de cette forme `http://mon_site.com` il faudra échapper les caractères `/` avec `\`, et on a vite fait de se prendre les pieds dans le tapis, si nous avons des chemins plus long.. * `sed -i 's/http:\\\\mon_site.com/https:\\\\un_autre_site.com'` * Ca devient vite assez illisible.

1.4) Cas pratique 2

Cela fait suite au [post rsync les bases](#)

```
rsync -avP ~/site_ordinatous/_site/ ~/dev_web/dotfiles
```

Comment faire maintenant pour substituer l'adresse d'un site pour une autre adresse dans toute une arborescence ?

Nous allons faire appel à `grep`, celui-ci va rechercher toutes les *occurrences* de manière *réursive*, en lui demandant d'en faire une liste.

1.4.1) Commande grep

```
grep -e "http://localhost:4000" -r ~/dev_web/dotfiles -l
```

- ⇒ option `-e` indique qu'il s'agit d'une occurrence et non un fichier.
- ⇒ entre guillemets "" la dite occurrence.
- ⇒ option `-r ~/dev_web/dotfiles` pour la récursion dans l'arborescence.
- ⇒ option `-l` pour lister.

```
grep -e "http://localhost:4000" -r ~/dev_web/dotfiles -l
/home/ordinatous/dev_web/dotfiles/npm/index.html
/home/ordinatous/dev_web/dotfiles/documentation/2018-04-10-
cyber_defense_Me.html
/home/ordinatous/dev_web/dotfiles/documentation/2018-04-10-cyber_defense_guide-
tome1.html
/home/ordinatous/dev_web/dotfiles/documentation/2018-04-10-commandes_linux.html
/home/ordinatous/dev_web/dotfiles/documentation/2018-04-10-
cyber_defense_guide_blogger.html
/home/ordinatous/dev_web/dotfiles/documentation/2018-04-10-
apprendre_python3_5.html
/home/ordinatous/dev_web/dotfiles/about.html
/home/ordinatous/dev_web/dotfiles/terms/index.html
/home/ordinatous/dev_web/dotfiles/404.html
sortie tronquée
```

Jusque là c'est cool, le répertoire est synchronisé à un autre endroit du disque (cf [post rsync les bases](#)), les fichiers contenant l'occurrence sont listés (cf `grep`), il nous faut maintenant que `sed` trouve les occurrences qui nous intéressent, puis les remplace, pour ça on va *piquer* avec `|`, et si on veut écrire la commande sur 2 lignes pour plus de clareté, on termine la première commande par `\`.

```
grep -e "http://localhost:4000" -r ~/dev_web/dotfiles -l | \
sed -i -e 's;http://localhost:4000;https://site.ordinatous.com;g'
```

1.4.1.1) Ça marche pas ...

Eh non, ça ne marche pas ...

```
grep -e "http://localhost:4000" -r ~/dev_web/dotfiles -l | \
sed -i -e 's;http://localhost:4000;https://site.ordinatous.com;g'
sed: pas de fichier d'entrée
```

On a bien vu que chaque commande fonctionnaient séparément les unes des autres, et c'est justement le "problème", `sed` ne voit pas les arguments précédents, il manque un détail: `* xargs`

```
grep -e "http://localhost:4000" -r ~/dev_web/dotfiles -l | \
xargs sed -i -e 's;http://localhost:4000;https://site.ordinatous.com;g'
```

Pour ce simplifier la vie, on peut mettre toutes nos commandes dans un fichier que l'on rendra *executable*, c'est ce que l'on appelle un script, *script*, par contre ce n'est pas très optimisé, ce *script* fera l'objet d'un post.

SED est capable de faire bien plus, vraiment beaucoup plus: * ajouter un mot en bout de ligne, ou en début * modifier et/ou ajouter dans une ligne ou série de ligne particulière * mettre des majuscules à toutes les lignes commençant par une minuscule

Voilà, merci.