

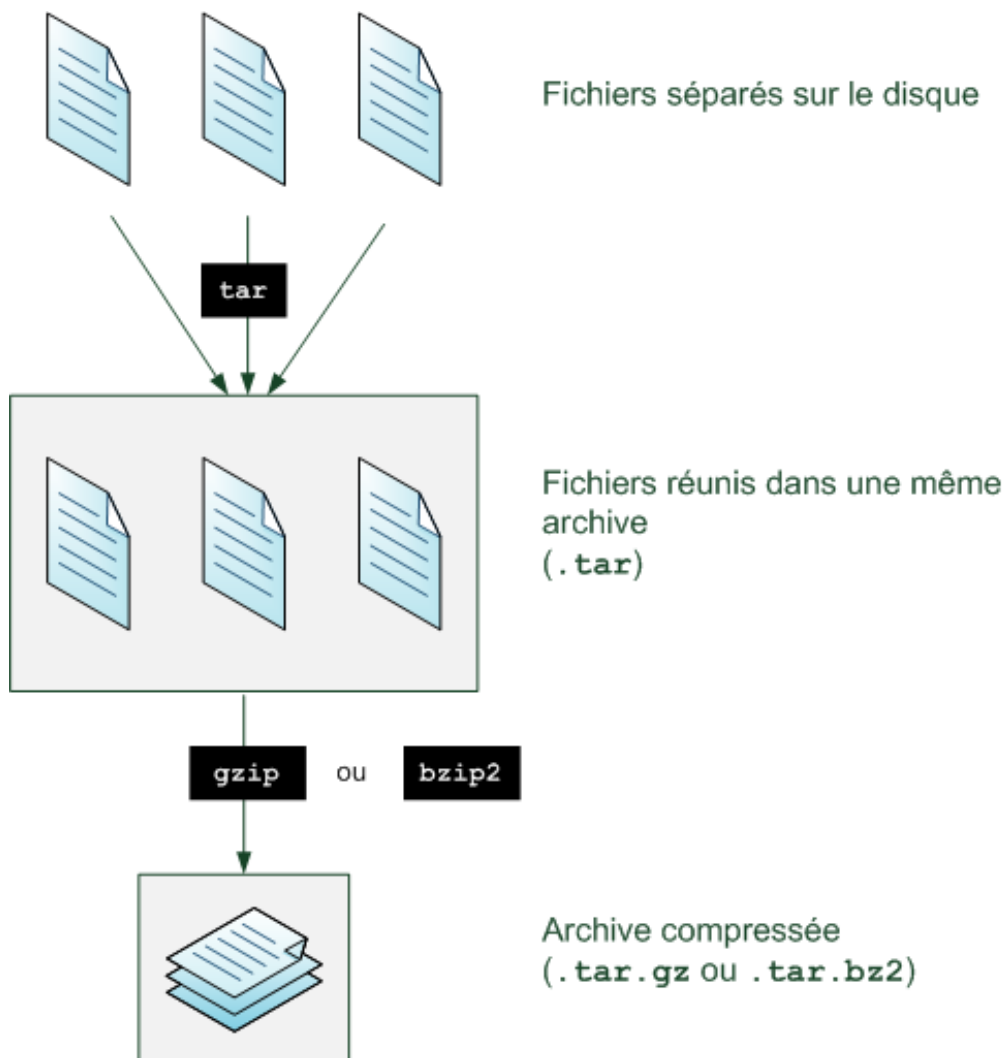
## tar : assembler des fichiers dans une archive

Comme je vous le disais en introduction, aussi étonnant que cela puisse paraître, les `gzip` et `bzip2` ne permettent de compresser qu'un seul fichier à la fois. Comment faire alors si vous voulez compresser une dizaine de fichiers ?

Sous Linux, on a depuis longtemps pris l'habitude de procéder en deux étapes :

1. réunir les fichiers dans un seul gros fichier appelé **archive**. On utilise pour cela le programme `tar` ;
2. compresser le gros fichier ainsi obtenu à l'aide de `gzip` ou de `bzip2`.

Ces deux étapes sont résumées dans le schéma de la figure suivante.



Nous allons dans un premier temps apprendre à manipuler `tar`, puis nous verrons la compression avec `gzip` et `bzip2`, sans oublier les formats `zip` et `rar` que vos amis utilisant Windows risquent de vous envoyer un jour ou l'autre.

Les formats `zip` et `rar` ne séparent pas les étapes comme sur le schéma suivant. Ils sont capables d'assembler plusieurs fichiers en une archive et de la compresser en même temps.

Comme vous le voyez, sous Linux il y a donc une méthode à suivre dans un ordre précis. Voyons ensemble comment faire !

## Regrouper d'abord les fichiers dans un même dossier

Vous avez plusieurs fichiers que vous souhaitez compresser. Dans mon cas, ce sont des fichiers `.tuto` (qui contiennent des chapitres de tutoriels du Site du Zéro), mais vous pouvez bien entendu compresser ce que vous voulez : des textes, présentations, tableurs, *logs*, etc.

Certains formats de fichier sont déjà compressés. C'est le cas des images `jpeg`, `png` et `gif`, mais aussi de la plupart des vidéos. Vous pouvez bien entendu les assembler dans une archive `tar` et même les compresser, mais vous ne les rendrez en général pas plus petits car ils ont déjà été compressés.

Mes fichiers `.tuto` que je souhaite archiver sont pour le moment placés en vrac dans mon home :

```
$ ls
Bureau      Images      l-
heritage.tuto  Public
Documents  la-surchage-d-
opérateurs.tuto  Modèles      Vidéos
Exemples    les-principaux-widgets.tuto  Musique
```

Il est recommandé de placer d'abord les fichiers à archiver dans un seul et même dossier. Créons-le et déplaçons-y tous nos `.tuto` :

```
$ mkdir tutoriels
$ mv *.tuto tutoriels/
$ ls
Bureau      Exemples  Modèles  Public  Vidéos
Documents  Images    Musique  tutoriels
```

Voilà, nos fichiers sont réunis dans le dossier `tutoriels`.

## -cvf : créer une archive tar

Nous allons maintenant créer une archive `tar` de ce dossier et de ses fichiers. La procédure à suivre pour créer une archive est :

```
tar -cvf nom_archive.tar nom_dossier/
```

J'utilise trois options :

- `-c` : signifie créer une archive `tar` ;
- `-v` : signifie afficher le détail des opérations ;
- `-f` : signifie assembler l'archive dans un fichier.

Essayons de faire cela sur notre dossier `tutoriels` :

```
$ tar -cvf tutoriels.tar tutoriels/  
tutoriels/  
tutoriels/les-principaux-widgets.tuto  
tutoriels/la-surcharge-d-operateurs.tuto  
tutoriels/l-heritage.tuto
```

Ici on archive le dossier `tutoriels` et donc son contenu. Grâce à `-v`, on voit bien la liste des fichiers qui ont été archivés.

*Est-on obligé de mettre systématiquement nos fichiers dans un même dossier pour archiver ensuite ce dossier ? On ne pourrait pas archiver directement les fichiers ?*

Si, c'est possible. Imaginons que nous soyons toujours dans notre home avec nos fichiers `.tuto`. On pourrait très bien faire :

```
tar -cvf archive.tar fichier1 fichier2 fichier3
```

C'est possible et ça fonctionne. Toutefois, il est de coutume sous Linux de placer **d'abord** les fichiers dans un dossier avant de les mettre dans le `tar`. Cela permet d'éviter, lorsqu'on extrait les fichiers de l'archive, que ceux-ci aillent se mêler à d'autres fichiers. Nous allons voir ce problème maintenant.

### **-tf : afficher le contenu de l'archive sans l'extraire**

Vous venez de recevoir une archive `tar` qu'on vous a envoyée. Bien. Mais que contient-elle ?

Avant d'extraire quoi que ce soit, vous aimeriez peut-être voir son contenu. C'est

possible avec `-tf` :

```
$ tar -tf tutoriels.tar
tutoriels/
tutoriels/les-principaux-widgets.tuto
tutoriels/la-surcharge-d-operateurs.tuto
tutoriels/l-heritage.tuto
```

Quand on fait cela, on voit que tous les fichiers sont réunis dans un même dossier `tutoriels`, et ça c'est très pratique.

J'en reviens justement au problème dont je parlais un peu plus haut : imaginez que vous « détariez » une archive contenant plus de 400 fichiers dans votre home. Si ces fichiers n'étaient pas réunis dans un dossier, ils iraient tous se mêler aux autres fichiers du home, et alors là, je vous dis pas la pagaille !

Voilà donc pourquoi je vous ai invités dès le début à réunir vos fichiers à archiver dans un même dossier. Cela permet d'éviter de mauvaises surprises à celui qui extrait les fichiers de l'archive. Quasiment toutes les archives que l'on vous proposera de télécharger suivent ce même schéma et font attention à tout réunir dans un même dossier, mais vérifiez le contenu avant de l'extraire, on ne sait jamais !

### **-rvf : ajouter un fichier**

Si vous avez oublié un fichier, vous pouvez toujours l'ajouter par la suite avec `-rvf` :

```
$ tar -rvf tutoriels.tar fichier_supplementaire.tuto
tutoriels/fichier_supplementaire.tuto
```

### **-xvf : extraire les fichiers de l'archive**

Pour extraire les fichiers, on va utiliser les options `-xvf` (`-x` pour *eXtract*) :

```
$ tar -xvf tutoriels.tar
tutoriels/
tutoriels/les-principaux-widgets.tuto
tutoriels/la-surcharge-d-operateurs.tuto
tutoriels/l-heritage.tuto
```

Les fichiers s'extraient dans le répertoire dans lequel vous vous trouvez. Vérifiez donc avant de les extraire que ceux-ci sont réunis dans un même dossier (avec `-tf`) si vous ne voulez pas que ces fichiers aillent se mélanger à d'autres !

## gzip & bzip2 : compresser une archive

Vous avez maintenant créé une belle archive `tar`. Tous vos fichiers sont réunis là-dedans. Voyons comment compresser cela.

Nous disposons de deux programmes de compression bien répandus dans le monde Linux :

- `gzip` : c'est le plus connu et le plus utilisé ;
- `bzip2` : il est un peu moins fréquemment utilisé. Il compresse mieux mais plus lentement que `gzip`.

À noter qu'il existe aussi le vieux programme `compress`. Il n'est cependant plus vraiment utilisé car on dispose aujourd'hui de meilleurs algorithmes de compression : `gzip` et `bzip2`.

Ces programmes sont simples à utiliser. Ils prennent comme paramètre le nom du fichier à compresser. Ils le compressent et modifient ensuite son nom.

Concrètement, ils ajoutent un suffixe pour indiquer que l'archive a été compressée :

- `.tar.gz` : si l'archive a été compressée avec `gzip` ;
- `.tar.bz2` : si l'archive a été compressée avec `bzip2`.

À titre indicatif, voici les différentes tailles de l'archive, avant et après compression :

Fichier	Taille
<code>tutoriels.tar</code>	130 Ko
<code>tutoriels.tar.gz</code>	35 Ko
<code>tutoriels.tar.bz2</code>	29 Ko

Cela confirme ce que je vous disais : `bzip2` est plus efficace... mais il compresse aussi plus lentement et est moins fréquemment utilisé.

### gzip : la compression la plus courante

Concrètement, le programme `gzip` s'utilise de la manière la plus simple qui soit :

```
gzip tutoriels.tar
```

L'archive est compressée et gagne ensuite le suffixe `.gz`. Elle s'appelle donc désormais `tutoriels.tar.gz`. Voilà pourquoi vous voyez circuler sur l'internet des fichiers `.tar.gz` : cela signifie que ce sont des archives compressées !

Pour décompresser l'archive ensuite, il suffit d'utiliser `gunzip` :

```
gunzip tutoriels.tar.gz
```

L'archive retrouve son état non compressé en `.tar`. Vous pouvez maintenant extraire les fichiers de l'archive comme vous avez appris à le faire un peu plus tôt avec `tar -xvf. ;-`)

### bzip2 : la compression la plus puissante

Le fonctionnement de `bzip2` est le même que celui de `gzip` :

```
bzip2 tutoriels.tar
```

Une archive compressée `tutoriels.tar.bz2` sera alors créée. Pour la décompresser, utilisez `bunzip2` :

```
bunzip2 tutoriels.tar.bz2
```

Vous retrouvez un `.tar` que vous pouvez extraire avec `tar -xvf.`

### Archiver et compresser en même temps avec tar

*C'est bien beau de séparer les étapes, mais cela nous demande de taper deux fois plus de commandes pour compresser et décompresser des fichiers ! Il n'y a pas plus rapide ?*

Si on fait comme cela, c'est essentiellement pour des raisons historiques. Souvenez-vous que Linux ne fait que recopier le fonctionnement d'Unix dont l'origine remonte aux années 1960 !

Heureusement, les choses ont un peu évolué. Il faut toujours archiver puis compresser, mais le programme `tar` est capable d'appeler lui-même `gzip` ou `bzip2` si vous lui donnez les bons paramètres.

## -zcvf : archiver et compresser en gzip

Vous connaissez `tar -cvf` pour créer une archive `tar`. Si vous rajoutez l'option `-z`, l'archive sera automatiquement compressée avec `gzip`.

```
tar -zcvf tutoriels.tar.gz tutoriels/
```

Voilà comment on obtient une archive compressée en une seule commande. :-)

Pour décompresser, c'est pareil, sauf que le `-c` est remplacé par un `-x` comme tout à l'heure :

```
tar -zxvf tutoriels.tar.gz
```

## -jcvf : archiver et compresser en bzip2

Le principe est le même avec `-j` à la place de `-z` :

```
tar -jcvf tutoriels.tar.bz2 tutoriels/
```

Et pour extraire :

```
tar -jxvf tutoriels.tar.bz2 tutoriels/
```

Vous pouvez toujours analyser le contenu de l'archive avant de la décompresser. Avec `-ztf`, vous regarderez à l'intérieur d'une archive « gziptée » et avec `-jtf`, vous regarderez à l'intérieur d'une archive « bzippée-deux ».

## zcat, zmore & zless : afficher directement un fichier compressé

Parfois, on compresses non pas une archive `tar` mais directement un fichier. Par exemple, je peux compresser un fichier `.tuto` directement :

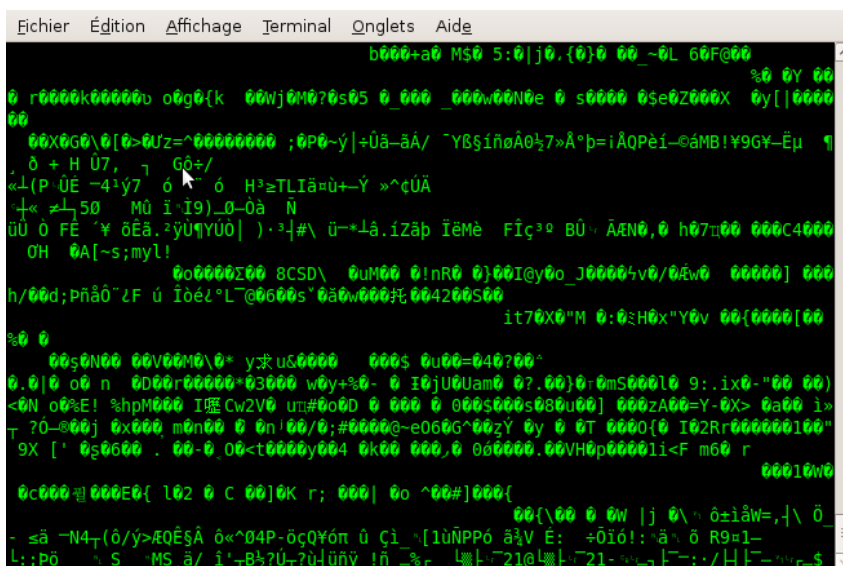
```
gzip l-heritage.tuto
```

Le fichier est alors compressé et renommé en `l-heritage.tuto.gz`.

Maintenant, supposons que nous voulions afficher le contenu de ce fichier sans le décompresser auparavant. Eh bien il existe des outils qui permettent de faire cela !

- `zcat` : équivalent de `cat`, capable de lire un fichier compressé (gzipté).
- `zmore` : équivalent de `more`, capable de lire un fichier compressé (gzipté).
- `zless` : équivalent de `less`, capable de lire un fichier compressé (gzipté).

Si vous essayez de faire un `cat l-heritage.tuto.gz`, vous allez voir des caractères bizarres s'afficher à l'écran, comme le montre la figure suivante.



Ces caractères bizarres constituent une représentation de votre fichier compressé. Comme vous pouvez le voir, ce n'est pas très lisible. À ce stade, votre console est d'ailleurs boguée. Si vous tapez des caractères, vous allez voir que vous allez taper n'importe quoi. Pour

réinitialiser la console, tapez la commande `reset` puis appuyez sur Entrée.

Maintenant, essayez plutôt d'utiliser `zcat`. Ce programme va décompresser le fichier à la volée et l'afficher dans la console :

```
$ zcat l-heritage.tuto.gz
```

```
<conclusion>
```

```
<![CDATA[Ce chapitre en impose peut-  
être un peu par sa taille, mais ne vous y fiez pas ce sont surtout les schémas qui prennent de la place  
zmore et zless, équivalents de more et less qui permettent d'afficher page par page, fonctionnent aussi !
```

## unzip : décompresser un .zip

Les `.tar.gz` et `.tar.bz2` ont beau être courants dans le monde Linux, vos amis utilisant Windows ne les connaissent pas et risquent tôt ou tard de vous envoyer un superbe `.zip` ou `.rar`... que vous ne pouvez pas décompresser avec `gunzip`.

Heureusement, il existe des utilitaires de décompression pour ces formats. Ils ne sont pas toujours installés par défaut, il faudra donc les installer si vous ne les avez pas.

Vous venez de recevoir un `.zip` ?

Pas de panique ! Le programme `unzip` est capable de l'extraire. Il est peut-être installé par défaut, mais si vous ne l'avez pas, vous savez ce qu'il vous reste à faire :

```
sudo apt-get install unzip
```

Ceci étant fait, l'utilisation d'`unzip` est très simple :

```
unzip archive.zip
```

Les fichiers vont s'extraire dans le dossier dans lequel vous vous trouvez ! Le problème est le même qu'avec les `.tar.gz` et `.tar.bz2`. Avant de décompresser, vérifiez si les fichiers sont réunis dans un même dossier.

Pour voir le contenu d'une archive `zip` sans l'extraire, utilisez `-l` :

```
$ unzip -l tutoriels.zip
Archive:  tutoriels.zip
 Length   Date   Time    Name
-----
      0  11-12-08  15:04  tutoriels/
  59515  11-12-08  14:44  tutoriels/les-principaux-widgets.tuto
  36757  11-12-08  14:43  tutoriels/la-surcharge-d-operateurs.tuto
  27685  11-12-08  14:44  tutoriels/l-heritage.tuto
-----
 123957                      4 files
```

On peut voir que les fichiers sont réunis dans un même dossier dans l'archive. C'est plutôt rare avec les `.zip` en principe, faites donc attention avant de décompresser les fichiers pour qu'ils n'atterrissent pas n'importe où.

En général, on a surtout besoin d'`unzip` pour décompresser un `zip`, mais il est peu fréquent que l'on soit amené à créer un fichier `zip` (on préférera toujours le `gzip` ou le `bzip2`). Si toutefois vous voulez vraiment créer un `zip`, installez le programme `zip` puis basez-vous sur la commande suivante :

```
zip -r tutoriels.zip tutoriels/
```

Le `-r` demande à compresser tous les fichiers contenus dans le dossier `tutoriels` (sans ce paramètre, seul le dossier, vide, sera compressé!).

## unrar : décompresser un .rar

Il vous faut installer le paquet `unrar` pour pouvoir utiliser cette commande :

```
sudo apt-get install unrar
```

Ensuite, pour extraire :

```
unrar e tutoriels.rar
```

Non, vous ne rêvez pas, l'auteur du programme ne veut pas que l'on mette un tiret devant l'option `e` ! Il faut bien qu'il y ait des exceptions dans la vie. :-)

Pour lister le contenu avant décompression, utilisez l'option `l` :

```
$ unrar l tutoriels.rar
```

```
UNRAR 3.80 beta 2 freeware      Copyright (c) 1993-2008 Alexander Roshal
```

```
Archive tutoriels.rar
```