



# GREP

`$ grep SEARCH FILENAME`

`-n` → afficher le numéro des lignes

`-i` → ne pas respecter la case

`-v` → afficher les lignes qui n'ont pas la recherche

`-r` → récursivité

`-E` → utiliser grep avec une regex

## WC

Compter le nombre de ligne d'un fichier (ligne / mot / octet)  
**\$ wc file.txt**

- l → nombre de lignes
- w → nombre de mots
- c → nombre d'octets
- m → nombre de caractères

# **SORT**

Sort classique → **\$ sort file.txt**

Sort dans un fichier de sortie → **\$ sort -o file\_tries.txt file.txt**

Sort en inverse → **\$ sort -r file.txt**

Sort aléatoire → **\$ sort -R file.txt**

Sort des nombres → **\$ sort -n nombre.txt**

## UNIQ

Supprimer les doublons d'une liste (dans un fichier)

```
$ uniq file.txt           $ uniq file.txt file_filtered.txt
```

-c → nombre d'occurrences des mots

-d → affiche uniquement les doublons

## Rechercher des fichiers sous linux

Avec locate

```
$ sudo updatedb
```

```
$ locate file.txt
```

Avec find

```
$ find -name "file.txt"
```

```
$ find /home/adrien/ -name "file.*"
```

```
$ find /home/adrien/ -name "file.*" -type f
```

```
$ find /home/adrien/ -name "folder" -type d
```

```
$ find /home/adrien/ -name "file.*" -type f -delete
```

```
$ find /home/adrien/ -name "*.txt" -exec chmod 600 {} \;
```

## Exercice 1

Dans votre home perso :

Créer un fichier file.txt avec nano et y écrire une liste de mot (1 mot par ligne)  
Créer un fichier number.txt avec nano et y écrire des nombres (1 nombre par ligne)

Retourner sur la home de root et rechercher les fichiers précédents créer

Compter le nombre de lignes, de mots et de caractère de chaque fichier

Ranger la liste file.txt par ordre alphabétique  
Ranger la liste number.txt du plus petit au plus grand

## **~/.profile & ~/.bashrc**

C'est deux fichiers permettent de gérer l'instanciation de votre utilisateur lors de votre connexion.

~/.profile permet de gérer l'instanciation de script

~/.bashrc permet de gérer vos alias



## Les Alias

```
$ alias ALIAS_NAME='COMMAND'
```

On peut ajouter ces alias au fichier ~/.bashrc puis faire un  
\$ source ~/.bashrc

Exemple d'alias :

```
alias up='sudo apt-get update && sudo apt-get  
upgrade'  
alias tf='tail -f'  
alias +x='chmod +x'  
alias c='clear'  
alias q='exit'
```

```
alias cd..='cd ..'  
alias ..='cd ..'  
alias ...='cd ../..'  
alias ....='cd ../../..'  
alias cd-='cd -'  
  
alias md='mkdir'
```

## Exercice 2

Créer vos propres alias.

Les publier sur un git peut être utile ;)

## Redirection de résultats

Il est possible de rediriger vos résultats dans un fichier.

Création de fichier avec le résultat

`$ commande > result.txt`

Ajout du résultat à la fin d'un fichier

`$ commande >> result.log`

## Redirection de résultats avec erreur

Il est possible de rediriger également les erreurs dans un fichier.

2> : redirection des erreurs dans un fichier (/!\ écrasé si existe)

`$ commande > result.txt 2> error.txt`

2>> : redirection des erreurs à la fin d'un fichier

`$ commande > result.txt 2>> error.txt`

2>&1 : redirection des erreurs dans le même fichier que les résultats

`$ commande > result.txt 2>&1`

## Commande sur plusieurs lignes

Il peut être pratique de pouvoir écrire plusieurs lignes pour une même commandes.

```
( $ cat < file.txt )
```

```
$ commande << FIN
```

```
1
```

```
2
```

```
3
```

```
FIN
```

### Exercice 3

Créez une commande (sur plusieurs ligne) qui permet de ranger une liste de nombre.

Cette liste devra être enregistrée dans un fichier de sortie number\_orderd.txt

## Commande sur plusieurs lignes avec redirection de résultat

Exemple :

```
sort -n << FIN > number_ordererd.txt 2>&1
```

12

14

19

8

FIN