

[Accueil](#) > [Cours](#) > [Reprenez le contrôle à l'aide de Linux !](#) > [Rechercher des fichiers](#)

Reprenez le contrôle à l'aide de Linux !

 30 heures  Facile

Mis à jour le 29/06/2021



Rechercher des fichiers

Sous Linux, les fichiers sont organisés d'une façon assez particulière. Nous l'avons vu en affichant la liste des répertoires à la racine avec un `ls /`, il y a une foule de dossiers aux noms assez variés : `var`, `opt`, `etc`, `bin`, `dev` ...

Une partie de ces répertoires est là pour des raisons historiques, depuis l'époque d'Unix. Le problème, c'est qu'il peut être difficile de retrouver le fichier dont on a besoin dans cette foule de répertoires.

Pas de panique ! On dispose heureusement sous Linux d'outils très puissants pour rechercher un fichier sur le disque dur. Certains d'entre eux sont très rapides, d'autres plus lents mais aussi plus complets.

Partons à la recherche de ces fichiers !

locate : une recherche rapide



La première façon d'effectuer une recherche que nous allons voir est de loin la plus simple. La commande s'appelle `locate` (« localiser »). Elle est très rapide.

Utiliser `locate`

Son utilisation est intuitive, il suffit d'indiquer le nom du fichier que vous voulez retrouver. Par exemple :

```
mateo21@mateo21-desktop:~$ locate notes.txt  
/home/mateo21/notes.txt
```

La commande a retrouvé notre fichier `notes.txt` qui était situé dans `/home/mateo21` .

Essayons maintenant de retrouver ces vieilles photos d'Australie...

```
mateo21@mateo21-desktop:/var/log$ locate australie  
/home/mateo21/photos/australie1.jpg  
/home/mateo21/photos/australie2.jpg  
/home/mateo21/photos/australie3.jpg
```

`locate` vous donne tous les fichiers qui contiennent le mot « australie » dans leur nom. Que ce soient des fichiers ou des dossiers, elle ne fait pas la différence. Elle vous donne la liste complète des fichiers qu'elle a trouvés.



Il existe aussi la commande `slocate` qui est un peu plus récente, mais on ne la retrouve pas sur toutes les distributions par défaut. Vous pouvez toutefois l'installer rapidement avec un `apt-get` si vous ne l'avez pas.

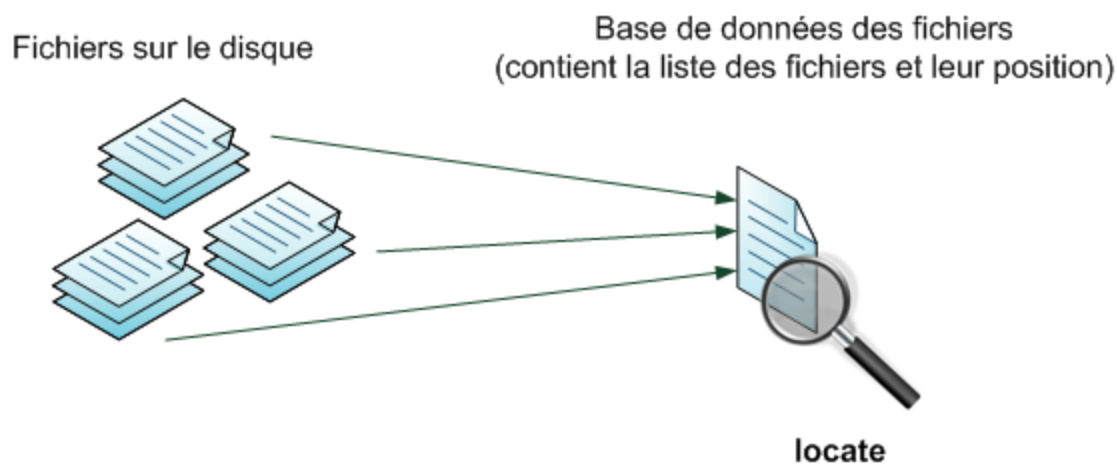
La commande `slocate` est identique à `locate` , à la différence près qu'elle vérifie les droits des fichiers avant de les lister. Avec `slocate` , un utilisateur ne pourra donc pas voir un nom de fichier apparaître s'il n'a pas le droit de le lire, tandis qu'avec un `locate` le nom du fichier serait apparu dans les résultats quand même.

La base de données des fichiers



Je ne comprends pas. Je viens de créer des fichiers (avec la commande `touch` par exemple), et `locate` ne me renvoie aucun résultat. Pourquoi ?

C'est justement le défaut de `locate` dont je voulais vous parler : la commande ne fait pas la recherche sur votre disque dur entier, mais seulement sur une base de données de vos fichiers (figure suivante).



Votre problème, c'est que les fichiers viennent tout juste d'être créés et n'ont pas encore été répertoriés dans la base de données. Ils ne seront donc pas découverts par `locate`.

Une fois par jour, votre système mettra à jour la base de données. Donc, si vous réessayez demain, il est probable que `locate` trouve enfin votre fichier.



Mais... je ne vais pas attendre 24 h pour retrouver un fichier, tout de même !

Non, bien sûr !

Vous pouvez forcer la commande `locate` à reconstruire la base de données des fichiers du disque dur. Cela se fait avec la commande `updatedb`, à exécuter en root (avec `sudo`) :

```
sudo updatedb
```

La mise à jour de la liste des fichiers peut être un peu longue, il faudra patienter. Une fois que c'est fini, réessayez de faire un `locate`, il devrait maintenant trouver votre fichier.

En résumé, `locate` est pratique car rapide et facile à utiliser.

Cependant, `locate` donne parfois trop de résultats car elle recherche dans tous les répertoires du disque dur, elle n'est donc pas très précise. De plus, les fichiers qui viennent tout juste d'être créés ne seront pas découverts, à moins d'exécuter `updatedb`.

Quand `locate` ne suffit pas, on a besoin d'une commande plus puissante. On sort l'artillerie lourde : `find`.

find : une recherche approfondie



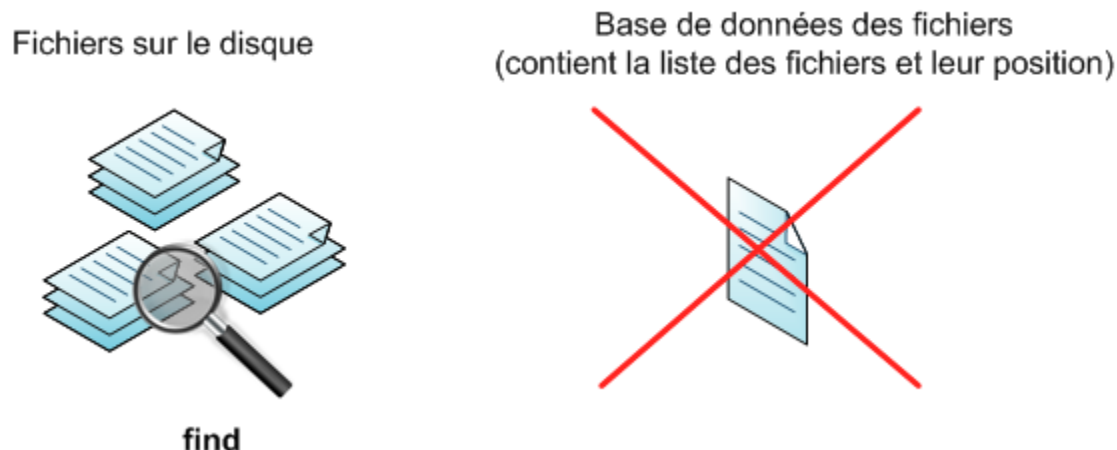
`find` est la commande de recherche par excellence pour retrouver des fichiers, mais aussi pour

effectuer des opérations sur chacun des fichiers trouvés. Elle est très puissante, permet donc de faire beaucoup de choses, et par conséquent... elle est un peu complexe.

Savoir la manipuler est toutefois indispensable, donc découvrons-la !

`find` recherche les fichiers actuellement présents

Contrairement à `locate`, `find` ne va pas lire dans une base de données mais au contraire parcourir tout votre disque dur (figure suivante). **Cela peut être très long** si vous avez plusieurs giga-octets de données !



Avec cette méthode de recherche, vous ne risquez pas de rater un fichier récent qui aurait pu être créé aujourd'hui. Et ce n'est pas le seul avantage, loin de là !

Fonctionnement de la commande `find`

La commande `find` s'utilise de la façon suivante :

`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, mais nous verrons que nous pouvons faire bien d'autres choses.

Par défaut, la commande `find` affiche les résultats trouvés et ne fait rien d'autre avec.

Utilisation basique de la commande `find`

Commençons doucement et voyons des exemples pratiques. Nous allons tout d'abord rechercher un fichier et afficher sa position.

Recherche à partir du nom

Je vais utiliser dans un premier temps le seul paramètre obligatoire : le nom du fichier à retrouver.

Je me place dans mon répertoire home et je vais essayer de retrouver un fichier appelé `logo.png` que j'ai égaré. Je dois écrire :

```
find -name "logo.png"
```

Le `-name "logo.png"` est un paramètre qui demande de retrouver un fichier qui s'appelle très exactement `logo.png`.

Voici le résultat :

```
mateo21@mateo21-desktop:~$ find -name "logo.png"
/home/mateo21/projet/images/logo.png
```

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



Comme nous n'avons pas précisé où rechercher, `find` a cherché dans le répertoire dans lequel nous nous trouvons et dans ses sous-répertoires. Le `~`

de l'invite de commandes signifie que j'étais dans mon home, c'est-à-dire dans `/home/mateo21/`. Tous les sous-répertoires ont été analysés.}

Maintenant, 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"
```

Essayons ça !

```
mateo21@mateo21-desktop:~$ find /var/log/ -name "syslog"
/var/log/syslog
/var/log/installer/syslog
```

Les paramètres correspondent à ceux que je vous ai présentés un peu plus tôt :

- **où** : dans `/var/log` (et ses sous-dossiers) ;
- **quoi** : le fichier `syslog` .

Il n'y a pas de paramètre « que faire avec », donc par défaut `find` choisit d'afficher les fichiers trouvés.

Notez que, contrairement à `locate` , `find` récupère uniquement la liste des fichiers qui s'appellent exactement comme demandé. Ainsi, s'il existe un fichier nommé `syslog2` , il ne sera pas listé dans les résultats. Pour qu'il le soit, il faut utiliser le joker : l'étoile « `*` » !

Exemple :

```
mateo21@mateo21-desktop:~$ find /var/log/ -name "syslog*"
/var/log/syslog.3.gz
/var/log/syslog.5.gz
/var/log/syslog.4.gz
/var/log/syslog
/var/log/syslog.6.gz
/var/log/syslog.2.gz
/var/log/syslog.1.gz
/var/log/installer/syslog
/var/log/syslog.0
```

Ici, nous avons affiché tous les fichiers qui commencent 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 `/`. Je vous rappelle que sous Linux, tous les dossiers sont situés dans un sous-niveau de `/`. C'est un petit peu l'équivalent du `C:` sous Windows.

```
find / -name "syslog"
```



La recherche depuis la racine prend beaucoup de temps si vous avez de nombreux fichiers. De plus, si vous n'êtes pas root, vous aurez de nombreux messages d'erreur vous indiquant que vous n'avez pas le droit d'aller dans certains répertoires. En général, à moins d'être très patient (ou désespéré), on ne fait pas de recherche depuis la racine.

Recherche à partir de la taille

Vous ne connaissez pas le nom du fichier que vous recherchez ? Pas de panique ! 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.

```
mateo21@mateo21-desktop:/var$ find ~ -size +10M
/home/mateo21/souvenirs.avi
/home/mateo21/backups/backup_mai.gz
/home/mateo21/backups/backup_juin.gz
```



Rappel : le tilde « `~` » signifie « rechercher dans mon home », en l'occurrence chez moi « `/home/mateo21/` ».

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 `.odt` 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.

```
mateo21@mateo21-desktop:~$ find -name "*.odt" -atime 6  
/home/mateo21/ecriture/resume_infos_juin.odt
```

J'ai combiné ici une recherche par le nom avec une recherche par la date. Si je ne me souvenais même 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.

J'utilise ici `6` et non `7` car la numérotation commence à 0 ! Ainsi, si nous sommes le mardi 29 août, j'écrirais :

- Hier, le lundi 28 août, il y a 1 jour → 0 ;
- Avant-hier, le 27 août, il y a 2 jours → 1.

Rechercher uniquement des répertoires ou des fichiers

On peut aussi rechercher uniquement des répertoires ou des fichiers.

Utilisez :

- `-type d` : pour rechercher uniquement des répertoires (*directories*) ;
- `-type f` : pour rechercher uniquement des fichiers (*files*).

Par défaut, `find` cherche des répertoires ET des fichiers. Ainsi, si vous avez un fichier appelé `syslog` et un répertoire appelé `syslog`, les deux résultats seront affichés.

Pour obtenir uniquement les répertoires qui s'appellent `syslog` (et non pas les fichiers), tapez donc :

```
find /var/log -name "syslog" -type d
```

Utilisation avancée avec manipulation des résultats

Pour l'instant, nous n'avons pas indiqué de paramètre « que faire avec » pour effectuer une action sur les résultats trouvés. Par défaut, les noms des fichiers trouvés étaient affichés.

En fait,


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

... est équivalent à :

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

`-print` signifie « afficher les résultats trouvés ».

Si le `-print` n'est pas écrit, la commande comprend toute seule qu'elle doit afficher la liste des fichiers.

On peut cependant remplacer ce `-print` par d'autres paramètres.

Afficher les fichiers de façon formatée

Par défaut, on liste juste les noms des fichiers trouvés. On peut cependant avec l'option `-printf`, qui rappellera à certains le langage C, manipuler un peu ce qui est affiché.

Exemple :

```
mateo21@mateo21-desktop:~$ find . -name "*.jpg" -printf "%p - %u\n"
./photos/australie1.jpg - mateo21
./photos/australie2.jpg - mateo21
./photos/australie3.jpg - mateo21
```

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
```

x

Soyez bien sûrs de ce que vous faites ! Il n'y aura pas de confirmation !

Et voilà, toutes les images ont disparu.

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 {} \;
```

La commande n'affiche rien s'il n'y a pas eu d'erreur.



Euh... comment ça marche, ce truc ?

Pour chaque fichier `.jpg` trouvé, on exécute la commande qui suit `-exec` :

- cette commande ne doit PAS être entre guillemets ;
- les accolades `{}` seront remplacées par le nom du fichier ;
- la commande doit finir par un `\;` obligatoirement.

C'est un peu compliqué au premier abord, mais c'est très puissant ! Vous pouvez faire ce que vous voulez avec ça.

Exercice : essayez de regrouper tous les fichiers `.jpg` éparpillés dans votre répertoire home dans un dossier `images`.



Si le fait que la commande ne vous demande pas de confirmation vous ennuie, vous pouvez utiliser `-ok` à la place de `-exec`. Le principe est le même, mais on vous demandera une confirmation pour chaque fichier rencontré. Il faudra répondre par « *y* » (*yes*) ou « *n* » (*no*) à chaque fois.

En résumé

- Pour rechercher un fichier sur tout le disque, la commande `locate` est très rapide mais ne trouvera pas les fichiers qui viennent d'être créés dans la journée. On peut mettre à jour la liste des fichiers qu'elle connaît en appelant `updatedb`.
- `find` est une commande plus puissante mais plus lente 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` .

Le professeur

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

[RTFM : LISEZ LE MANUEL !](#)

[QUIZ : QUIZ 2](#)



Découvrez aussi ce cours en...



Livre



PDF

OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation

Forum

Blog

Presse

OPPORTUNITÉS

Nous rejoindre

Devenir mentor

Devenir coach carrière

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique 

Mentions légales

Conditions générales d'utilisation

Politique de protection des données personnelles

Cookies

Accessibilité



Français



Télécharger dans
l'App Store