

Comment renommer plusieurs fichiers en utilisant find

En utilisant find

```
find . -type f -name 'file*' -exec sh -c 'x="{ }"; mv "$x" "${x}_renamed"' \;
```

Cependant, cela coûte très cher lorsqu'on a beaucoup de fichiers correspondants, car on démarre un nouveau shell (qui exécute un mv) pour chaque correspondance.

Une approche plus efficace et sécurisée est la suivante:

```
find . -type f -name 'file*' -exec mv {} {}_renamed \;
```

Cependant:

1. Il est très dangereux d'utiliser {} directement dans une commande shell (sh -c “...”) - vous devriez toujours le transmettre sous forme d'argument.
2. Toutes les versions de find prennent en charge la construction {}_renamed .

```
find . -type f -name 'file*' -print0 | xargs --null -I{} mv {} {}_renamed
```

A l'avantage d'enchaîner sh et mv par fichier alors que la première version ne démarre qu'un mv fichier par fichier,

Utilisation d'une boucle while

Une autre approche consiste à utiliser une boucle while read sur find output. Cela permet d'accéder à chaque nom de fichier sous forme de variable pouvant être manipulée sans avoir à s'inquiéter des coûts supplémentaires / des problèmes de sécurité potentiels liés à la création d'un processus sh -c séparé à l'aide de l'option -exec .

```
find . -type f -name 'file*' | while IFS= read file_name; do mv "$file_name" "${file_name##*/}_renamed" done
```

Et si le shell utilisé prend en charge l'option -d pour spécifier un délimiteur de read vous pouvez prendre en charge les fichiers nommés de manière étrange (par exemple, avec une nouvelle ligne) en utilisant les éléments suivants:

```
find . -type f -name 'file*' -print0 | while IFS= read -d '' file_name; do mv "$file_name" "${file_name##*/}_renamed" done
```

Changer récursivement les extensions de fichiers

```
find . -name "*.t1" -exec bash -c 'mv "$1" "${1%.t1}".t2' - '{}' \;
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=notes:bash-rename>

Last update: **2025/02/19 10:59**

