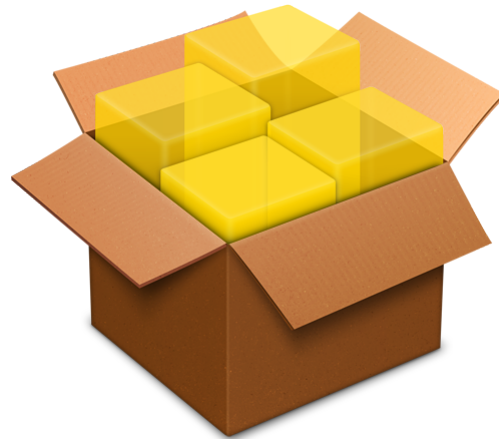


Installation et archivage Linux



Sauvegarde et archivage avec tar

● Généralité :

La commande tar (**=Type ARchive**) est une ancienne commande Unix qui permet aisément d'archiver, c'est-à-dire de réaliser la sauvegarde d'un ensemble de fichiers en un seul fichier, que l'on peut également compresser. Certaines applications et des mises à jour (les noyaux Linux notamment) ne sont livrées que sous forme soit binaire, soit de source à compiler, dans ce format (bien que les applications soient de plus en plus disponibles précompilées, prêtes à l'emploi, sous format **.rpm**)

● Syntaxe :

tar options fichiers

fichiers :

désigne un ensemble de fichiers ou toute une arborescence précédée d'un chemin absolu (à partir de /) ou relatif. Il est recommandé d'indiquer un chemin absolu qui sera conservé dans l'archive et permettra ensuite un désarchivage correctement positionné (sinon il y a installation conformément au chemin relatif conservé, ce qui nécessiterait un exact positionnement dans le système de fichiers).

Sauvegarde et archivage avec tar

tar -options :

Les 3 premières -c -x -t spécifient les 3 types d'actions de la commande

tar -x *extraire le contenu d'une archive*

tar -c *créer une nouvelle archive*

tar -t *afficher seulement la liste du contenu de l'archive, sans l'extraire*

tar -f *fichier indiquer le nom du fichier archive*

tar -v *mode bavard*

tar -z *compresser ou décompresser en faisant appel à l'utilitaire **gzip***

tar -y *compresser ou décompresser avec l'utilitaire **bgzip2***

tar --help *aide*

Sauvegarde et archivage avec tar

EXEMPLES :

1- Création :

tar -cvf sauve.toto.tar /home/toto *effectue la sauvegarde de tous les fichiers du répertoire /home/toto dans le fichier sauve.toto.tar placé dans le répertoire courant.*

tar -cvf /tmp/sauve.toto.tar /home/toto *idem, mais le fichier archive est placé dans le répertoire /tmp*

tar -c /home/toto > sauve.toto.tar *variante de la commande précédente*

tar -cvf sauve.toto.tar /home/toto

tar -cvzf sauve.toto.tar.gz /home/toto *effectue une compression en plus*

Sauvegarde et archivage avec tar

EXEMPLES :

2- Listage :

tar -tvf sauve.toto.tar *pour connaitre l'arborescence regroupée dans le fichier archive, en particulier la place où sera installée son contenu lors du désarchivage.*

3- Extraction :

tar -xvf sauve.toto.tar *exécute le désarchivage dans le répertoire courant.*
si l'archive a été créée par tar -cvf sauve.toto.tar /home/toto, il faut se placer à la racine / pour restaurer exactement le répertoire. personnel de toto.

4-décompresse et désarchive :

tar -xvfz sauve.tar.gz *home/toto/tmp ne désarchive dans l'archive, que le répertoire désigné.*

Compression avec gzip

La commande gzip :

Elle est utilisée pour compacter un fichier quelconque, et en particulier une archive tar. Le décompactage se fait par la commande **gunzip**, ou de manière totalement équivalente par **gzip -d**.

Elle peut décompacter les fichiers .gz, mais aussi les fichiers .z , .Z

Options :

Gzip -# --fast --best fixe le niveau de compression (9 étant le maximum)

Réguler la vitesse de compression en utilisant le chiffre # spécifié, où -1 ou --fast indique la méthode de compression la plus rapide (la moins bonne compression), et -9 ou --best indique la méthode de compression la plus lente (la meilleure compression). Le niveau de compression par défaut -6 (càd biaisé en faveur d'une forte compression au lieu de la vitesse d'exécution).

gzip -d décompresse

Compression avec gzip

La commande gzip :

Options :

gzip -c *Ecrire la sortie sur la sortie standard ; garder les fichiers originaux non modifiés. S'il y a de multiples fichiers, la sortie consiste en une séquence de membres compressés indépendamment. Pour obtenir une meilleure compression, gzip concatène tous les fichiers d'entrées avant de les compresser.*

gzip -l *Pour chaque fichier compressé, afficher les champs suivants :*

compressed size : taille du fichier compressé

uncompressed size : taille du fichier décompressé

ratio : coefficient de compression (0.0% si inconnu)

uncompressed_name: nom du fichier décompressé

La taille du fichier compressé vaut -1 pour les fichiers n'étant pas au format gzip, comme les fichiers compressés .Z.

gzip -f –force

Forcer la compression ou la décompression même si le fichier est cible de plusieurs liens durs, ou si le fichier correspondant existe, ou encore si les données compressées sont lues ou écrites à partir d'un terminal.

Compression avec gzip

La commande gzip :

Options :

gzip -r --recursive

Parcourir la structure du répertoire récursivement. Si l'un des noms de fichiers spécifiés sur la ligne de commandes est un répertoire, gzip descendra dans ce répertoire et compressera tous les fichiers qu'il y trouve (ou les décompressera dans le cas de gunzip).

gzip -t --test

Tester. Vérifier l'intégrité du fichier compressé.

gzip -v --verbose

Mode bavard. Afficher le nom et le pourcentage de réduction pour chaque fichier compressé ou décompressé.

gzip -q --quiet

Supprimer tous les avertissements.

Compression avec gzip

La commande gzip :

Exemples :

gzip backup.tar /home/toto compresse **backup.tar** et le remplace par le fichier **backup.tar.gz**, d'une taille beaucoup plus réduite.

Attention, le fichier d'origine est donc détruit !

gzip -9 *.txt compresse au maximum chaque fichier **.txt** séparément, et les renomme en ajoutant le suffixe **.gz**

Autre utilitaire :

bzip2

utilise la même syntaxe que **gzip**, mais compresse mieux avec un besoin accru de mémoire

Compression avec gzip

La commande bzip2 :

bzip2

utilise la même syntaxe que gzip, mais compresse mieux avec un besoin accru de mémoire

Exemples :

- Compression de fichier en ".bz2 " :

```
# bzip2 fichier_archive.tar.tar
```

- Resultats Compression de fichier en ".bz2" :

```
#ls  
fichier_archive.tar.bz2
```

- décompression de fichier en « bz2 :

```
# bzip2 -d fichier_archive.tar.tar
```

Attention : ici l'archive **.tar** va être directement remplacé par l'archive compressé en **tar.bz2**. Afin d'éviter cela mettez l'option **-k** comme ceci :
#bzip2 -k fichier_archive.tar.tar

A noter que cet option **-k** est aussi , valable pour la commande **tar**

EXERCICE avec tar ; gzip ; bzip2

Mettez vos paramètres gdm et gdm-password à l'état initial.

1- *Créer un utilisateur générique nommé stagex*

2- *Sous l'identité root, créer le répertoire. /home/stagex/archives
Pourquoi nécessairement est-ce le travail de root ?*

3. *Sous quel masque root a t-il créé ce rép ? stagex va t-il pouvoir y archiver ses documents ?*

4. *Faire ensuite le nécessaire pour que l'user stagex puisse se réserver exclusivement l'accès et l'usage de son répertoire. personnel d'archivage.*

5. *stagex archive dans /home/archives/stagex, sous le nom sauve.stagex.tar, son répertoire personnel /home/stagex.*

6 *Puis il effectue maintenant des sauvegardes compressées, par **gzip** et **bzip2**, respectivement sous les noms : sauve.stagex.tar.gz et sauve.stagex.tar.bz2*

EXERCICE avec tar ; gzip ; bzip2

- 7) *Vérifier l'existence et comparer les tailles des 3 archives obtenues.*
- 8) *Supprimer le repertoire courant de stagex*
- 9) *stagex_très maladroit_ détruit son répertoire. personnel /home/stagex*
- 10) *Heureusement, il peut effectuer un sauvetage ! comment ? aidez-le !*

Gestionnaire de package RPM et YUM

Fedora utilise les fichiers dits « RPM » (RPM Package Manager) qui sont des archives contenant des programmes pré-compilés prêts à l'emploi.

Évidemment, il est possible comme sur certains systèmes non communautaires de les chercher un par un sur internet et de les installer manuellement. Mais la gestion des dépendances peut quelquefois devenir un véritable casse-tête (un logiciel qui en nécessite un autre, et un autre...).

Pour gérer les paquets du système, Fedora utilise YUM, un outil permettant de gérer les installations, les désinstallations et les mises à jour de paquets au format RPM. Il gère les dépendances en téléchargeant ce qui est nécessaire. Il trouve les paquets sur différentes sources (sites internet) que l'on appelle des dépôts (*english ~ repository*) .

YUM est fourni en standard dans toutes les versions de Fedora.

Gestionnaire de package RPM et YUM

RPM est l'outil de base permettant d'installer des logiciels préparés au format « RPM ».

Yum est d'une certaine façon une surcouche à rpm facilitant la vie de l'utilisateur. -
Téléchargement de paquets depuis des dépôts, recherche des dépendances, etc.

Yum n'est pas le seul logiciel permettant ce genre de choses, Smart par exemple est une alternative à Yum.

Contrairement à Smart, Yum s'utilise uniquement en ligne de commande, c'est pour cela que des interfaces graphiques existent tel que PackageKit qui est fournit par défaut sur Fedora (PackageKit est accessible depuis le menu, rubrique « Application > Ajout/suppression de programmes »).

Toutes les opérations de base sont faisables avec PackageKit : ajouter/modifier des dépôts, installer/supprimer un logiciel, mettre à jour, etc.

Mais pas tout. De plus, il est plus facile d'aider quelqu'un directement avec Yum, copier/coller du texte c'est beaucoup plus facile. C'est pour cela que sur les forums vous trouverez très souvent l'utilisation de Yum au lieu d'une interface graphique quelle qu'elle soit.

LA COMMANDE YUM

1) *Mis à jour tout votre système via yum (toujours en root)*

```
# yum update
```

2) *Si c'est la première fois que vous mettez votre système à jour sur une installation toute fraîche de Fedora, lancez la commande .*

Cette commande permet à YUM de se mettre à jour indépendamment des autres paquets.

```
# yum update yum
```

3) *Pour rechercher un paquet, faites la commande :*

```
# yum list <nom du paquet>
```

Bien sûr, ceci n'est pas très pratique si l'on ne connaît pas très bien le nom du paquet.

*Heureusement pour vous, YUM accepte le caractère joker * (étoile). Ce caractère joker fonctionne également avec les autres commandes. Remarque : il est nécessaire d'« échapper » le joker en plaçant le caractère \ devant celui-ci ou bien en sécurisant toute la chaîne de caractère avec les guillemets. Exemple :*

```
# yum list kernel\*
```

LA COMMANDE YUM

1) *Mis à jour tout votre système via yum (toujours en root)*

```
# yum update
```

2) *Si c'est la première fois que vous mettez votre système à jour sur une installation toute fraîche de Fedora, lancez la commande .*

Cette commande permet à YUM de se mettre à jour indépendamment des autres paquets.

```
# yum update yum
```

3) *Pour rechercher un paquet, faites la commande :*

```
# yum list <nom du paquet>
```

Bien sûr, ceci n'est pas très pratique si l'on ne connaît pas très bien le nom du paquet.

*Heureusement pour vous, YUM accepte le caractère joker * (étoile). Ce caractère joker fonctionne également avec les autres commandes. Remarque : il est nécessaire d'« échapper » le joker en plaçant le caractère \ devant celui-ci ou bien en sécurisant toute la chaîne de caractère avec les guillemets. Exemple :*

```
# yum list kernel\*
```

ou

```
# yum list "kernel*"
```

YUM vous listera les paquets correspondants à votre recherche installés et ceux que vous pouvez installer.

LA COMMANDE YUM

8) Pour exclure un dépôt temporairement :

```
# yum --disablerepo=<nom du dépôt>
```

9) Pour inclure un dépôt temporairement :

```
# yum --enablerepo=<nom du dépôt> update
```

10) Yum peut de plus gérer des groupes de RPMs, pour faciliter l'installation d'ensembles complexes, comme par exemple les environnements graphiques.

Pour connaître la liste des groupes disponible:

```
# yum grouplist
```

11) Pour avoir des informations sur un groupe, comme les paquets qui le composent:

```
# yum groupinfo "Nom du Groupe"
```

```
# yum --exclude=kernel* update
```

LA COMMANDE YUM

12) Pour installer un groupe:

```
# yum groupinstall "Nom du Groupe"
```

13) Pour supprimer un groupe

```
# yum groupremove "Nom du Groupe"
```

14) Pour connaitre la liste des dépôts configurés:

```
# yum repolist all
```

15) Pour les autres commandes YUM

```
# yum help
```

LA COMMANDE RPM

Interrogation

1) Installation d'un paquetage :

```
# rpm -ivh package.rpm
```

2) Mise à jour d'un paquetage

```
# rpm -Uvh package.rpm"
```

-i -> install

-U -> Upgrade (met à jour un paquet ou l'installe s'il n'est pas présent)

-v -> verbose (détaille l'avancement de l'installation)

-F -> Freshen (ne met à jour un paquet que s'il est installé)

-h -> hash (permet d'avoir une « barre de progression »)

NB:

Pour l'installation (ou la mise à jour), le paquet peut être un fichier sur un support quelconque ou une URL.

LA COMMANDE RPM

Installation / Mise à jour

3) *Lister des informations relatives aux paquetage :*

```
# rpm -qip package.rpm
```

Détails:

- q -> interrogation
- i -> texte décrivant le paquetage
- l -> liste les fichiers du paquetage
- p -> pour interroger un fichier.rpm et non pas un paquetage installé

4) *On peut interroger un paquetage installé de la même manière avec :*

```
# rpm -qip paquetage
```

5) *Interroge tous les paquetages installées :*

```
# rpm -qa
```

LA COMMANDE RPM

Installation / Mise à jour

6) *Liste tous les fichiers de chaque paquetage (pratique pour rechercher un fichier particulier)*

```
# rpm -qa | grep -i package
```

7) *Liste tous les fichiers de chaque paquetage (pratique pour rechercher un fichier particulier)*

```
# rpm -qail
```

8) *Renvoie quel paquet rpm a installé un certain fichier : permet ainsi de déterminer le nom du paquet ayant installé tel programme ou telle librairie*

```
# rpm -q --whatprovides /usr/bin/ogg123
```

9) *Renvoie tous les fichiers installés par un paquet :*

```
# rpm -q --fileprovides vorbis-tools  
# rpm -ql vorbis-tools
```

LA COMMANDE RPM

Installation / Mise à jour

10) *Si un paquet fait sa maline et veut pas s'installer pour x ou y raison*

```
# rpm -ivh --nodeps --force nom du package
```

11) *Connaître le paquet d'un fichier (exemple : ci-dessous pour l'annuaire open ldap)*

```
# rpm -qf /etc/openldap/ldap.conf
```

12) *Connaître le contenu d'un rpm*

```
# rpm -pql package.rpm
```

13) *Connaitre la date d'installation d'un paquet :*

```
# rpm -qa --last | grep truc
```

LA COMMANDE RPM

Installation / Mise à jour

14) Forcer la désinstallation d'un paquet :

```
# rpm -e --justdb --nodepse
```

15) Afficher les changes logs

```
# rpm -q truc --changelog
```

13) Affiche plus d'information ainsi que l'architecture d'un paquet, exemple avec libxml2 :

```
# rpm -rpm --query --queryformat "%{NAME}-%{VERSION}.%{RELEASE} (%{ARCH})\n" libxml2
```

LA COMMANDE RPM

Suppression

Il est fortement conseillé d'utiliser YUM pour supprimer des paquets, l'utilisation de la commande rpm directement peut conduire à l'altération de la base RPM

```
# rpm -e package
```

14) Deux options très utiles :

*--force -> force l'installation (utile pour forcer la réinstallation d'un paquet, par exemple)
--nodeps -> ne vérifie pas les dépendances*

Leur utilisation reste à proscrire dans la plupart des cas si l'on veut conserver une base RPM cohérente.

15) *si vous êtes bloqué avec un paquet, que vous n'arrivez pas à le désinstaller parce que vous avez supprimé des fichiers à la main avant de faire un rpm -e vous avez cette solution qui vire le paquet de la liste des paquets installé :*

Forcer la désinstallation d'un paquet :

```
# rpm -e --justdb --nodeps
```