



PARTIE V

.....

➤ La commande FIND

II – CHERCHER : FIND

- `find` est une commande qui va parcourir votre disque à la recherche de vos fichiers. Elle peut s'utiliser avec trois paramètres, dans l'ordre : où chercher, que chercher et que faire avec.
- On peut rechercher des fichiers en fonction de leur nom (`-name`), de leur taille (`-size`), de leur date de dernier accès (`-atime`)...
- Au lieu d'afficher les fichiers trouvés, on peut automatiquement les supprimer avec `-delete` ou exécuter la commande de son choix sur chacun d'eux avec `-exec`.

Fonctionnement de la commande **find**

find « où » « quoi » « que faire avec ». Seul le paramètre « quoi » est obligatoire.

- **Où** : c'est le nom du dossier dans lequel la commande va faire la recherche. Tous les sous-dossiers seront analysés. Contrairement à **locate**, il est donc possible de limiter la recherche à **/home** par exemple. Par défaut, si ce paramètre n'est pas précisé, la recherche s'effectuera dans le dossier courant et ses sous-dossiers.
- **Quoi** : c'est le fichier à rechercher. On peut rechercher un fichier par son nom, mais aussi en fonction de la date de sa création, de sa taille, etc. Ce paramètre est obligatoire.
- **Que faire avec** : il est possible d'effectuer des actions automatiquement sur chacun des fichiers trouvés (on parle de « post-traitement »). L'action la plus courante consiste à afficher simplement la liste des fichiers trouvés.

FIND

```
find « logo.png »
```

Si la recherche n'affiche rien, cela signifie qu'aucun fichier n'a été trouvé.

FIND

si je suis dans mon home mais que je veux rechercher dans un autre répertoire, il faudra préciser en premier paramètre le répertoire dans lequel chercher.

Par exemple, si je veux retrouver tous les fichiers qui s'appellent `syslog` situés dans `/var/log` (et ses sous-répertoires), je dois écrire :

```
find /var/log/ -name « syslog"
```


FIND

- Ici, nous avons affiché tous les fichiers qui commençaient par « syslog ».
- Si on avait voulu avoir la liste des fichiers qui se terminent par « syslog », on aurait écrit "`*syslog`".
De même, si on avait voulu avoir la liste des fichiers qui contiennent « syslog », que ce soit au début, au milieu ou à la fin, on aurait écrit "`*syslog*`".
- L'étoile est donc un joker qui signifie « il peut y avoir n'importe quel caractère à cet endroit ».
- Et si je veux rechercher sur tout le disque dur, et pas seulement dans un dossier ?
- Facile, il suffit d'indiquer le répertoire racine /. sous Linux, tous les dossiers sont situés dans un sous-niveau de /

Recherche en fonction de la taille

Vous ne connaissez pas le nom du fichier que vous recherchez ?

Il y a bien d'autres façons de retrouver des fichiers (ou des dossiers, d'ailleurs).

Par exemple, on peut rechercher tous les fichiers qui font plus de 10 Mo.

```
desktop@machine-desktop:/var$ find -size +10M
```

Rappel : le tilde ~

signifie « rechercher dans mon home », chez moi « `/home/raouf/` ».

Au lieu de se baser sur le nom, on se base ici sur la taille (`-size`). Le `+10M` indique que l'on recherche des fichiers de plus de 10 Mo. On peut aussi utiliser `k` pour les Ko, `G` pour les Go, etc.

Vous pouvez aussi utiliser un moins « `-` » à la place du « `+` » pour obtenir par exemple les fichiers de moins de 10 Mo. Et si vous enlevez le « `+` », la commande cherchera des fichiers de 10 Mo exactement (ni plus, ni moins).

Recherche à partir de la date de dernier accès

Vous êtes sûrs d'avoir accédé à votre rapport au format `.key` il y a moins de 7 jours, mais vous n'arrivez pas à le retrouver ?

Avec `-atime`, vous pouvez indiquer le nombre de jours qui vous séparent du dernier accès à un fichier.

```
desktop:~$ find -name "*.key" -atime -7  
/home/user/ecriture/resume_infos_juin.odt
```

Recherche par le nom + une recherche par la date.

Si je ne me souvenais plus de l'extension du fichier, j'aurais dû utiliser seulement `-atime`, mais ça m'aurait probablement donné beaucoup de fichiers (en fait, tous les fichiers modifiés depuis 7 jours dans mon répertoire personnel !).

Là encore, vous pouvez utiliser un « `+` » à la place... ou même enlever le signe pour rechercher un fichier auquel on aurait accédé il y a exactement 7 jours.

Afficher les fichiers de façon formatée

Par défaut, on liste juste les noms des fichiers trouvés. On peut cependant avec l'option `-printf`,

Exemple :

```
desktop:~$ find . -name "*.jpg" -printf "%p - %u\n"
```

Supprimer les fichiers trouvés

Ici, j'affiche le nom du fichier, un tiret et le nom du propriétaire de ce fichier. Le `\n` permet d'aller à la ligne.

Je vous conseille fortement de lire la doc', à la section « `-printf` » (faites une recherche). Direction : `man find` ! Vous y trouverez tous les éléments utilisables avec `-printf`, en plus du `%p` et du `%u`.

Supprimer les fichiers trouvés

Un des usages les plus courants de `find`, à part retrouver des fichiers, consiste à les supprimer.

Si je veux faire le ménage dans mon home et par exemple supprimer tous mes fichiers « `jpg` », je vais écrire ceci :

```
find -name "*.jpg" -delete
```

Appeler une commande

Avec `-exec`, vous pouvez appeler une commande qui effectuera une action sur chacun des fichiers trouvés.

Imaginons que je souhaite mettre un `chmod` à 600 pour chacun de mes fichiers « jpg », pour que je sois le seul à pouvoir les lire :

```
find -name "*.jpg" -exec chmod 600 {} \;
```

EXERCICES POTENTIELS D'EXAMENS SUR FIND

- <http://www.tecmint.com/35-practical-examples-of-linux-find-command/>

1. Find Files With 777 Permissions

Find all the files whose permissions are 777.

```
# find . -type f -perm 0777 -print
```

2. Find Read Only Files

Find all Read Only files.

```
# find / -perm /u=r
```

3. Find Executable Files

Find all Executable files.

```
# find / -perm /a=x
```

4. (Be careful)Find Files with 777 Permissions and Chmod to 644

Find all 777 permission files and use chmod command to set permissions to 644.

```
$ find $HOME -type f -perm 0777 -print -exec chmod 644 {} \;
```

5. Find Directories with 777 Permissions and Chmod to 755

Find all 777 permission directories and use chmod command to set permissions to 755.

```
# find / -type d -perm 777 -print -exec chmod 755 {} \;
```

6. Find and remove single File

To find a single file called tecmint.txt and remove it.

```
# find . -type f -name "tecmint.txt" -exec rm -f {} \;
```

7. Find all Empty Files

To file all empty files under certain path.

```
# find /tmp -type f -empty
```

8. Find all Empty Directories

To file all empty directories under certain path.

```
# find /tmp -type d -empty
```

9. File all Hidden Files

To find all hidden files, use below command.

```
# find /tmp -type f -name ".*"
```

10. Find Single File Based on User

To find all or single file called tecmint.txt under / root directory of owner root.

```
# find / -user root -name tecmint.txt
```

11. Find all Files Based on User

To find all files that belongs to user Tecmint under /home directory.

```
# find /home -user tecmint
```

12. Find Last 50 Days Modified Files

To find all the files which are modified 50 days back.

```
# find / -mtime 50
```

13. Find Last 50 Days Accessed Files

To find all the files which are accessed 50 days back.

```
# find / -atime 50
```

14. Find Last 50-100 Days Modified Files

To find all the files which are modified more than 50 days back and less than 100 days.

```
# find / -mtime +50 -mtime -100
```

15. Find Changed Files in Last 1 Hour

To find all the files which are changed in last 1 hour.

```
# find / -cmin -60
```

16. Find Modified Files in Last 1 Hour

To find all the files which are modified in last 1 hour.

```
# find / -mmin -60
```

17. Find Accessed Files in Last 1 Hour

To find all the files which are accessed in last 1 hour.

```
# find / -amin -60
```

18. Find 50MB Files

To find all 50MB files, use.

```
# find / -size 50M
```

19. Find Size between 50MB – 100MB

To find all the files which are greater than 50MB and less than 100MB.

```
# find / -size +50M -size -100M
```

20. Find and Delete 100MB Files

To find all 100MB files and delete them using one single command.

```
# find / -size +100M -exec rm -rf {} \;
```

21. Find Specific Files and Delete

Find all .mp3 files with more than 10MB and delete them using one single command.

```
# find / -type f -name *.mp3 -size +10M -exec rm {} \;
```

Les options de find : Exercices et solution

.....

1. **Comment chercher tous les fichiers dont les noms commencent par un «a» majuscule ou une minuscule, suivi d'éventuellement quelques lettres ou chiffres, et se terminent par un chiffre entre 3 et 6 ?** C'est l'option `-name` qui permet de spécifier le nom du ou des fichiers recherchés. On peut indiquer le nom d'un fichier complet (`bofichier.tex`), ou utiliser des expressions régulières (celles du shell, pas celles de `grep`...) :
 - L'étoile (*) désigne «un ou plusieurs caractères»;
 - Le point d'interrogation remplace un (et un seul) caractère quelconque;
 - Les crochets permettent de désigner une série de caractères au choix.
2. Dans notre cas, le premier caractère est un «a» ou un «A» (`[aA]`), suivi de quelque chose (*) et terminé par un chiffre entre 3 et 6 (`[3456]` ou `[3-6]`). On écrit donc :
`find . -name '[aA]*[3-6]' -print`
- 3.
4. **Comment fait-on pour indiquer que le fichier recherché a été modifié il y a plus de 30 jours ? Il y a 30 jours ? Il y a moins de 30 jours ?** C'est l'option `-mtime` qui permet de donner une indication en jours. La syntaxe varie selon ce que l'on souhaite :
 - `-mtime 30` : le fichier a été modifié il y a 30 jours;
 - `-mtime +30` : le fichier a été modifié il y a 30 jours ou plus;
 - `-mtime -30` : le fichier a été modifié il y a 30 jours ou moins.

<http://www.tuteurs.ens.fr/unix/exercices/solutions/find-sol.html>

.....

1. Comment faire pour dire que le fichier a été modifié plus récemment qu'un autre fichier donné ? On utilise l'option `-newer` («plus récent»). Par exemple, on cherche un fichier `.tex` modifié plus récemment que `bofichier.tex` :

```
find . -newer bofichier.tex -name '*.tex'
```

2.

On peut raffiner la demande, en combinant cette option avec l'option `-mtime` : cherchons les fichiers modifiés plus récemment que `bofichier.tex`, mais il y a plus de 5 jours :

```
find . -newer bofichier.tex -mtime +5 -name '*.tex'
```

3.

4. Comment fait-on pour spécifier que le fichier recherché est un répertoire ? On utilise l'option `-type` pour spécifier le genre de fichier recherché : les principaux sont `f` (*file*) pour un fichier normal, et `d` (*directory*) pour un répertoire. On tape donc :

```
find . -type d
```

5.

6. Comment indiquer que le fichier recherché à une taille supérieure à une taille donnée ? On utilise l'option `-size`, suivie d'un nombre et d'une lettre indiquant l'unité de mesure (`c` : octets, `k` : kilo-octets). Comme pour `-mtime`, on utilise `+`, `-` ou `[rien]` pour indiquer que la taille est, respectivement, supérieure, inférieure ou égale à la valeur donnée.

Par exemple, on recherche un fichier modifié il y a moins de 12 jours et dont la taille soit supérieure à 30 K :

```
find . -type f -size +30k -mtime -12 -print
```