

Les commandes relatives aux fichiers

- Identifier le type d'un fichier : **file**
- Convertir l'encodage : **iconv**
- Visualiser le contenu d'un fichier :
 - en entier : **cat**
 - page par page : **more**
 - le début : **head**
 - la fin : **tail**
- Compter les mots et les lignes d'un fichier : **wc**
- Rechercher les lignes multiples dans un fichier : **uniq**
- Chercher une chaîne de caractères dans un fichier : **grep**
- Trier les champs d'un fichier : **sort**
- Extraire des champs d'un fichier : **cut**
- Ajouter des champs à un fichier : **paste**
- Copier un fichier : **cp**
- Créer un lien sur un fichier : **ln**
- Déplacer ou changer le nom d'un fichier : **mv**
- Découper un fichier : **split**
- Supprimer un fichier : **rm**
- Droits d'accès aux fichiers : **chmod**
- Comparer 2 fichiers :
 - Rechercher les lignes identiques : **comm**
 - Rechercher les lignes différentes : **diff**
- Concaténer 2 fichiers : **cat**
- Imprimer un fichier : voir Impressions et imprimantes

file

Identifie le type d'un fichier

Syntaxe : **file *fichier***

Exemple :

file .scilab : renvoie `commands text`

Les types **text** peuvent être transformés en PostScript pour impression au moyen de l'utilitaire `a2ps`.

Il existe tellement de types de fichiers différents que nous ne voyons ici que les plus courants. Il s'agit des types :

- `script text executable`
- `ASCII C program text`
- `ASCII C++ program text`
- `ASCII English text`
- `ASCII fortran program text`

- ASCII HTML document text
- ASCII Java program text
- ASCII mail text
- ASCII news text
- ASCII Pascal program text
- ASCII text
- Bourne shell script text
- ISO-8859 C program text
- ISO-8859 C++ program text
- ISO-8859 English text
- ISO-8859 HTML document text
- ISO-8859 Java program text
- ISO-8859 mail text
- ISO-8859 Pascal program text
- ISO-8859 text
- script text executable
- UTF-8 Unicode C program text
- UTF-8 Unicode C++ program text
- UTF-8 Unicode English text
- UTF-8 Unicode Java program text
- UTF-8 Unicode mail text
- UTF-8 Unicode text

Il existe des fichiers archivés, compressés ou codés qui peuvent se ramener à des fichiers textes. Ce sont les types :

- bzip2 compressed data : fichier compressé par l'utilitaire bzip2.
- gzip compressed data : fichier compressé par l'utilitaire gzip.
- POSIX tar archive : fichier archivé par tar.
- USTAR tar archive : fichier archivé par tar.
- ZIP archive : fichiers MS-DOS compressés. Peut être décompressé par l'utilitaire unzip.

Les types suivants correspondent à des fichiers non imprimables. Ce peuvent être des binaires tels :

- audio data
- ELF 32-bit LSB executable
- ELF 32-bit MSB dynamic lib SPARC
- ELF 32-bit MSB executable SPARC
- ELF 32-bit MSB relocatable SPARC

des images :

- GIF file
- GIF image data
- JPEG file
- JPEG image data
- PNG file
- PNG image data
- PNG file
- TIFF file

Enfin lorsque `file` ne reconnaît pas le type d'un fichier, il affiche :

- `data`
-

iconv

Convertir l'encodage

Syntaxe : **iconv -f *oldcode* -t *newcode* *entree* -o *sortie***

Les versions anciennes de linux utilisaient l'encodage ISOLATIN-1 (*ou ISO 8859-1 ou ISO 8859-15*). Depuis la distribution RedHat 9, l'encodage utilisé est l'Universal Character Set (UCS) qui contient les caractères nécessaires pour représenter presque tous les langages connus. Il inclut les alphabets latin, grec, cyrillique, hébreu, arabe, arménien, chinois, japonais, coréen, bengali, thai, tibétain, mongol, ...

Exemple :

iconv -f latin1 -t utf8 test.iso -o test.unic : convertit le fichier `test.iso` en `test.unic` de l'encodage iso-latin en utf-8.

cat

Visualise le contenu d'un fichier en entier.

Syntaxe : **cat [-n] *fichier***

Option principale : **-n** : numérote les lignes

NB : Il existe une autre utilisation de la commande `cat`

Exemples :

cat -n *essai.1* : affiche le fichier `essai.1` à l'écran en numérotant les lignes.

cat *essai.1* *essai.2* : affiche à l'écran le fichier `essai.1` puis le fichier `essai.2`.

more

Visualise le contenu d'un fichier page par page.

Syntaxe : **more [-c] *fichier***

Option principale : **-c** : efface la fenêtre avant l'affichage.

Exemple :

more *essai.c* : affiche la première page du fichier `essai.c`. Puis tapez :

- **espace** pour afficher la **page** suivante.
 - [**Enter**] pour afficher la **ligne** suivante.
 - **h** pour accéder à l'aide en ligne.
 - **q** pour quitter.
-

head

Affiche le début d'un fichier.

Syntaxe : **head [-num] *fichier***

Option principale : **-num** : affiche les *num* premières lignes du fichier. Par défaut, num = 10.

Exemple :

head -10 *.c : affiche les 10 premières lignes de tous les fichiers sources C du répertoire courant.

tail

Affiche la fin d'un fichier.

Syntaxe : **tail [-num] *fichier***

Option principale : **-num** : affiche les *num* dernières lignes du fichier. Par défaut, num = 10.

Exemple :

tail -50 *essai* : affiche les 50 dernières lignes du fichier *essai*.

wc

Compte le nombre de lignes, de mots ou de caractères d'un fichier.

Syntaxe : **wc [-clw] *fichier***

Options principales :

- **-c** : renvoie le nombre de caractères.
- **-l** : renvoie le nombre de lignes.
- **-w** : renvoie le nombre de mots.

Exemple :

wc -l *essai.c* : renvoie le nombre de lignes du fichier *essai.c*.

uniq

Recherche les lignes consécutives identiques dans un fichier.

Syntaxe : **uniq [-udc] [+m] [-n] *fichier***

Options principales :

- **-u** : affiche les lignes consécutives identiques une seule fois.
- **-d** : n'affiche que les lignes consécutives identiques.
- **-c** : chaque ligne est précédée de son nombre d'occurrences.
- **+m** : saute les *m* premiers caractères de chaque ligne.
- **-n** : saute les *n* premiers champs non blancs.

Exemple : Prenons le fichier *noms* :

```
alain
jean
maurice
alain
robert
jean
```

Après l'avoir trié en tapant : **sort noms -o nomstries** , la commande :
uniq -c nomstries renvoie

```
2 alain
2 jean
1 maurice
1 robert
```

grep

Recherche une expression dans un fichier.

Syntaxe : **grep [-cilmv] *expression fichier***

Options principales :

- **-c** : affiche le nombre de lignes contenant l'expression.
- **-i** : contrairement à la plupart des commandes, l'option **-i** ne signifie pas interactif mais *ignore-case*, c'est à dire que grep ne fera pas de différence entre minuscules et majuscules.
- **-l** : n'affiche que le nom des fichiers contenant l'expression.
- **-n** : affiche les numéros des lignes contenant l'expression.
- **-v** : inverse la recherche, c'est à dire affiche les lignes ne contenant pas l'expression.

Exemple :

grep -i -c 'integer' * : affiche le nombre de fois où les expressions *integer*, *Integer* ou *INTEGER* sont rencontrées dans chaque fichier du répertoire courant.

sort

Trie les champs d'un fichier.

Syntaxe : **sort [-fnru] [-t *car*] [+i] [-j] [-o *sortie*] *fichier***

Options principales :

- **-f** : ne tient pas compte des minuscules et majuscules.
- **-n** : trie numérique et non alphabétique.
- **-r** : inverse l'ordre de tri.
- **-u** : n'affiche qu'une fois les lignes multiples.
- **-t *car*** : indique le caractère séparateur de champs. (*Par défaut, c'est l'espace*).
- **-o *sortie*** : écrit le résultat dans le fichier *sortie*.
- **+i** : on trie à partir du *i*-ième champ.
- **+j** : on trie jusqu'au *j*-ième champ.

Le **premier** champ est noté **0**, le **second 1** ,...

Exemples :

Le fichier *liste* contient une liste de noms, prénoms de la forme :

```
Alex TERIEUR
Alain VERSE
Jean NEMARD
Robert BIDOCHON
```

sort +1 *liste* : trie le fichier *liste* par ordre alphabétique des noms.

Le fichier *adresses* contient une liste de prénoms,numéros de téléphone et villes de la forme suivante (*les champs sont séparés par :*) :

```
alain:0298123456:Quimper
alex:0466789012:Marseille
jean:0144567890:Paris
robert:0380234567:Dijon
```

sort -n -t : +1 -2 *adresses* : trie le fichier *adresses* par numéros de téléphones.

cut

Extrait des champs d'un fichier.

Syntaxe : **cut [-cf] [-d *car*] [+i] [-j] *fichier***

Options principales :

- **-c** : extrait suivant le nombre de caractères.
- **-f** : extrait suivant le nombre de champs.
- **-d car** : indique le caractère séparateur de champs. (*Par défaut, c'est la tabulation*).
- **+i** : on trie à partir du *i-ième* champ.
- **+j** : on trie jusqu'au *j-ième* champ.

Contrairement à `sort`, le **premier** champ est noté **1**, le **second** **2**, ...

Exemples:

En reprenant le fichier *adresses* (*Voir sort*), la commande :

cut -c10 adresses : extrait les 10 premiers caractères de chaque ligne, soit :

```
alain:0298
alex:04667
jean:01445
robert:038
```

cut -d : -f2 adresses : extrait le deuxième champ, soit les numéros de téléphones :

```
0298123456
0466789012
0144567890
0380234567
```

paste

Ajoute des champs à un fichier

Syntaxe : **paste [-s] [-d car] fichier1 fichier2**

Options principales :

- **-s** : les lignes sont remplacées par des colonnes
- **-d car** : indique le caractère séparateur de champs. (*Par défaut, c'est la tabulation*).

Exemple :

Reprenons le fichier *adresses* (*Voir sort*) et considérons le fichier *professions* :

```
boucher
journaliste
instituteur
mécanicien
```

la commande **paste -d : adresses professions > carnet** crée le fichier *carnet* contenant :

```
alain:0298123456:Quimper:boucher
alex:0466789012:Marseille:journaliste
jean:0144567890:Paris:instituteur
robert:0380234567:Dijon:mécanicien
```

cp

Copie de fichier.

Syntaxe : **cp [-ipr] source cible**

Pour copier plusieurs fichiers d'un seul coup, il faut utiliser les caractères spéciaux du shell.

Option principale : **-p** : le fichier cible conserve les permissions et la date de dernière modification du fichier source (*au lieu de la date courante*).

Exemples:

cp tp1/* tp2 : copie tous les fichiers du répertoire *tp1* dans le répertoire *tp2*

cp -ir tp1 tp2 : copie le répertoire *tp1* dans le répertoire *tp2* en demandant confirmation à chaque fichier.

ln

Crée un lien sur un fichier.

Syntaxe : **ln [-s] source cible**

Option principale : **-s** : crée un lien *symbolique*.

Créer un lien revient à définir un *alias* à un fichier qui existe déjà. Un lien ne peut se faire qu'entre fichiers d'un même filesystem. Sinon il faut utiliser l'option **-s**.

De manière générale, il est **recommandé** d'utiliser systématiquement l'option **-s**.

Exemple :

ln -s /export/home/pub/stage/unix/pg.c pg.c : crée un lien symbolique. Pratiquement, c'est comme si le fichier *pg.c* se trouvait dans le répertoire courant mais sans avoir à en faire la copie.

mv

Déplace ou change le nom d'un fichier

Syntaxe : **mv [-fi] source cible**

Si *cible* est un nom de **fichier**, *source* est **renommé** en *cible*.

Si *cible* est un nom de **répertoire**, *source* est **déplacé** dans le répertoire *cible*.

Option principale : **-f** : force la commande.

Exemples :

mv essai.1 essai.2 : renomme le fichier *essai.1* en *essai.2*

mv tp11.c tp/tp1 : déplace le fichier *tp11.c* dans le répertoire *tp/tp1*

split

Découpe un fichier.

Syntaxe : **split** [-num] *fichier* [*nom*]

Options principales :

- **-num** : découpe *fichier* en fichiers de *num* lignes. (*Par défaut, 1000 lignes.*)
- **nom** : par défaut les fichiers créés sont nommés *xaa,xab ,xac,...* jusqu'à *xzz*. Avec cette option, on remplace *x* par *nom* : on obtient donc les fichiers *nomaa, nomab....*

Exemple :

split -100 essai.c petit : découpe le fichier *essai.c* en fichiers de 100 lignes nommés *petitaa, petitab,...*

Pour les utilisateurs de FORTRAN, il existe la commande **fsplit** qui découpe un fichier source en autant de fichiers qu'il y a de fonctions ou sous-routines dans le fichier principal.

rm

Supprime un fichier ou répertoire.

Syntaxe : **rm** [-if] *fichier* ou **rm** [-if] **-r** *répertoire*

Option principale : **-f** : force la commande.

Pour les utilisateurs de MS-DOS : Unix étant multitâche, il n'existe pas de commande analogue à UNDELETE. Soyez donc très prudents dans l'utilisation de cette commande.

Exemples :

rm essai.2 : suppression du fichier *essai2*

rm -r tp : suppression du répertoire *tp*

chmod

Droits d'accès aux fichiers

NB : On ne peut modifier les droits d'accès à un ou à plusieurs fichiers ou répertoires que si l'on est propriétaire.

Syntaxe : **chmod** [-R] [*ugoa*] [+/-r] [+/-w] [+/-x] *fichier*

- Type d'utilisateur :
 - **u** : propriétaire (*user*)
 - **g** : groupe du propriétaire (*group*)
 - **o** : les autres (*other*)
 - **a** : tous (*all*)

- Type d'action :
 - **+** signifie qu'on ajoute des droits d'accès,
 - **-** signifie qu'on enlève des droits d'accès..

- Type d'accès :
 - **r** : lecture (*read permission*)
 - **w** : écriture (*write permission*)
 - **x** :exécution (*execute permission*)

Exemple :

chmod -R g+r tp : donne les accès en lecture au sous-répertoire *tp* à tous les membres du groupe.

comm

Recherche les lignes identiques à 2 fichiers.

Syntaxe : **comm [-123] fichier1 fichier2**

comm sépare les lignes en 3 groupes :

1. les lignes présentes seulement dans le *premier*fichier.
2. les lignes présentes seulement dans le *deuxième*fichier.
3. les lignes présentes seulement dans les 2 fichiers.

Options :

- **-1** : n'affiche pas les lignes du *premier*groupe.
- **-2** : n'affiche pas les lignes du *deuxième*groupe.
- **-3** : n'affiche pas les lignes du *troisième*groupe.

Exemple :

Soit le fichier *adresses2* suivant :

```
alain:0298662544:Brest
alex:0466789012:Marseille
jean:0144567890:Paris
robert:0380234567:Dijon
```

comm **adresses** **adresses2** le compare au fichier *adresses* (*Voir sort*), et renvoie :

```
alain:0298123456:Quimper
alain:0298662544:Brest
alex:0466789012:Marseille
jean:0144567890:Paris
robert:0380234567:Di jon
```

On voit que la *première* ligne est différente.

diff

Recherche les lignes différentes à 2 fichiers.

Syntaxe : **diff** *fichier1* *fichier2*

Options principales :

- **-b** : ignore les espaces.
- **-i** : contrairement à la plupart des commandes, l'option **-i** ne signifie pas **interactif** mais *ignore-case*, c'est à dire que diff ne fera pas de différence entre minuscules et majuscules.
- **-w** : ignore les espaces et les tabulations.

Exemples :

Reprenons les fichiers *adresses* et *adresses2* (*Voir comp*).

diff **adresses2** **adresses** renvoie :

```
1c1
< alain:0298123456:Quimper
---
> alain:0298662544:Brest
```

On voit que la *première* ligne est différente.

cat

Concaténer 2 fichiers (ou plus)

Syntaxe : **cat** *fichier1* *fichier2* ... > *fichierx*