

# groups

La commande `groups` est utilisée pour afficher les groupes auxquels un utilisateur appartient sur un système Unix ou Linux. Lorsqu'elle est exécutée sans argument, la commande `groups` affiche les groupes de l'utilisateur courant (l'utilisateur qui exécute la commande). Voici un exemple d'utilisation de la commande `groups` :

```
groups
```

Le résultat sera une liste des groupes auxquels appartient l'utilisateur actuel, séparés par des espaces.

Par exemple, si l'utilisateur courant est `alice` et qu'elle appartient aux groupes `users` et `sudo`, la commande `groups` affichera quelque chose comme ceci :

```
alice users sudo
```

La commande `groups` est utile pour vérifier rapidement les groupes auxquels un utilisateur est associé, ce qui peut être important pour comprendre ses autorisations et ses privilèges sur le système. Vous pouvez également spécifier un nom d'utilisateur en argument pour afficher les groupes auxquels un utilisateur spécifique appartient, par exemple :

```
groups alice
```

Cela affichera les groupes auxquels l'utilisateur `alice` appartient.

## Scripts

La commande `groups` peut être utilisée dans des scripts shell pour récupérer la liste des groupes auxquels un utilisateur appartient et utiliser ces informations dans des opérations ou des prises de décision. Voici un exemple simple de la façon dont vous pourriez utiliser `groups` dans un script :

Supposons que vous ayez un script shell (par exemple, un script `bash`) et que vous souhaitiez effectuer une action différente en fonction des groupes auxquels l'utilisateur appartient. Voici un exemple de script qui utilise `groups` pour cela :

```
#!/bin/bash

# Obtient la liste des groupes de l'utilisateur courant
user_groups=$(groups)

# Vérifie si l'utilisateur appartient au groupe "admin"
if [[ "$user_groups" == *admin* ]]; then
    echo "L'utilisateur appartient au groupe 'admin'."
    # Effectuez ici les actions spécifiques pour le groupe "admin"
else
    echo "L'utilisateur n'appartient pas au groupe 'admin'."
```

```
# Effectuez ici les actions pour les utilisateurs qui ne sont pas dans  
le groupe "admin"  
fi  
  
# Autres actions communes à tous les utilisateurs
```

Dans cet exemple, le script récupère la liste des groupes de l'utilisateur courant à l'aide de la commande `groups` et stocke cette liste dans la variable `user_groups`. Ensuite, il vérifie si l'utilisateur appartient au groupe "admin" en utilisant une condition `if`. Vous pouvez personnaliser le script en fonction des groupes que vous souhaitez vérifier.

L'utilisation de `groups` dans des scripts peut être utile pour personnaliser le comportement du script en fonction des autorisations ou des groupes d'utilisateurs. Assurez-vous que le script est exécuté par un utilisateur qui a la permission d'exécuter la commande `groups`, et n'oubliez pas de gérer les erreurs potentielles, notamment si l'utilisateur n'appartient à aucun groupe.

From:

<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:

<https://www.abonnel.fr/informatique/linux/commandes/groups>

Last update: **2023/11/27 19:33**

