La commande find

Saïd El Mamouni – Janvier 2019

Le commande find

Les commandes de recherche

On peut combiner les critères avec des opérateurs logiques :

critère1 critère2 ou critère1 -a critère2	correspond au **ET** logique
!critère	correspond au **NON** logique
\(critère1 -o critère2 \)	correspond au **OU** logique, La commande find doit être utilisée avec l'option **-print** sous certaines versions de shell. Sans l'utilisation de cette option, même en cas de réussite dans la recherche, find n'affiche rien à la sortie standard (l'écran, plus précisément le shell).

La commande **find** est récursive, c'est-à-dire où que vous tapiez, elle va aller scruter dans le répertoires, et les sous-répertoires qu'il contient, et ainsi de suite**.**

Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

1. Recherche par nom de fichier

Pour chercher un fichier dont le nom contient la chaîne de caractères toto à partir du répertoire /usr, vous devez tapez :

```
$find /usr -name toto [-print]
```

En cas de réussite, si le(s) fichier(s) existe(nt), vous aurez comme sortie :

/usr/toto

En cas d'échec, vous n'avez rien.

Pour rechercher tous les fichiers se terminant par .c dans le répertoire /usr et ses sous-répertoires, vous taperez :

```
$find /usr -name " *.c "
```


Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

2. Recherche suivant la date de dernière modification

Pour connaître les derniers fichiers modifiés dans les 3 derniers jours dans toute l'arborescence (/),

vous devez taper :

```
$find / -mtime 3
```

3. Recherche suivant la taille

Pour connaître dans toute l'arborescence, les fichiers dont la taille dépasse 1Mo (2000 blocs de 512Ko), vous devez taper :

```
$find / -size 2000
```


Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

Recherche combinée

Vous pouvez chercher dans toute l'arborescence, les fichiers ordinaires appartenant à paul, dont la permission est fixée à 755, on obtient :

```
$find / -type f -user paul -perm 755
```

5. Redirection des messages d'erreur

Vous vous rendrez compte assez rapidement qu'en tant que simple utilisateur, vous n'avez pas forcément le droit d'accès à un certain nombre de répertoires, par conséquent, la commande find peut générer beaucoup de messages d'erreur (du genre permission denied), qui pourraient noyer l'information utile. Pour éviter ceci, vous pouvez rediriger les messages d'erreur dans un fichier poubelle (comme /dev/null), les messages d'erreur sont alors perdus (rien ne vous empêche de les sauvegarder dans un fichier, mais ça n'a aucune utilité avec la commande find).

Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

6. Recherche en utilisant les opérateurs logiques

Si vous voulez connaître les fichiers n'appartenant pas à l'utilisateur olivier, vous taperez :

```
$find . ! -user olivier
```

! -user olivier : est la négation de -user olivier, c'est à dire c'est tous les utilisateurs sauf olivier.

Recherche des fichiers qui ont pour nom a.out et des fichiers se terminant par .c. On tape :

```
$find . \( -name a.out -o -name " *.c " \)
```

On recherche donc les fichiers dont le nom est a.out ou les fichiers se terminant par *.c, une condition ou l'autre.

Recherche des fichiers qui obéissent à la fois à la condition ''a pour nom core'' et à la condition ''a une taille supérieure à 1Mo''.

```
$find . \( -name core -a size +2000 \)
```


Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

7. Les commandes en option

L'option **-print** est une commande que l'on passe à find pour afficher les résultats à la sortie standard. En dehors de **print**, on dispose de l'option **-exec**.

find couplé avec **exec** permet d'exécuter une commande sur les fichiers trouvés d'après les critères de recherche fixés.
Cette option attend comme argument une commande, celle ci doit être suivi de **{} \ ;**.

Exemple : recherche des fichiers ayant pour nom core, suivi de l'effacement de ces fichiers.

```
$find . -name core -exec rm {} \ ;
```

Tous les fichiers ayant pour nom core seront détruits, pour avoir une demande de confirmation avant l'exécution de rm, vous pouvez taper :

```
$find . -name core -ok rm {} \ ;
```


Said.elmamouni@univ-nantes.fr

Le commande find

Les commandes de recherche

8. Autres subtilités

Une fonction intéressante de find est de pouvoir être utilisée avec d'autres commandes UNIX.

Par exemple:

```
$find . -type f -print | xargs grep toto
```

En tapant cette commande vous allez rechercher dans le répertoire courant tous les fichiers normaux (sans les répertoires, fichiers spéciaux), et rechercher dans ces fichiers tous ceux contenant la chaîne toto.

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

Grep

La commande grep permet de rechercher une chaîne de caractères dans un fichier (ou des fichiers).
‘’chercher quelque chose dans un fichier ‘’

grep "motif" FICHIER

Quelques options :

grep -l motif fichier		affiche le nom des fichiers qui contiennent la chaîne
grep -n motif fichier		chaque ligne contenant la chaîne est numérotée
grep -i motif fichier		sans tenir compte de la casse
grep -c motif fichier		en comptant les occurrences
grep -v motif fichier		inverse la recherche, en excluant le "motif"
grep -x motif fichier		ligne correspondant exactement à la chaîne
grep -r motif fichier	-r, --recursive	Lire récursivement tous les fichiers à l’intérieur de chaque répertoire, en ne suivant les liens symbolique que s’ils sont indiqué dans la ligne de commande

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

Exercice

Créez un répertoire **SI5** dans votre home.
Créez à l'aide de vim, 5 fichiers textes avec les contenus ci-dessous

Fichier	contenu
Grep1.txt	Ma Commande Ma commande Mes commandes Notes
Grep2.txt	Commande commande
Test1.txt	Commande commande
Test.prn	Commande commande
Test4.txt	Commandes commande

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

- Recherchez le mot Commande dans le fichier Grep1.txt

- lister les occurrences du mot Commande dans les fichiers .txt du répertoire SI5

- lister les occurrences du mot commande dans les fichiers du dossier SI5

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

- Recherchez le mot Commande dans le fichier Grep1.txt

```
$ grep Commande SI5/Grep1.txt

Ma Commande
```

- lister les occurrences du mot Commande dans les fichiers .txt du répertoire SI5

- lister les occurrences du mot commande dans les fichiers du dossier SI5

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

- Recherchez le mot Commande dans le fichier Grep1.txt

```
$ grep Commande SI5/Grep1.txt

Ma Commande
```

- lister les occurrences du mot Commande dans les fichiers .txt du répertoire SI5

```
$ grep -x "Commande" SI5/*.txt

SI5/Grep2.txt:Commande
SI5/Test1.txt:Commande
```

- lister les occurrences du mot commande dans les fichiers du dossier SI5

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

- Recherchez le mot Commande dans le fichier Grep1.txt

```
$ grep Commande SI5/Grep1.txt

Ma Commande
```

- lister les occurrences du mot Commande dans les fichiers .txt du répertoire SI5

```
$ grep -x "Commande" SI5/*.txt

SI5/Grep2.txt:Commande
SI5/Test1.txt:Commande
```

- lister les occurrences du mot commande dans les fichiers du dossier SI5

```
$ grep -or "commande" SI5/

SI5/Grep2.txt:commande
SI5/Test4.txt:commande
SI5/Grep1.txt:commande
SI5/Grep1.txt:commande
SI5/Test.prn:commande
SI5/Test1.txt:commande
```


Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

Affiche toutes les lignes de fichier qui contiennent « Paris » :

grep "Paris" fichier

Affiche le nombre de lignes de fichier qui contiennent « Paris » :

grep -c "Paris"' fichier

Affiche les lignes de fichier qui commencent par « Paris » (le métacaractère « ^ » signifie qui commence par) :

grep "^Paris" fichier

Affiche la liste des fichiers du répertoire rep1 qui contiennent « Paris »

grep "Paris" rep1/*

Said.elmamouni@univ-nantes.fr

La commande grep

Les commandes de recherche

find . -name "*.mp4" -exec ls -lh "{}" \;

find . -type f -iname "*.mp4" -exec du -b -c {} + | tail -1

Said.elmamouni@univ-nantes.fr

La commande grep

Il existe de nombreux dérivés de grep. Parmi eux :

agrep, pour approximate grep, c'est-à-dire grep approximatif, qui facilite la recherche de chaînes approchées ;

fgrep, pour des recherches dans un ou des fichiers ;

egrep, pour les recherches nécessitant une syntaxe d'expressions rationnelles plus sophistiquée ;

rgrep, pour une recherche dans tous les fichiers d'un répertoire ;

zgrep, pour une recherche dans un ou des fichiers compressés ;

Tcgrep, qui est une réécriture de grep et qui utilise les expressions rationnelles de Perl.

La plupart de ces variantes de grep ont été portées sur de nombreux systèmes d'exploitation. On parle alors d'implémentation moderne de grep.

De nombreuses autres commandes contiennent le mot « grep ». Par exemple, l'utilitaire pgrep affiche le numéro des processus dont le nom correspond à l'expression rationnelle.

Le commande find

Les commandes de recherche

La commande locate

La commande locate a la même mission que find. Pourtant vous verrez qu'en utilisant la commande locate, le fichier sera trouvé beaucoup plus rapidement. Pourquoi ? Parce que locate ne va pas chercher le fichier dans toute l'arborescence des répertoires mais va localiser la position du fichier dans une base de données qui contient la liste des fichiers existants.
Cette base de données est en général automatiquement générée une fois par jour par le système grâce à une commande appelée updatedb.
Sur un système Linux Redhat, cette base de donnée se trouve dans le répertoire /usr/lib et se nomme locatedb.

La syntaxe est donc simple:

```
$locate nom_du_fichier
```

Bien que la commande locate soit très intéressante, elle ne possède pas la puissance des options de find. De plus, si vous créez des fichiers pendant la journée et que vous les recherchez avec la commande locate, il n'est pas sûr que la base de donnée ait été remise à jour. Bref, locate est un complément de find.

La commande sed

Les commandes de recherche

4. La commande sed

sed est éditeur ligne non interactif, il lit les lignes d'un fichier une à une (ou provenant de l'entrée standard) leur applique un certain nombre de commandes d'édition et renvoie les lignes résultantes sur la sortie standard.
Il ne modifie pas le fichier traité, il écrit tout sur la sortie standard.

sed est une évolution de l'éditeur ed lui même précurseur de vi, la syntaxe n'est franchement pas très conviviale, mais il permet de réaliser des commandes complexes sur des gros fichiers.

La syntaxe de sed est la suivante:

```
$sed -e 'programme sed' fichier-a-traiter
```

ou

```
$sed -f fichier-programme fichier-a-traiter
```


Said.elmamouni@univ-nantes.fr

Les commandes de recherche

4. La commande sed

Vous disposez de l'option **-n** qui supprime la sortie standard par défaut, sed va écrire uniquement les lignes concernées par le traitement (sinon il écrit tout même les lignes non traitées).

L'option **-e** n'est pas nécessaire quand vous avez une seule fonction d'édition.

pour plus de détails faites un man sed et/ou man ed.

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

2. La fonction de substitution s

La fonction de substitution **s** permet de changer la première ou toutes les occurrences d'une chaîne par une autre.

sed "s/toto/TOTO/" fichier

→ va changer la première occurence de la chaîne toto par TOTO (la première chaîne toto rencontrée dans le texte uniquement)

sed "s/toto/TOTO/3" fichier

→ va changer la troisième occurence de la chaîne toto par TOTO (la troisième chaîne toto rencontrée dans le texte uniquement)

sed "s/toto/TOTO/g" fichier

→ va changer toutes les occurences de la chaîne toto par TOTO (toutes les chaînes toto rencontrées sont changées)

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

2. La fonction de substitution s

sed "s/toto/TOTO/p" fichier

→ en cas de remplacement la ligne concernée est affichée sur la sortie standard (uniquement en cas de substitution)

sed "s/toto/TOTO/w resultat" fichier

→ en cas de substitution la ligne en entrée est inscrite dans un fichier résultat

La fonction de substitution peut évidemment être utilisée avec une expression régulière.

sed -e "s/[Ff]raise/FRAISE/g" fichier substitue toutes les chaînes Fraise ou fraise par FRAISE.

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

3. La fonction de suppression d

La fonction de suppression **d** supprime les lignes comprises dans un intervalle donné.

sed "20,30d" fic

→ cette commande va supprimer les lignes 20 à 30 du fichier ''fic''.

On peut utiliser les expressions régulières:

sed "/toto/d" fic

→ cette commande supprime les lignes contenant la chaîne toto.

Si au contraire on ne veut pas effacer les lignes contenant la chaîne toto (toutes les autres sont supprimées), on tapera:

sed "/toto/!d" fic

En fait les lignes du fichier d'entrée ne sont pas supprimées, elles le sont au niveau de la sortie standard.

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

4. Les fonctions p, l, et =

La commande **p (print)** affiche la ligne sélectionnée sur la sortie standard.
Avec l'option **-n** onn'affichera que les lignes impactées.

La commande **l (list)** affiche la ligne sélectionnée sur la sortie standard avec en plus les caractères de contrôles en clair avec leur code ASCII (deux chiffres en octal).

La commande **=** donne le numéro de la ligne sélectionnée sur la sortie standard.

Ces trois commandes sont utiles pour le débogage, quand vous mettez au point vos programmes sed.

sed "/toto/=" fichier

→ Cette commande va afficher le numéro de la ligne contenant la chaîne toto.

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

5. Les fonctions q, r et w

La fonction **q (quit)** va interrompre l'exécution de sed, la ligne en cours de traitement est affichée sur la sortie standard (uniquement si -n n'a pas été utilisée).

La fonction **r (read)** lit le contenu d'un fichier et écrit le contenu sur la sortie standard.

La fonction **w (write)** écrit la ligne sélectionnée dans un fichier.

sed "/^toto/w resultat" fichier

→ Cette commande va écrire dans le fichier resultat toutes les lignes du fichier fichier commençant par la chaîne toto.

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

6. Les fonctions a et i

La fonction **a (append)** va placer un texte après la ligne sélectionnée, La syntaxe est la suivante:

a
le texte

La fonction **i (insert)** va placer un texte avant la ligne sélectionnée. La syntaxe est la suivante:

i\le texte

Si votre texte tient sur plusieurs lignes la syntaxe pour le texte est la suivante:

ligne 1 du texte
**ligne 2 du texte **
**ligne n du texte **
dernière ligne

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

Concrètement vous pouvez appeler la fonction **i ou a** dans un fichier de commande de sed.
Par exemple, soit votre fichier prog.sed suivant:

```
1i\
début du traitement
s/[tT]oto/TOTO/g
$a \ fin du traitement\
de notre fichier
```

On exécute la commande en tapant:

sed -f prog.sed fichier-a-traiter

prog.sed a pour effet d'**insérer (i)** avant la première ligne **(1)** le texte "début de traitement", et d'**ajouter (a)** après la dernière ligne (**$**) le texte "**fin du traitement** (retour à la ligne) **de notre fichier**".

Said.elmamouni@univ-nantes.fr

La commande sed

Les commandes de recherche

7. sed et les sous chaînes

La commande:

sed -e "s/\([0-9][0-9]*\)/aa\1aa/" fichier

La sous-expression (sous-chaîne) **\([0-9][0-9]*\)** désigne un ou plusieurs chiffres, chacun sera entouré des caractères **aa**. La chaîne **to2to** deviendra **toaa2aato**.

Said.elmamouni@univ-nantes.fr