

Voici plusieurs démonstrations de la commande **grep** sous Linux, couvrant diverses options et usages courants :

Introduction à **grep**

La commande **grep** recherche des motifs dans des fichiers ou des flux d'entrée. Elle est très puissante et offre de nombreuses options pour affiner les recherches.

Démo 1 : Recherche simple

Rechercher un motif dans un fichier

```
grep "erreur" fichier.log
```

Cela affichera toutes les lignes contenant le mot "erreur" dans **fichier.log**.

Rechercher un motif dans plusieurs fichiers

```
grep "erreur" fichier1.log fichier2.log
```

Cela affichera les lignes contenant "erreur" dans **fichier1.log** et **fichier2.log**.

Démo 2 : Options de recherche

Recherche insensible à la casse (option -i)

```
grep -i "erreur" fichier.log
```

Cela trouvera "erreur", "Erreur", "ERREUR", etc.

Afficher les numéros de ligne (option -n)

```
grep -n "erreur" fichier.log
```

Cela ajoutera les numéros de ligne avant chaque ligne correspondante.

Rechercher des mots entiers uniquement (option -w)

```
grep -w "erreur" fichier.log
```

Cela ne trouvera que "erreur" en tant que mot entier, pas "erreurs".

Rechercher de manière récursive dans les répertoires (option -r)

```
grep -r "erreur" /var/log
```

Cela recherchera "erreur" dans tous les fichiers du répertoire `/var/log` et ses sous-répertoires.

Démo 3 : Filtrage et affichage

Afficher uniquement les noms de fichiers contenant le motif (option -l)

```
grep -l "erreur" *.log
```

Cela affichera uniquement les noms des fichiers contenant "erreur".

Afficher uniquement les lignes qui ne contiennent pas le motif (option -v)

```
grep -v "erreur" fichier.log
```

Cela affichera toutes les lignes qui ne contiennent pas "erreur".

Afficher un nombre de lignes avant et après le motif (options -B et -A)

```
grep -B 3 -A 2 "erreur" fichier.log
```

Cela affichera 3 lignes avant et 2 lignes après chaque occurrence de "erreur".

Afficher un nombre de lignes de contexte autour du motif (option -C)

```
grep -C 3 "erreur" fichier.log
```

Cela affichera 3 lignes avant et après chaque occurrence de "erreur".

Démo 4 : Expressions régulières

Utiliser des expressions régulières étendues (option -E)

```
grep -E "erreur|warning" fichier.log
```

Cela affichera les lignes contenant soit "erreur" soit "warning".

Utiliser des expressions régulières complètes (option -P)

```
grep -P "\berreur\b" fichier.log
```

Cela utilisera les expressions régulières Perl pour trouver "erreur" comme mot entier.

Rechercher des lignes commençant par un motif

```
grep "^Début" fichier.txt
```

Cela trouvera les lignes qui commencent par "Début".

Rechercher des lignes se terminant par un motif

```
grep "Fin$" fichier.txt
```

Cela trouvera les lignes qui se terminent par "Fin".

Démo 5 : Comptage et statistiques

Compter le nombre de lignes correspondant au motif (option -c)

```
grep -c "erreur" fichier.log
```

Cela affichera le nombre de lignes contenant "erreur".

Afficher uniquement le nombre total de correspondances (option -o et wc)

```
grep -o "erreur" fichier.log | wc -l
```

Cela comptera le nombre total d'occurrences de "erreur".

Démo 6 : Divers

Rechercher dans les fichiers binaires (option -a)

```
grep -a "erreur" fichier_binaire
```

Cela traitera les fichiers binaires comme des fichiers texte.

Forcer l'affichage des noms de fichiers (option -H)

```
grep -H "erreur" fichier.log
```

Cela affichera les noms de fichiers même s'il y a un seul fichier dans la recherche.

Ignorer les fichiers et répertoires spécifiés (option --exclude et --exclude-dir)

```
grep -r "erreur" /var/log --exclude "*.old"
```

Cela exclura les fichiers avec l'extension `.old`.

```
grep -r "erreur" /var/log --exclude-dir "archives"
```

Cela exclura le répertoire `archives`.

Limiter la profondeur de la recherche récursive (option --max-depth)

```
grep -r --max-depth=2 "erreur" /var/log
```

Cela recherchera "erreur" jusqu'à une profondeur de 2 niveaux de répertoires.

Conclusion

La commande `grep` est un outil extrêmement puissant pour la recherche de motifs dans des fichiers sous Linux. En combinant les diverses options, vous pouvez affiner vos recherches et extraire des informations précises et pertinentes des fichiers et répertoires.