

# sort

- Objet : utiliser la commande sort
- Niveau requis :  
[débutant, avisé](#)
- Commentaires : utilisation en ligne de commandes
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-tester](#)
  - Création par [Hypathie](#) 07/08/2014
  - Testé par <...> le <...>
- Commentaires sur le forum : <https://debian-facile.org/viewtopic.php?pid=96325#p96325> <sup>1)</sup>

## Syntaxe et options courantes

```
sort [options [+POS1] [-POS2]] [fichier...]
```

La commande sort permet :

- d'afficher le contenu de fichiers ;
- de trier les lignes d'un fichier texte, en fonction des colonnes.

Les options utiles:

- **-o** : inscrire la sortie dans un fichier
- **-t** : pour spécifier le séparateur (par défaut ce sont les espaces ou les tabulations qui délimitent les colonnes)
- **-k** : pour spécifier la colonne (à la place de +POS1 -POS2)
- **-n** : pour spécifier qu'il s'agit d'un tri numérique
- **-d** : pour tri dictionnaire
- **-r** : pour un tri inverse (-rn [inverse de numérique] ou -rd [inverse d'alphabétique])
- **-M** : ordre chronologique
- **-h** : comparer des nombres dans un format lisible par un humain (par exemple, 2K 1G)
- **-f** : (ignore-case) convertir les minuscules en majuscule

Voir :

```
man sort
```

pour les autres options ou pour un panorama rapide de ces options :

```
sort --help
```

## Afficher et trier

- Comparaison avec la commande cat :

```
cat sort1.txt
```

[retour de la commande](#)

```
5 1 Amandine
1 7 Bruno
3 2 Caroline
```

```
cat sort2.txt
```

[retour de la commande](#)

```
2 5 Daniel
2 6 Émilien
0 3 Fabrice
0 4 Guillaume
```

Dans chacun de ses deux fichiers, les lignes ne sont pas rangées en fonction des colonnes

```
sort sort1.txt sort2.txt
```

[retour de la commande](#)

```
0 3 Fabrice
0 4 Guillaume
1 7 Bruno
2 5 Daniel
2 6 Émilien
3 2 Caroline
5 1 Amandine
```

Même sans option, les deux fichiers ont été concaténés et leurs lignes rangées par ordre numérique croissant en fonction de la première colonne.

Si des doublons apparaissent dans la première colonne pour deux lignes consécutives, ces deux lignes seront rangées en fonction de leur deuxième colonne.

## Enregistrer le résultat dans un fichier : option -o

- Concaténer “sort1.txt” et “sort2.txt” et inscrire le résultat dans un fichier :

```
sort -o sort3.txt sort1.txt sort2.txt
```

```
cat sort3.txt
```

[retour de la commande](#)

```
0 3 Fabrice
0 4 Guillaume
1 7 Bruno
2 5 Daniel
2 6 Émilien
3 2 Caroline
5 1 Amandine
```

## Trier selon l'ordre alphabétique d'une colonne : option -d

- soit le fichier "alpha-sort.txt" déjà trié par ordre numérique selon les colonnes 1 et 2 pour les doublons :

```
cat alpha-sort.txt
```

[retour de la commande](#)

```
1 2 Francine
2 3 Édith
3 4 Géraldine
4 5 Béatrice
5 6 Christelle
5 7 Dorothée
6 8 Amanda
```

- Pour le trier selon l'ordre alphabétique d'une colonne :

Si on indique le type (numérique ou alphabétique), c'est que le type de la première colonne ne nous intéresse pas.

Par exemple ici, le fichier est rangé selon l'ordre numérique selon ses colonnes 1 et 2.

```
sort -k3d alpha-sort.txt
```

[retour de la commande](#)

```
6 8 Amanda
4 5 Béatrice
5 6 Christelle
2 3 Édith
5 7 Dorothée
```

```
1 2 Francine
3 4 Géraldine
```

l'option -k pour indiquer qu'on va utiliser une colonne spécifique pour le tri;  
3 pour spécifier le n° de colonne  
d pour spécifier le type de tri "dictionnaire" ou ordre alphabétique

## Trier selon l'ordre numérique d'une colonne

- soit le fichier "sort.txt" :

```
cat sort.txt
```

[retour de la commande](#)

```
pomme 2
fraise 6
abricot 9
kiwi 3
```

- Pour ordonner ce fichier selon l'ordre numérique de la deuxième colonne :

```
sort -k2n sort.txt
```

[retour de la commande](#)

```
pomme 2
kiwi 3
fraise 6
abricot 9
```

Si on avait voulu ranger ce fichier selon l'ordre alphabétique, il aurait suffi de faire :

```
sort sort.txt
```

Il aurait été inutile d'indiquer la colonne car la première colonne comporte des chaînes de caractères.

Comme vous l'aurez compris, par défaut, sort trie en fonction de la colonne n°1 selon le type qui est le sien, puis s'il le faut, en fonction de la colonne n° 2 et du type qui est celui de cette deuxième colonne.

En fait, sed s'adapte automatiquement au type du contenu de la première colonne, voire au type du contenu de la seconde colonne pour départager éventuellement les doublons de cette première colonne, même si les types diffèrent d'une colonne à l'autre.

## Tri par défaut d'un fichier dont les colonnes présentent différents types

- Soit fichier "xxx-sort.txt" désordonné selon ses types :

```
cat xxx-sort.txt
```

[retour de la commande](#)

```
10 fraise
pomme 6
Titi grosminet
25 8
```

- Par défaut, sort trie selon l'ordre numérique d'abord puis par ordre alphabétique :

```
sort xxx-sort.txt
```

[retour de la commande](#)

```
10 fraise
25 8
pomme 6
Titi grosminet
```

- de même pour l'autre colonne :

```
sort -k2 xxx-sort.txt
```

[retour de la commande](#)

```
pomme 6
25 8
10 fraise
Titi grosminet
```

## Trier selon plusieurs colonnes et différents types

- Soit le fichier "sort4.txt" qui est un extrait du fichier **/etc/passwd** dont j'ai pris quelques lignes que ai interverties pour cet exemple :

```
cat sort4.txt
```

[retour de la commande](#)

```
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
root:x:0:0:root:/root:/bin/bash
man:x:6:12:man:/var/cache/man:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

- Pour faire un tri selon le shell des utilisateurs (7ième colonnes : type “dictionnaire”) et en cas de doublon pour cette colonne, en fonction ensuite de l'UID (3ième colonne : type “numérique”)

```
sort -t':' -k 7d -k 3n sort4.txt
```

**-t':'** pour spécifier le séparateur : (par défaut c'est l'espace ou la tabulation, il faut donc indiquer le séparateur pour les fichiers de >type.)

**-k** pour indiquer les colonnes souhaitées

**7d** pour 7ième colonne type “dictionnaire”

**3n** pour 3ième colonne type “numérique”

[retour de la commande](#)

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
```

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:  
<http://debian-facile.org/> - Documentation - Wiki

Permanent link:  
<http://debian-facile.org/doc:systeme:sort>

Last update: **03/10/2015 13:41**



