

# D'autres commandes pour la gestion des fichiers-texte

- Objet : Voici d'autres commandes pour la gestion de fichiers texte
- Niveau requis :  
[débutant](#), [avisé](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-tester](#)
  - Création par [Hypathie](#) le 07/06/2014
  - Testé par <...> le <...>
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) <sup>1)</sup>

## Introduction

Pour les commandes principales de gestion des fichiers :

- [Les commandes Linux de gestion des fichiers et répertoires](#)

Nous allons aborder quelques commandes complémentaires permettant de :

- améliorer l'affichage : `fmt` ; `nl`
- combiner le contenu de fichiers : `join` ; `paste`
- transformer le contenu de fichiers : `sort` ; `split` ; `tr` ; `uniq` ; `expand` et `unexpand`

## la commande `fmt`

### Syntaxe

```
fmt [options] [fichier...]
```

### Préparation

Création du fichier “~/Essai-text-processing/essai-fmt”

```
cat > ~/Essai-text-processing/essai-fmt <<EOF
> chez debian-facile c'est facile ... très facile ... très très facile ...
très très très très très très facile ...
facile et encore facile !
> EOF
```

## Affichage du contenu d'un fichier en respectant les mots

- Affichage avec la commande cat

```
cat ~/Essai-text-processing/essai-fmt
```

[retour de la commande](#)

```
chez debian-facile c'est facile ... très facile ... très très facile
... très tr
ès très très très très facile ... facile et encore facile !
```

L'affichage avec la commande cat peut être difficile à lire.

- Affichage avec la commande fmt :

```
fmt ~/Essai-text-processing/essai-fmt
```

[retour de la commande](#)

```
chez debian-facile c'est facile ... très facile ... très très facile
... très très très très très très facile ... facile et encore
facile !
```

- Par défaut, la commande fmt affiche le contenu d'un fichier par ligne de 75 caractères maximum.
- L'option `-w` : pour indiquer le nombre de caractère par ligne

```
fmt -w 20 ~/Essai-text-processing/essai-fmt
```

[retour de la commande](#)

```
chez debian-facile
c'est facile
... très facile
... très très
facile ... très
très très très
très très facile
... facile et
encore facile !
```

## Combiner `fmt` et `less` pour encore plus de lisibilité

```
fmt -w 20 ~/Essai-text-processing/essai-fmt | less
```

[retour de la commande](#)

```
chez debian-facile
c'est facile
... très facile
... très très
facile ... très
très très très
très très facile
... facile et
encore facile !
(END)
```

- Voir [la commande less](#)

## La commande `nl`

Permet de numéroter l'affichage d'un fichier.

### Syntaxe

```
nl [options] [fichier...]
```

### Préparation

- copie du fichier `~/.bashrc` et visualisation :

```
cat ~/.bashrc >> ~/Essai-text-processing/bashrc-bis
```

- comparons avec l'affichage de `less`

```
less ~/Essai-text-processing/bashrc-bis
```

[retour de la commande](#)

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-
# doc)
# for examples

# If not running interactively, don't do anything
```

```
case $- in
  *i*) ;;
  *) return;;
esac
```

Premières lignes du retour.

## Affichage numéroté

```
nl ~/Essai-text-processing/bashrc-bis
```

[retour de la commande](#)

```
1 # ~/.bashrc: executed by bash(1) for non-login shells.
2 # see /usr/share/doc/bash/examples/startup-files (in the package
bash-doc)
3 # for examples

4 # If not running interactively, don't do anything
5 case $- in
6     *i*) ;;
7     *) return;;
8 esac
```

Premières lignes du retour

## Pour encore plus de lisibilité

```
fmt -w 65 ~/Essai-text-processing/bashrc-bis | nl ~/Essai-text-
processing/bashrc-bis | less
```

[retour de la commande](#)

```
1 # ~/.bashrc: executed by bash(1) for non-login shells.
2 # see /usr/share/doc/bash/examples/startup-files (in the
package bash-doc)
3 # for examples

4 # If not running interactively, don't do anything
5 case $- in
6     *i*) ;;
7     *) return;;
8 esac
```

## Premières lignes du retour

Par défaut la commande nl numérote les lignes non-vides.

- Quelques options :
  1. -b : numérotation des lignes vides
  2. -b -t[expression régulière] : affiche tout mais numérote les lignes qui correspondent à l'expression régulière
  3. -n ln : affichage avec numérotation tout à droit.
  4. -n rn : affichage par défaut, c'est-à-dire avec une tabulation à droite avec le chiffre de la numérotation.
  5. -n rz : affichage de zéro entre le bord droit et le chiffre de numérotation
- voir man nl pour plus de meilleures explications.

## La commande join

### Syntaxe de la commande join

```
join [options]... fichier1 fichier2
```

### Préparation

Soit deux fichiers : "users" et "infos-users"

```
cat ~/Essai-text-processing/users
```

[retour de la commande](#)

```
01 Toto  
02 Titi  
03 Lili
```

```
cat ~/Essai-text-processing/infos-users
```

[retour de la commande](#)

```
toto@gmail.com 01  
titi@gmx.fr 02  
lili@hotmail.com 03
```

### Fusion de ces deux fichiers

```
join -1 1 -2 2 ~/Essai-text-processing/users ~/Essai-text-processing/infos-
```

## users

[retour de la commande](#)

```
01 Toto toto@gmail.com
02 Titi titi@gmx.fr
03 Lili lili@hotmail.com
```

- Explications :
  - l'option -1 : donne le n° de colonne du champ commun, pour le premier document
  - 1 : premier fichier
  - l'option -2 : n° de colonne du champ commun, pour le deuxième document
  - 2 : deuxième fichier
- D'autres options utiles :
  - -j (option CHAMP) : équivalent à « -1 CHAMP -2 CHAMP »
  - -i : ignorer les différences de casse
- voir man join pour plus d'options

Pour fusionner des fichiers qui ont en commun :

```
join -t':' -2 1 -2 2 fichier1 fichier2
```

### Concaténation de la fusion dans un fichier

```
join -1 1 -2 2 ~/Essai-text-processing/users ~/Essai-text-processing/infos-users > ~/Essai-text-processing/adr-users
```

### Vérification

```
cat ~/Essai-text-processing/adr-users
```

[retour de la commande](#)

```
01 Toto toto@gmail.com
02 Titi titi@gmx.fr
03 Lili lili@hotmail.com
```

## La commande paste

### Syntaxe de la commande paste

```
paste [options]... [fichier]...
```

La jointure se fait selon lignes à ligne, donc il faut des fichiers avec des lignes rangées dans le même ordre.

## Préparation

Visualisons les deux fichiers concernés, "users" et "infos-users"

```
cat ~/Essai-text-processing/users
```

[retour de la commande](#)

```
01 Toto
02 Titi
03 Lili
```

```
cat ~/Essai-text-processing/infos-users
```

[retour de la commande](#)

```
toto@gmail.com 01
titi@gmx.fr 02
lili@hotmail.com 03
```

## Regroupement des lignes des deux fichiers

```
paste ~/Essai-text-processing/users ~/Essai-text-processing/infos-users
```

[retour de la commande](#)

```
01 Toto toto@gmail.com 01
02 Titi titi@gmx.fr 02
03 Lili lili@hotmail.com 03
```

## Concaténation du retour dans un fichier "paste-users-infos" qui sera créé

```
paste ~/Essai-text-processing/users ~/Essai-text-processing/infos-users >
~/Essai-text-processing/paste-users-infos
```

## Vérification

[retour de la commande](#)

```
01 Toto toto@gmail.com 01
02 Titi titi@gmx.fr 02
03 Lili lili@hotmail.com 03
```

## la commande tr

Elle permet de traduire, c'est-à-dire, changer des lettres

### Syntaxe

```
tr [options] ens1 ens2
```

### Exemples

```
cat ~/Essai-text-processing/tr-essai
```

[retour de la commande](#)

```
Pierre Paul Jacques
```

```
tr P p < ~/Essai-text-processing/tr-essai
```

[retour de la commande](#)

```
pierre paul Jacques
```

```
cat ~/Essai-text-processing/tr-essai
```

[retour de la commande](#)

```
Pierre Paul Jacques
```

```
tr P X < ~/Essai-text-processing/tr-essai
```



[retour de la commande](#)

```
Xierre Xaul Jacques
```

```
cat ~/Essai-text-processing/tr-essai
```

[retour de la commande](#)

```
Pierre Paul Jacques
```

```
tr PJ pj < ~/Essai-text-processing/tr-essai
```

[retour de la commande](#)

```
pierre paul jacques
```

## La commande uniq

Elle permet d'éliminer les lignes dupliquées d'un fichier trié.

### Syntaxe

```
uniq [options] [fichier-entrée] [fichier-sortie]
```

### Préparation

```
cat ~/Essai-text-processing/users-uniq
```

[retour de la commande](#)

```
Papi  
1 Toto  
3 Lili  
2 Titi  
4 Mami  
003 Zaza  
01 Nani
```

## Ajouter une ligne

```
echo 01 Nani >> ~/Essai-text-processing/users-uniq
```

## Supprimer les doublons

```
cat ~/Essai-text-processing/users-uniq
```

[retour de la commande](#)

```
5 Papi
1 Toto
3 Lili
2 Titi
4 Mami
003 Zaza
01 Nani
01 Nani
```

Il faut que les doublons soient consécutifs, et que le fichier soit “trié”

```
sort ~/Essai-text-processing/users-uniq | uniq > ~/Essai-text-processing/users-sort-uniq
```

## Vérification

```
cat ~/Essai-text-processing/users-sort-uniq
```

[retour de la commande](#)

```
003 Zaza
01 Nani
1 Toto
2 Titi
3 Lili
4 Mami
5 Papi
```

## La commande split

## Syntaxe

```
split [options] [fichier_entrée [préfixe_sortie]]
```

## Préparation

- Création du fichier “~/Essai-text-processing/liste-fruits” :

```
cat > ~/Essai-text-processing/liste-fruits <<EOF
> 1 fraise
> 2 abricot
> 3 pomme
> EOF
```

- Vérification :

```
cat ~/Essai-text-processing/liste-fruits
```

[retour de la commande](#)

```
1 fraise
2 abricot
3 pomme
```

## Découpage par taille et répartition en autant de fichiers

- options de découpage par taille :

-C : découper par taille maximale

ex: split -C 5 fichier : découpage en 5 octets

```
( ou de même split -C 5b)\
```

```
split -C 5k fichier : découpage en 5 kilo-octets\
```

```
split -C 5m fichier : découpage en 5 mega-octets\
```

-b : découper par 1 taille précise

Ex: split -b 5 fichier) ⇒ tous les fichiers créés par découpage auront cette taille (sauf peut-être le dernier)

- Remarque:

Il faut 8 bits pour faire 1 byte (1 octet)

1. bit : c'est **B**inary **D**igi**T** : un bit est un chiffre binaire, la valeur d'un bit, c'est 0 ou 1.
2. byte : terme anglais pour octet. L'ordinaire calcule sur un ou plusieurs byte.

- Exemple de découpage par taille d'un octet avec le préfixe fruit1b :

```
split -C 1b ~/Essai-text-processing/liste-fruits fruit1b
```

Le fichier n'est pas lourd et le découpage en taille de 1 byte (=octet) ne le découpe pas :

```
ls fruit1b*
```

[retour de la commande](#)

```
fruit1baa
```

Un seul fichier a été créé.

```
cat fruit1b*
```

[retour de la commande](#)

```
1 fraise
2 abricot
3 pomme
```

Le fichier est créé dans le répertoire dans lequel le terminal est ouvert.

Pour que le fichier créé, lors du découpage, se retrouve dans le répertoire du fichier à découper :

```
split -C 1b ~/Essai-text-processing/liste-fruits fruit1b && mv fruit1b*
~/Essai-text-processing/
```

```
ls ~/Essai-text-processing/fruit1b*
```

[retour de la commande](#)

```
/home/hypathie/Essai-text-processing/fruit1baa
```

- Ou encore :

```
cd Essai-text-processing/
```

```
split -C 1b ~/Essai-text-processing/liste-fruits fruit1b
```

```
ls fruit1b*
```

[retour de la commande](#)

```
fruitlbaa
```

## Découpage par ligne

- option de découpage par ligne qui détermine le découpage :
  1. l : pour indiquer le nombre de ligne
- Remarque :

Sans indication de préfixe, par défaut les fichiers créés par le découpage porte le préfixe x??

```
cd Essai-text-processing/
```

```
split -l 1 ~/Essai-text-processing/liste-fruits
```

```
ls x*
```

[retour de la commande](#)

```
xaa xab xac
```

```
cat x*
```

[retour de la commande](#)

```
1 fraise  
2 abricot  
3 pomme
```

## La commande unexpand

La commande "unexpand" peut servir à convertir un groupe de sept espaces en une tabulation.

### Syntaxe

```
unexpand [-a] [fichier...]
```

## Préparation

- Création du fichier “~/Essai-text-processing/users-infos-espaces”
- Avec 7 ou plus espaces pour ligne 1 et 3
- Avec moins de 7 espaces pour ligne 2

```
cat > ~/Essai-text-processing/users-infos-espaces<<EOF
>      01      Toto      toto@gmx.fr      modo
>      02      Titi      titi@gmail.com      membre
>      03      Lili      lili@hotmail.com      membre
> EOF
```

Ci-dessus ce sont plusieurs espaces qui sont insérés entre les chaînes de caractères. Pour intégrer le contenu d'un fichier avec les redirections et EOF, la touche tabulation ne fonctionne pas.

## Convertir des espaces en tabulation

### Affichage avec cat : ce sont des espaces

```
cat ~/Essai-text-processing/users-infos-espaces
```

[retour de la commande](#)

```
      01      Toto      toto@gmx.fr      modo
02      Titi      titi@gmail.com      membre
      03      Lili      lili@hotmail.com      membre
```

## Conversion des espaces en tabulation

```
unexpand -a ~/Essai-text-processing/users-infos-espaces
```

[retour de la commande](#)

```
01      Toto      toto@gmx.fr      modo
 02      Titi      titi@gmail.com      membre
03      Lili      lili@hotmail.com      membre
```

- Option -a pour que la conversion des espaces ait lieu partout et pas seulement pour les espaces constituant la première colonne de droite.
- Les groupes de moins de sept espaces consécutifs n'ont pas été convertis en tabulation.

## La commande "expand"

La commande "expand" peut servir à convertir les tabulations en espaces.

### Syntaxe

```
expand [-t numero] [fichier...]
```

### Préparation

Soit le fichier "~/Essai-text-processing/users-infos-tabul" contenant des tabulations

```
unexpand -a ~/Essai-text-processing/users-infos-espaces > ~/Essai-text-processing/users-infos-tabul
```

### Visualisation

```
cat ~/Essai-text-processing/users-infos-tabul
```

[retour de la commande](#)

```
01      Toto      toto@gmx.fr      modo
      02      Titi      titi@gmail.com      membre
      03          Lili      lili@hotmail.com      membre
```

### Conversion des tabulations en espaces

```
expand ~/Essai-text-processing/users-infos-tabul
```

[retour de la commande](#)

```
01      Toto      toto@gmx.fr      modo
02      Titi      titi@gmail.com      membre
03          Lili      lili@hotmail.com      membre
```

### Déterminer le nombre d'espaces qui remplaceront chaque tabulation

- option -t nombre

```
expand -t 1 ~/Essai-text-processing/users-infos-tabul
```

[retour de la commande](#)

```
01 Toto toto@gmx.fr modo
   02 Titi titi@gmail.com membre
   03 Lili lili@hotmail.com membre
```

## Aller plus loin avec les commandes `expand` et `unexpand`

Voir : [expand](#)

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:commandes:les-commandes-join-paste-split-et-nl>



Last update: **06/11/2015 17:57**