

? Les alias Linux : créer des raccourcis dans le terminal

Les alias permettent de créer des raccourcis personnalisés pour des commandes Linux. Ils sont très utiles pour simplifier des commandes longues ou ajouter des options par défaut.

? Sommaire

- [Définition d'un alias](#)
 - [Exemple classique : ll et ls](#)
 - [Créer un alias](#)
 - [Lister les alias](#)
 - [Supprimer un alias](#)
 - [Contourner un alias](#)
 - [Exemples pratiques](#)
 - [Persistance \(.bashrc\)](#)
-

? Qu'est-ce qu'un alias ?

Information

Un alias est un raccourci qui remplace une commande par une autre.

Exemple :

```
ll = ls -l
```

Lorsque vous tapez `ll`, Linux exécute en réalité `ls -l`.

? Exemple classique : ll, la et l

Sur certaines distributions, ces alias ne sont pas activés par défaut.

```
ll
```

Peut donner :

```
ll : commande introuvable
```

Attention

Les alias peuvent varier selon les distributions (Ubuntu, Debian, CentOS, Fedora...).

Les commandes équivalentes sans alias sont :

```
ls -l  
ls -la  
ls -CF
```

? Créer un alias

Syntaxe

```
alias nom='commande'
```

Exemple simple

```
alias ll='ls -l'
```

Exemple avec options

```
alias la='ls -la'  
alias l='ls -CF'
```

Astuce

Les alias sont actifs uniquement dans le shell courant (temporaire).

? Lister les alias existants

```
alias
```

Exemple de sortie :

```
alias ll='ls --color=auto -l'
alias la='ls --color=auto -la'
```

?? Supprimer un alias

Syntaxe

```
unalias nom
```

Exemple

```
unalias ll
```

Cela supprime l'alias uniquement dans la session actuelle.

Supprimer tous les alias

```
unalias -a
```

? Exécuter la commande sans alias

Si un alias existe mais que vous voulez exécuter la commande originale :

```
\ls
```

Le backslash ignore les alias.

Important

Un alias peut masquer une commande système. Toujours vérifier avec `type`.

Vérifier si une commande est un alias

```
type ls
```

Exemple :

```
ls is aliased to `ls --color=auto`
```

?? Exemples pratiques réels

? 1. Alias pour cp sécurisé

Ajout du mode interactif pour éviter les écrasements accidentels :

```
alias cp='cp -i'
```

? 2. Alias de sauvegarde automatique

```
alias safecp='cp --backup=numbered'
```

? 3. Alias utile pour logs

```
alias log='journalctl -xe'
```

? 4. Alias dangereux à éviter

```
alias rm='rm -i'
```

Utile pour éviter les suppressions accidentelles, mais peut gêner les scripts.

? Exemple complet : cp + alias + suppression

```
alias cp='cp -i'

cp fichier1 fichier2
```

Résultat : confirmation avant écrasement.

Supprimer l'alias :

```
unalias cp
```

? Rendre les alias permanents (.bashrc)

Information

Les alias créés avec la commande `alias` disparaissent à la fermeture du terminal.

Pour les conserver :

```
nano ~/.bashrc
```

Ajouter :

```
alias ll='ls -l'
alias la='ls -la'
alias l='ls -CF'
```

Puis appliquer :

```
source ~/.bashrc
```

Alternative : sans modifier .bashrc

Vous pouvez aussi créer des alias uniquement pour la session actuelle :

```
alias ll='ls -l'
```

Ils seront supprimés à la déconnexion.

Bonnes pratiques

- Éviter de surcharger les commandes critiques (rm, cp, mv) dans les scripts
- Documenter ses alias personnalisés
- Utiliser type pour diagnostiquer un comportement étrange

? Résumé

Commande	Action
<code>alias ll='ls -l'</code>	Créer un alias
<code>alias</code>	Afficher les alias
<code>unalias ll</code>	Supprimer un alias
<code>unalias -a</code>	Supprimer tous les alias
<code>\ls</code>	Ignorer les alias
<code>type ls</code>	Vérifier alias/commande

À retenir

Les alias sont des raccourcis temporaires puissants pour améliorer la productivité dans le terminal Linux.

Revision #2

Created 2026-04-24 13:27:45 UTC by Nico là

Updated 2026-06-11 12:27:12 UTC by Nico là