

Table des matières

Manipuler les chaînes de caractères avec SED	2
<i>Man Page</i>	2
<i>Ajouter du texte avant un résultat</i>	7
Exemple	7
avec SED	7



Manipuler les chaînes de caractères avec SED

Man Page

I. Syntaxe d'utilisation

La commande sed contient une ou plusieurs commandes d'éditations et éventuellement des options.

```
sed [OPTIONS] 'commande d'édition' [fichier...]
```

Le filtre Sed met à dispositions diverses options et commandes d'éditations. Nous verrons dans un premier temps (dans le sous chapitre I.a.) quelles sont les différentes options disponibles, puis dans un deuxième temps (dans le sous chapitre I.b.), nous aborderons les commandes d'éditations.

I.a. Options :

Je ne vous parlerai ici que des options sed les plus couramment utilisées. Vous pourrez éventuellement taper la commande man sed pour avoir une liste d'options plus complète.

-e commande

L'option -e est très utile car elle permet d'utiliser plusieurs commandes d'édition à la suite.

Exemple : `sed -e 'commande' -e 'commande'`

-f Fichier_Script

Cette option permet d'ajouter des commandes d'édition se trouvant dans un fichier script. Faire un fichier script est très simple, il suffit de remplir ce dernier avec des commandes d'éditations (chaque ligne doit comporter une commande d'édition différente).

-i[SUFFIXE]

Par défaut sed ne modifie pas directement le fichier qui est fournit en paramètre mais affiche le résultat du traitement sur la sortie standard (stdout). Grâce à cette option sed modifiera directement le fichier fournit en paramètre. On éventuellement mettre un suffixe pour effectuer une sauvegarde du fichier fournit (Exemple de suffixe: -i.bak)

I.b. Commandes d'éditations sed:

Pour éditer le texte sed met à disposition plusieurs commandes d'éditions permettant de spécifier les règles de modification souhaitées. Ces dernières ont une syntaxe d'utilisation particulière que je vais tenter de vous décrire.

La syntaxe d'utilisation des commandes d'éditions sed est la suivante :

```
[adresse[,adresse]]commande d'édition[arguments]
```

Comme vous pouvez l'observer, une commande d'édition sed peut être préfixé par une ou deux adresses.

Les adresses :

Les adresses correspondent à des lignes du texte. Il est possible de spécifier les numéros de lignes, par exemple les adresses 5,7 indiqueront à la commande sed d'agir sur les lignes 5 à 7. On peut également utiliser des expressions régulières, par exemple l'adresse /motif/ indiquera à sed de traiter les lignes correspondantes à l'expression régulière motif. Chaque ligne contenant la chaîne de caractères motif sera alors traitée. On peut de la même façon utiliser deux expressions régulières, par exemple /motif1/,/motif2/ correspond à un domaine de lignes compris entre motif1 et motif2. Nous verrons plus bas comment utiliser les expressions régulières.

Maintenant nous allons aborder les commandes d'éditions sed disponibles pour éditer le texte. Ces dernières peuvent supporter parfois qu'une seule adresse maximale. D'autres peuvent en supporter deux.

Les commandes d'éditions sed pouvant supporter une adresse :

```
[adresse]i
```

Cette commande d'édition sed permet d'insérer du texte avant la ligne correspondante à l'adresse. Si aucune adresse n'est spécifiée le texte sera inséré au début.

```
[adresse]a Ajoute du texte après la ligne.
```

```
q Affiche les n premières lignes.
```

II. Insérer et ajouter du texte

Vous pouvez utiliser la commande i pour insérer du texte avant une ligne et a pour ajouter du texte après une ligne.

```
Ajouter Bonjour monsieur après la ligne 5  
sed '5a\Bonjour monsieur\' fichier.txt
```

```
Insérer Bonjour monsieur avant la ligne 5  
sed '5i\Bonjour monsieur\' fichier.txt
```

III. Substitution

Sed permet de rechercher et remplacer des mots ou des parties de texte, on appelle cela: substituer des chaînes de caractères.

Par exemple pour substituer une occurrence par une autre:

```
Remplacer romain par julien
sed -e 's/romain/julien/' fichier.txt
```

III.a. Utiliser l'option global g

Sed traite le flux ligne par ligne. Attention, sans l'option g sed ne traitera que la première occurrence trouvée sur la ligne en cours de traitement, c'est à dire que si sur une même ligne, la chaîne de caractères recherchée apparaît plusieurs fois, seulement la première occurrence sera traitée. L'option g permet de traiter toutes les occurrences.

Utilisation de l'option g (global)

```
sed -e 's/toto/titi/g' fichier.txt
```

III.b. Utiliser l'option de négation !

Afin d'inverser la fonction sed utilisée, il est possible d'utiliser l'opérateur ! dit de négation.

Voici un exemple:

```
Remplacer tonton par tati sauf à la ligne 5
sed '5!s/tonton/tati/g' fichier.txt
```

III.c. Agir uniquement sur certaines lignes du fichier

Il est possible de dire à sed de ne traiter qu'une partie du fichier texte, pour cela on utilise une plage d'adresse correspondant aux lignes qu'on souhaite traiter.

Voici ci dessous un exemple:

```
Remplacer tonton par tati uniquement entre les lignes 5 et 12
sed '5,12s/tonton/tati/g' fichier.txt
```

III.d. Sélectionner les occurrences

Si jamais il y a plusieurs occurrences d'une même chaîne sur une même ligne, il est possible d'indiquer à sed laquelle on souhaite traiter.

Exemple:

```
Remplacer seulement la deuxième occurrence du mot téléphone par le mot fax
sed 's/telephone/fax/2'
```

IV. Afficher du texte

L'option p qui a pour signification print permet de sélectionner et d'afficher du texte.

Afficher les lignes 18 à 25

```
sed -n '18,25p' fichier.txt
```

Il est possible avec sed de sélectionner un paragraphe.

Afficher le paragraphe qui commence par la ligne contenant la chaîne de caractère prendre la voiture et qui termine par la ligne contenant la chaîne arriver le soir

```
sed -n '/prendre la voiture/,/arriver le soir/p' monfichier
```

V. Effacer du texte

Sed permet grâce à la fonction d (delete) d'effacer des chaînes de caractères.

Par exemple, si vous souhaitez effacer les lignes 18 à 25 d'un fichier, vous pouvez taper la commande suivante:

```
sed '18,25d' fichier.txt
```

Si maintenant, vous souhaitez effacer tout sauf les trois premières lignes d'un fichier, vous pouvez taper la commande suivante:

```
sed '1,3!d' fichier.txt
```

VI. Les méta-caractères

Les métacaractères sont des caractères typographiques spéciaux qui permettent de créer des expressions rationnelles; ces dernières sont utiles pour rechercher des chaînes de caractères répondant à certain critères.

Metacaractère	Description
---------------	-------------

.	Désigne un seul caractère quelconque excepté le caractère '\n' (nouvelle ligne).
---	--

Par exemple:

tot. correspond à tota, totb, totc, totd, tote etc...

^ Indique le début de la ligne

Par exemple:

^root identifie une ligne commençant par root

\$ Indique la fin d'une ligne

Par exemple:

mécanisme\$ identifie une ligne finissant par mécanisme

[] Correspond à n'importe quel caractères cités entre les crochets

Exemple 1 :

[tT]toto correspond aux chaînes de caractères toto et Toto

Exemple 2 :

[a-z] correspond à l'intervalle de caractères de a jusqu'à z

Exemple 3 :

[4-9] correspondant à l'intervalle de caractères de 4 à 9 c'est à dire 4,5,6,7,8,9

Exemple 4 :

l[4-9]0 correspondant à 140,150,160,170,180,190

VI.a. Exemples d'utilisation des métacaractères avec la commande Sed:

VI.a.1. Exemple d'utilisation du métacaractère .

Remplacer ce qui commence par tot et qui fait 4 caractères par toto
sed -e "s/tot./toto/g"

Le métacaractère . représente un caractère quelconque.

VI.a.2. Exemple d'utilisation du metacaractère ^

Effacer les lignes débutant par ^root

```
sed -e "/^root/d"
```

Modifier les lignes qui débutent par root en substituant ce dernier par user

```
sed -e "s/^root/user/"
```

VI.a.3. Exemple d'utilisation du metacaractère \$

Supprimer les lignes finissant par la chaîne de caractères toto

```
sed "/toto$/d"
```

Supprimer les lignes vides

```
sed -e "/^$/d"
```

Modifier les lignes terminant par ; par le caractère .

```
sed -e "s$/;/\."/
```

VI.a.4. Exemple d'utilisation des metacaractères []

Remplacer les chaînes qui commencent par toto et qui finissent par un chiffre entre 0 et 9 par la chaîne tata

```
sed -e "s/toto[0-9]/tata/g"
```

Supprimer les caractères a, b ou c

```
sed -e "s/[abc]//g"
```

VII. Les quantificateurs

Les quantificateurs se place après un caractères et permettent d'exprimer une valeur de répétition ou de nullité.

Quantificateurs Description

* Aucun ou plusieurs caractères qui le précède

Exemple :

toto* correspond à tot, toto, totoo, totooo etc...

Remarque:

.*Correspond à zéro ou plusieurs caractères quelconques

? Zéro ou une occurrence du caractère qui le précède

Exemple :

toto? correspond à tot ou toto

+ Un ou plusieurs caractères qui le précède

Exemple :

toto+ correspond à toto, toto, totoo, totooo etc...

VII.a. Exemples d'utilisation des quantificateurs avec Sed:

VII.a.1. Exemple d'utilisation du quantificateur *

Remplacer tout ce qui commence par tot et qui est suivi du caractère o zéro ou plusieurs fois (c'est à dire: tot, toto, totoo, totooo, totoooo...) par la chaîne baba

```
sed -e "s/toto*/toto/g"
```

Supprimer les chaînes de caractères qui commencent par toto, qui contiennent n'importe quels caractères et finissent par baba

```
sed -e "/^toto.*baba$/d"
```

VII.a.2. Exemple d'utilisation du quantificateur ?

Remplacer tat ou tati par toto

```
sed -e "s/tati?/toto/g"
```

VII.a.2. Exemple d'utilisation du quantificateur +

Remplacer ce qui correspond à tata, tataa, tataaaa etc... par le mot toto

```
sed -e "s/tata+/toto/g"
```

Ajouter du texte avant un résultat

Exemple

```
find /mnt/Mon-NAS/Doc/ -name "*.doc" -type f -exec du -ch {} + | grep total$
```

```
120G    total
```

avec SED

```
find /mnt/Mon-NAS/Doc/ -name "*.doc" -type f -exec du -ch {} + | grep total$  
| sed 's/^/Mes Doc: /'
```

```
Mes Doc: 120G    total
```

— [sylvain](#) 2019/06/15 16:13

From:

<https://wiki.mazinger.fr/wiki/> - **My Personal Wiki**

Permanent link:

<https://wiki.mazinger.fr/wiki/doku.php?id=linux:commande:shell:sed>

Last update: **2024/05/31 13:04**

