* 1. L’utilitaire RPM package manager
1. **Présentation**

Il y a longtemps, lorsque Linux en était encore à ses balbutiements, le moyen le plus courant d’installer un logiciel était un fichier compressé (généralement sous forme d'archive. tar .gz) avec le code source, que vous décompressez et compiliez vous-même.

Cependant, à mesure que la quantité et la complexité des logiciels augmentaient, le besoin d'un moyen d’installer des logiciels précompilés est devenu évident. Après tout, tout le monde n'avait pas les ressources, à la fois en temps et en puissance de calcul, pour compiler de gros projets comme le noyau Linux ou un serveur X.

Bientôt, les efforts pour normaliser une manière d’installer ces progiciels se sont multipliés et les premiers gestionnaires de progiciels sont nés. Ces outils facilitaient grandement l'installation, la configuration ou la suppression de logiciels d'un système.

L'un d'eux était le RPM Package Manager et son outil correspondant (rpm), développé par Red Hat.

Aujourd'hui, ils sont largement utilisés non seulement sur Red Hat Enterprise Linux (RHEL), mais aussi sur ses descendants, comme Fedora, CentOS et Oracle Linux, d'autres distributions comme UBUNTU et même d'autres systèmes d'exploitation, comme IBM'S AIX.

D'autres outils de gestion de packages populaires sont yum (YellowDog Updater Modified), dnf (Dandified YUM) et apt (Advanced Packaging Tool), qui peuvent rationaliser de nombreux aspects de l'installation, de la maintenance et de la suppression des packages, en créant des packages gestion beaucoup plus facile.

1. **Chercher un package**

Pour avoir des informations sur les package installés, comme la version, l’architecture, date d’installation, etc… vous tapez **rpm** avec le paramètre **-qi** (query info) suivi par le nom du package**:**



Pour obtenir une liste des fichiers contenus dans un package installé, utilisez les paramètres **-ql** (query list ) suivi du nom du package :



Pour avoir plus d’information sur les options il suffit de mettre :

**rpm - ? | --help**

1. **Installer, mettre à jour et supprimer un package**

L'opération la plus basique consiste à installer un package, ce qui peut être fait avec :

**rpm {-i | --install} [options-installation] FICHIER\_PAQUETAGE**

Pour supprimer un paquet installé, passez le paramètre **-e** (comme dans « effacer ») à rpm, suivi du nom du paquet que vous souhaitez supprimer :

**rpm {-e | --erase} [options-installation] FICHIER\_PAQUETAGE**

Si le package est installé il sera supprimé si non vous aurez un message d’erreur : **Erreur : le paquet « NOM\_PAQUET » n’est pas installé**

S'il existe une version précédente d'un package sur le système, vous pouvez effectuer une mise à niveau vers une version plus récente à l'aide du paramètre -U :

**rpm {-U | --upgrade} [options-installation] FICHIER\_PAQUETAGE**

****

S'il n'y a pas de version précédente de **FICHIER\_PAQUETAGE** installée, une nouvelle copie sera installée. Pour éviter cela et mettre à niveau uniquement un package installer, utilisez l'option **-F**.

Dans les deux opérations, vous pouvez ajouter le paramètre **-v** pour obtenir une sortie verbeuse (plus d'informations sont affichées pendant l'installation) et **-h** pour obtenir des signes dièse (#) imprimés comme une aide visuelle pour suivre la progression de l'installation. Plusieurs paramètres peuvent être combinés en un seul, donc **rpm -i -v – h** est identique à **rpm -ivh**.

1. **Trouver quel paquet fournit un fichier spécifique**

Pour savoir quel paquet installé possède un fichier, utilisez le **-qf** (query file) suivi du chemin d'accès au fichier :



Pour vérifier vis-à -vis d'un fichier particulier :

**rpm -Vf Nom\_Fichier**

1. **Obtenir une liste de tous les packages installés dans le système**

Pour vérifier tous les packages :

**rpm -Va**

Pour vérifier un package particulier :

**rpm -Vp Nom\_Package**

Lorsqu'il n'y a pas d'erreur, le système n'affiche rien. Si par contre une erreur a été détectée dans le package installé, le système renvoie une chaîne de huit caractères indiquant le type d'erreur. Voici la signification des caractères

* D: Périphérique
* G: Groupe
* L: Lien symbolique
* M: Mode (Permissions)
* T: date du fichier
* S: Taille du fichier
* U: Utilisateur



* 1. YUM/DNF (Dandified yum)
1. **Présentation**

**yum** a été développé à l'origine sous le nom de Yellow Dog Updater (YUP), un outil de gestion de paquets sur la distribution Yellow Dog Linux. Au fil du temps, il a évolué pour gérer les packages sur d'autres systèmes basés sur **RPM**, tels que Fedora, CentOS, Red Hat Enterprise Linux et Oracle Linux.

Fonctionnellement, il est similaire à l'utilitaire **apt** sur les systèmes basés sur Debian, étant capable de rechercher, d'installer, de mettre à jour et de supprimer des packages et de gérer automatiquement les dépendances. **yum** peut être utilisé pour installer un seul paquet ou pour mettre à niveau tout un système à la fois

**dnf** est l'outil de gestion de paquets utilisé sur Fedora, et elle est une dérivé de **yum**. En tant que tel, de nombreuses commandes et paramètres sont similaires.

**NB :** l’usage de **dnf** est identique à celui de **yum** ce qui va remplacer l’utilisation de cette dernière

1. **Chercher un package**

Pour installer un paquet, vous devez connaître son nom. Pour cela, vous pouvez effectuer une recherche avec **yum/dnf search PATTERN**, où **PATTERN** est le nom du package que vous recherchez. Le résultat est une liste de packages dont le nom ou le résumé contient le modèle de recherche spécifié.



1. **Récupérer des informations sur un package**

Pour obtenir des informations sur un package, telles que sa version, son architecture, sa description, sa taille, etc., utilisez **yum/dnf** **info PACKAGE\_NAME** où PACKAGE\_NAME est le nom du package pour lequel vous souhaitez des informations :



1. **Installer, mettre à jour et Supprimer un package**

Pour installer un package à l'aide de **yum**, utilisez la commande **yum install PACKAGE\_NAME**, où PACKAGE\_NAME est le nom du package. **yum** récupérera le package et les dépendances correspondantes à partir d'un référentiel en ligne et installera tout sur votre système. idem pour la commande **dnf** on table **dnf install PACKAGE\_NAME.**

****

Pour supprimer un package on tape **yum remove PACKAGE\_NAME**

1. **Trouver quel paquet fournit un fichier spécifique**

Si vous connaissez la commande, le fichier de configuration ou le nom de la bibliothèque que vous souhaitez, mais que vous ne savez pas dans quel package il se trouve, utilisez la sous-commande **provides** ou **whatprovides** pour rechercher le package.



En ce qui concerne la commande dnf on tape **dnf provides PACKAGE\_NAME**



1. **Obtenir une liste de tous les packages installés dans le système**

Pour obtenir la liste de tous les packages installés dans le système on tape **yum list installed** ou via la commande **dnf list --installed**





Pour ajouter un référentiel, utilisez **dnf config-manager --add\_repo URL**, où URL est l'URL complète du référentiel. Pour activer un référentiel, utilisez **dnf config-manager --set-enabled REPO\_ID**.

De même, pour désactiver un référentiel, utilisez **dnf config-manager --set-disabled REPO\_ID**. Dans les deux cas, REPO\_ID est l'identifiant unique du référentiel, que vous pouvez obtenir à l'aide de **dnf repolist**. Les référentiels ajoutés sont activés par défaut.

Les référentiels sont stockés dans des **fichiers .repo** dans le répertoire **/etc/yum.repos.d/**, avec exactement la même syntaxe que celle utilisée pour **yum**.



* 1. Installer un logiciel

• **Téléchargement et chargement**

Bien que Linux dispose de gestionnaires de packages qui facilitent beaucoup l'installation du logiciel, on est parfois obligé de créer un package à partir de la source.

La plupart des packages Linux sont gratuits et open source, ça nous donne la liberté de personnaliser ou de modifier n'importe quel logiciel. De plus, on est également libre de consulter le code source des packages Linux pour apprendre les bonnes pratiques d'architecture et modèles de codage d'autres projets logiciels.

La plupart des logiciels Linux sont écrits dans les langages de programmation C ou C++, on aura donc besoin d'un compilateur C ou C++. Par exemple, la GNU Compiler Collection GCC et CMake pour construire notre package.

En plus de cela, on aura besoin d'autres packages tels que **curl** et **gettext**. En fonction de la distribution Linux, on peut installer les outils requis en une seule commande comme suit.

**sudo dnf install dh-autoreconf curl-devel expat-devel gettext-devel openssl-devel perl-devel zlib-devel gcc curl cmake**

Pour cette partie, nous allons choisir le package GIT à partir de la source puisque c’est très utilisé par les ingénieurs et ça va nous donner une idée très élargie sur la manière d’installation à partir de la source.

La plupart des packages que nous allons compiler se trouvent sur le site officiel du package en question. Nous pouvons télécharger les fichiers de code source en utilisant l'outil de ligne de commande **curl**. Alternativement, Nous pouvons utiliser **wget** ou l'interface graphique s’il s’agit d’une version desktop ou encore s’il s’agit d’un dépôt github vous pouvez le cloner en utilisant **git clone [lien]**.

On procède au téléchargement et par la suite on se déplace vers le dossier cible via la commande **CD**

****

• **Compilation**

C'est toujours une bonne idée de jeter un œil sur README ou INSTALL fichiers car ils contiennent des informations précieuses sur la façon de compiler et d'installer le package. Ces fichiers sont généralement situés dans le dossier racine du code source.



Un autre fichier important est le configurer script. Il vérifie les dépendances logicielles pour le package que nous allons compiler, et vous verrez un message d'erreur si le script trouve des dépendances manquantes.

Configurez et préparez votre code source en exécutant le script. La commande créera faire fichiers et configurations pour le logiciel que vous êtes sur le point de compiler et d'installer.

 **$ make install**

****

 **$ make configure**

****

**• Installation et exécution**

Après avoir exécuter la commande **« ./configure »** on obtient le résultat suivant :





 **$ make all doc**

La commande make utilise le Makefile, qui contient les instructions nécessaires sur la façon de construire le progiciel.

Le processus de compilation prendra un certain temps en fonction de la puissance de traitement de l’ordinateur et de la taille du package.



Dans cette dernière étape, nous allons installer le package logiciel Git que nous venons de créer à partir du code source. Cette commande installe le package nouvellement compiler en copiant les fichiers de construction aux emplacements corrects sur PC.

 **$make install install-doc install-html;** *en tant que root*

****

Pour vérifier on tape la commande git –-version :



Le numéro de version peut varier en fonction du package télécharger.

Questions :

1. En utilisant la commande **rpm** sous **Fedora** linux installer le file-roller-3.28.1-2.el7.x86\_64.rpm qui se trouve dans le lien « [**http://mirror.centos.org/centos/7/os/x86\_64/Packages/file-roller-3.28.1-2.el7.x86\_64.rpm**](http://mirror.centos.org/centos/7/os/x86_64/Packages/file-roller-3.28.1-2.el7.x86_64.rpm)**»** montrant une barre de progression pendant l'installation.

🡺rpm -ivh file-roller-3.28.1-2.el7.x86\_64.rpm

1. À l'aide de la commande rpm, découvrez quel paquet contient le fichier /etc/redhat-release

🡺rpm -qf /etc/redhat-release

1. Comment utiliseriez-vous yum et dnf pour vérifier les mises à jour de tous les packages du système ?

🡺dnf check-update | dnf update

🡺yum check-update | yum update

1. En utilisant dnf désinstaller le package cockpit

🡺dnf remove cockpit

1. Désactivez le dépôt **updates** via la commande **dnf**

🡺dnf config-manager –-set-disable updates

🡺Dnf repolist all

 TP:

1. Mettez à jour votre système et assurez-vous que vous avez la dernière version.
2. Installez le « cockpit » sur votre machine linux
3. Désactivez le dépôt updates via la commande dnf et yum
4. À l'aide de la commande rpm, découvrez quel paquet contient le fichier /etc/redhat-release
5. En utilisant la commande rpm sous Fedora linux installer le file-roller-3.28.1-2.el7.x86\_64.rpm qui se trouve dans le lien « http://mirror.centos.org/centos/7/os/x86\_64/Packages/file-roller-3.28.1-2.el7.x86\_64.rpm » montrant une barre de progression pendant l'installation.
6. En utilisant dnf désinstaller le package cockpit
7. Comment utiliseriez-vous yum et dnf pour vérifier les mises à jour de tous les packages du système ?
8. Le formateur vous demande d’utiliser Packet tracer et vous avez une machine linux. En respectant les étapes du cours, installez ce logiciel.
9. Votre supérieur vous demande de lui trouver une solution pour avoir un service de VOIP dans l’entreprise. Durant votre recherche, vous avez trouvé la solution « Asterisk\* ». Vous lui avez proposé et il vous a demandé de juste l’installer rien de plus. Votre mission et d’installer cette solution en suivant les étapes.
10. Après avoir installé la solution assurez-vous que cette dernière est bien installée et qu’il n y a pas d’erreur de compilation ou de manque d’indépendances.