

Linux

Commande **find**

Par

Anissa Omrane



find

La commande find

- Outil puissant en ligne de commande pour effectuer efficacement diverses tâches d'administration système, telles que la gestion de l'espace disque, les opérations de fichiers récursives et les sauvegardes.
- La commande de recherche de Linux recherche les fichiers et les dossiers selon des critères définis et permet en outre et d'effectuer une action sur les résultats de la recherche.

Syntaxe de base

- `$find` Rép-de-recherche [-option] [Action]
- Exemples 1: Liste tous les fichiers du répertoire courant ainsi que les sous-répertoire:
 - `find`
 - `find .`
 - `find . -print`
- Exemple 2
 - `find /home/user -name "*.txt"`

Options de recherche

- **-name** : Recherche des fichiers par leur nom
- **-type** : Recherche des fichiers par leur type (fichier, répertoire, lien symbolique, etc.)
- **-perm** : recherche sur les droits d'accès du fichier
- **-user** : recherche sur le propriétaire du fichier,
- **-group** : recherche sur le groupe auquel appartient le fichier
- **-links** : recherche sur le nombre de liens du fichier
- **-size** : Recherche des fichiers par leur taille

Options de recherche (autres)

- `-atime` recherche par date de dernier accès en lecture du fichier,
 - `-mtime` recherche par date de dernier accès en modification du fichier,
 - `-ctime` recherche par date de création du fichier.
-
- `-amin` recherche par date de dernier accès en lecture du fichier,
 - `-mmin` recherche par date de dernier accès en modification du fichier,
 - `-cmin` recherche par date de création du fichier.

Rechercher par nom : `-name`

- ❑ `$ find . -name "a.txt"`
 - ❑ `.` fait référence au répertoire courant , **`-name`** spécifie les critères à chercher
 - ❑ Cette recherche avec `-name` test est sensible à la casse.
 - ❑ On peut pour cela utiliser **`-iname`**: Exemple : `$find . -iname "a.txt "`
- ❑ Utilisation des caractères Joker
 - ❑ `*` est utilisé pour représenter une partie inconnue d'un nom de fichier ou de répertoire.
 - ❑ `$ find . -name "*.jpg "` le modèle générique `*.jpg`:
 - ❑ `$ find /home -name "*.jpg"`
- ❑ Exemple utile: `find /home -name "*.jpg" 2>/dev/null`
 - ❑ Si trop d'erreurs d'autorisations refusées, `2> /dev/null` à la fin de la commande.
 - ❑ Redirige les messages d'erreur vers `/dev/null` et donne une sortie plus claire:

Recherche par type de fichier – `-type`

- En utilisant l'option `-type`, On peut rechercher des fichiers par type.
- Les types de fichiers peuvent être:
 - ▣ `f` plain files
 - ▣ `d` directories
 - ▣ `l` symbolic links
 - ▣ `b` block devices
 - ▣ `c` character devices
 - ▣ `p` named pipes
 - ▣ `s` sockets
- Par exemple, l'utilisation du `-type d` ne listera que les répertoires:

```
$ find . -type d
```

Recherche par taille de fichier . `-size` `-empty`

- Rechercher des fichiers vides `-empty` pour trouver des répertoires et des fichiers vides
`$find . -empty`
- `-size` permet de chercher en fonction de la taille
 - ▣ Le `+` : les fichiers de recherche sont plus grands que le nombre suivant.
 - ▣ Le moins (`-`) peut être utilisé pour indiquer une valeur inférieure à.
 - ▣ Ne pas utiliser de signe signifierait correspondre exactement à la taille.
- Exemple : `$find . -size +1G`
 - ▣ On cherche les fichiers dont la taille est `+ 1G`.
 - ▣ Le numéro est suivi de l'unité de taille
 - ▣ Les unités peuvent être:

b	512 blocs	by
c	Octets	
k	Kilobytes	
M	Mégaoctets	
G	Gigaoctets	

Recherche `-user` `-perm`

- Recherche par nom d'utilisateur avec `-user`

Chercher tous les fichiers et répertoires de `aaaaa` dans le répertoire `/home`:

```
$find /home -user aaaa 2>/dev/null
```

- Recherche par mode avec `-perm`

Le `-perm` pour chercher des fichiers avec des autorisations spécifiques.

Exemple : chercher avec les autorisations `777`:

```
$find /home -perm 777
```

L'exemple recherche des fichiers qui a au moins des autorisations `777`

Recherche par temps d'accès

- (a/m/c) time

- -atime
- -mtime:
- -ctime:

- ▣ Recherche de fichiers dont l'heure de changement de statut de fichier (ctime) remonte à environ $n \times 24$ heures.
- ▣ n est un nombre entier.

- (a/m/c) min

- -amin
- -mmin
- -cmin :

- ▣ rechercher des fichiers en fonction de date de changement d'état (change time).
- ▣ C'est le moment où des métadonnées du fichier, comme les autorisations, les propriétaires, les dates de modification, ont été modifiées.
- ▣ Il ne concerne pas les modifications du contenu.

Recherche par heure –cmin par temps d'accès -atime

- **Recherche par heure** de modification avec -cmin tester.

Rechercher tous les fichiers modifiés au cours des 60 dernières minutes (moins de 60):

```
$find . -cmin -60
```

+60 Pour les fichiers modifiés avant les 60 dernières minutes

- Recherche par heure du dernier accès, avec -atime tester.

la commande suivante recherche les fichiers non consultés au cours des 180 derniers jours :

```
$find . -atime +180
```

Les opérateurs logiques

Il est possible de combiner plusieurs critères de recherche moyennant les opérateurs logique :

□ **-a** ou **and** : `critère1 -a critère2`

□ **!critère** **négation**

□ **-o** ou **or** : `\(critère1 -o critère2\)`

Exemples avec les opérateurs

- Par exemple, la commande suivante recherche les fichiers de plus de 100 Mo appartenant à l'utilisateur moi:

```
$ find /home -user me -and -size +100M 2>/dev/null
```

- La commande suivante recherche les fichiers de plus de 100 Mo appartenant à l'utilisateur me ou à l'utilisateur aaaa:

```
$ find /home \( -user aaaa -or -user me \) -and -size +100M 2>/dev/null
```

Actions

find donne les résultats de la recherche, puis le choix d'effectuer une **action** sur eux. Voici quelques actions prédéfinies:

- **-delete** Supprime les fichiers répondant aux critères de recherche
- **-ls** Affiche une sortie ls détaillée
- **-print** Affiche le chemin complet des fichiers correspondants.
C'est l'action par défaut si aucune autre action n'est spécifiée
- **-exec** Exécute la commande qui suit, sur chaque élément trouvé
- **-ok** Exécute la commande qui suit, sur chaque élément trouvé avec demande de confirmation

L'action **-exec**

- La **-exec** permet d'exécuter une commande sur les résultats de la recherche.
- Syntaxe :
-exec command { } \;
 - ▣ **Command** la commande à exécuter sur les résultats de la recherche, Exemple : **rm**, **mv** ou **cp**.
 - ▣ **{ }** représente les résultats de la recherche.
 - ▣ La commande se termine par un point-virgule ;
 - ▣ échappé par une barre oblique inverse. \;

Exemples avec les actions

- Chercher tous les fichiers vides et les supprimer

```
$find . -empty -delete
```

NB: avant d'utiliser l'action de suppression, il est toujours prudent d'exécuter la commande une fois avec **-print** et confirmer les résultats.

- Pour rechercher et supprimer tous les fichiers vides :

```
$find . -empty -exec rm {} \;
```

- Voici un autre exemple utilisant **-exec** action. La commande suivante copie tout **.png** fichiers image dans le répertoire **backup/images**

```
$find . -name "*.png" -exec cp{} /backups/images \;
```



Lab sur la commande find

Trouver des fichiers par noms

- `#find . -name myfile.txt`
- `# find /home`
- `# find /home -name "spin.exo"`
- `# find /home -name "*.pdf"`
- `# find /home -iname "*.PDF"`

Trouver des fichiers par noms

- `#find . -name myfile.txt`

tous les fichiers dont le nom « myfile.txt » dans le répertoire courant.

- `# find /home`

Chercher des fichiers dans le répertoire home.

- `# find /home -name "spin.exo"`

Chercher le fichier par son nom dans le répertoire home.

- `# find /home -name "*.pdf"`

Nous pouvons aussi utiliser des caractères génériques (wildcards)

- `# find /home -iname "*.PDF"`

Ignorer la casse :Il est souvent utile d'ignorer la casse lors de la recherche d'un nom de fichiers. Pour ignorer la casse, utilisez simplement l'option « **-iname** » au lieu de l'option « **-name** ».

Find : exemples

□ `# find / -type d -name Formation`

Trouver des répertoires par nom Cherchez tous les répertoires dont le nom est Formation à partir de la racine /.

□ `# find ~ -type f -name ".*"`

Trouver des fichiers cachés Les fichiers cachés sur Linux qui commencent par un point. Il est donc facile de mentionner cela dans les critères de recherche et de lister tous les fichiers cachés.

□ `# find ./dossier1 ./dossier2 -type f -name "image*"`

Chercher dans plusieurs dossiers en parallèle.

Find : examples

- `# find / -type d -name Formation`
- `# find ~ -type f -name ".*"`
- `# find ./dossier1 ./dossier2 -type f -name "image*"`

□ `# find / -type d -name Formation`

Trouver des répertoires par nom Cherchez tous les répertoires dont le nom est Formation à partir de la racine /.

□ `# find ~ -type f -name ".*"`

Trouver des fichiers cachés Les fichiers cachés sur Linux qui commencent par un point. Il est donc facile de mentionner cela dans les critères de recherche et de lister tous les fichiers cachés.

□ `# find ./dossier1 ./dossier2 -type f -name "image*"`

Chercher dans plusieurs dossiers en parallèle.

Exemples find

□ Trouver des fichiers par nom exemples divers

- ▣ `find ./test -name file*`

- ▣ `find . -name *.txt`

- ▣ `find ./ -name "qa*"`

- ▣ `find ./ -type f -name "qa*"`

- ▣ `find ./ -type d -name "qa*"`

- ▣ `find ./ -iname "qa*"`

- ▣ `find ./test ./numeric -name file22.txt -type f`

- ▣ `find . -type f (-name "*.txt" -o -name "*.pdf" -o -name "*.doc")`

Exemples find

- ❑ `$find . -type f -perm 0666 -print`

Trouvez tous les fichiers dont les permissions sont 0666.

- ❑ `$find / -type f ! -perm 0666`

Chercher des fichiers qui n'ont pas les permissions 0666 Trouvez tous les fichiers dont ne dispose pas les permissions 0666.

- ❑ `$find / -perm /u=r` chercher les fichiers en lecture seule

- ❑ `$find / -perm /a=x` : Trouver les fichiers exécutables

- ❑ `$find -perm 777`

- ❑ `$find -perm /222` (/ avec une permission d'écriture)

- ❑ `find . -type d -perm 644`

Exemples find

- ❑ `# find . -user anissa`

- ❑ `find /home -user jack`

- ❑ `find /home -user jack -iname "*.txt"`

Trouver des fichiers appartenant à un utilisateur particulier

- ❑ `# find . -user anissa -name '*.pdf'`

Nous pouvons également spécifier le nom du fichier avec les critères de l'utilisateur

- ❑ `# find /var/www -group programmeur`

- ❑ Trouver des fichiers appartenant à un groupe particulier.

Exemples find

- `# find / -mtime 10`

Les fichiers modifiés depuis 10 jours

- `# find / -atime 3`

Chercher les fichiers consultés au cours des 3 derniers jours

- `# find / -mtime +10 -mtime -50`

Trouvez tous les fichiers modifiés entre 10 et 50 jours auparavant.

- `# find /home/tp -cmin -5`

Chercher les fichiers modifiés au cours des 5 dernières minutes

- `# find / -mmin -60`

Chercher les fichiers modifiés depuis une heure

-

Examples find

- ❑ `find . -mtime -10 -type d`
- ❑ `find /var/backup -type f -mtime +3`
- ❑ `find /var/backup -type f -mmin +60`
- ❑ `find /directory_path -mtime -1 -ls`
- ❑ `find /home -atime +10 accessed`
- ❑ `find /home -atime 10`
- ❑ `find /home -atime -10`
- ❑ `find /home -type f -mtime +6 -mtime -15`
- ❑ `find . -amin -10 -type f`
- ❑ `find . -amin -10 -type d`

Exemples find

- ❑ `find / -size 100M`
- ❑ `find / -size 30M`
- ❑ `find -size +2M`
- ❑ `find . -type f -size -10M`
- ❑ `find / -size +100M -size -200M`
- ❑ `find / -type d -size +20k`
- ❑ `find ./ -type f -size 0` ou `find ./ -type f -empty`
- ❑ `find ./ -type d -empty`

Exemples find

- Trouver des fichiers qui ont une taille > 10 Mo et < 40 Mo.

```
# find / -size +10M -size -40M
```

- Trouver des fichiers et des répertoires vides La commande suivante utilise l'option « -empty », qui cherche tous les fichiers vides.

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

- Pour lister tous les répertoires vides, utilisez l'option « -type » avec le paramètre « d ».

```
□ # find ~/ -type d -empty
```

Exemples find avec les actions

- ❑ `find ./numeric -type f -exec grep -l -i "hyperconvergence" {} \;`
- ❑ `find /path/ -type f -name '*.txt' -mtime +8 //older then 8 days`
- ❑ `find -name "*.conf" | ls -l`
- ❑ `find ver -name "*.php" -type f -exec chmod 755 {} \;`
- ❑ `find . -type f -perm 644 -exec chmod 655 {} \;`
- ❑ `find /path/ -type f -name '*.pdf' -mtime +20 -exec mv {} /backup1 \;`

Exemples find avec les actions

- ❑ `find . -type f -name "file to delete" -exec rm -f {} \;`
- ❑ `find . -type d -name "dir-to-delete" -exec rm -rf {} \;`
- ❑ `find . -type f -name "til*" -exec rm {} \;`
- ❑ `find . -type d -name "til*" -exec rm {} \;`
- ❑ `find . -name "til*" -exec rm -rf {} \;`
- ❑ `find . -type f -name "*.txt" -exec rm -f {} \;`

Exemples find avec les actions

- ❑ `find . -type f -name "*.txt" -exec rm -i {} \;`
- ❑ `find . -type f -name '*.bak' -mtime +20 -exec rm {} \;`
- ❑ `find . -type d -name dir22 -exec rm -rf {} \;`
- ❑ `find . -type d -iname Dir22 -exec rm -rf {} \;`
- ❑ `find . -type d -iname Dir22 -exec rm -rf -i {} \;`
- ❑ `find / -type f -size 0 -exec rm -f {} \;`
- ❑ `find / -type d -empty -exec rmdir {} \;`
- ❑ `find / -type d -empty -delete`

Exemples find avec les actions

- ❑ `find \( <criterion> -o <criterion> \) -exec <command>`
- ❑ `find . -type f \( -name "*.txt" -o -name "*.log" \) -mtime -1`
- ❑ `find . \( -type f -or -type d \) -name "*.txt" -ok rm {} \;`
- ❑ `find . -type f -mtime -7 -size +1M`