

# Installation et gestion de paquetages

Abdelali SAIDI

abdelali.saidi@gmail.com

# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources
- 3 RPM: Redhat paquetage Manager
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian paquetage
- 6 APT-GET

# Plan

- 1 Bibliothèques et dépendances
- 2 Installation depuis les sources
- 3 RPM: Redhat paquetage Manager
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian paquetage
- 6 APT-GET

# Bibliothèques et dépendances

- La plupart des logiciels utilisent des bibliothèques
  - Le développeur bénéficie de plusieurs fonctions de base
- Sur quelques SE, les logiciels sont attachés aux bibliothèques utilisées
  - Redondance de bibliothèques
- Les SE GNU/Linux procurent les logiciels sous forme de paquetage sans les bibliothèques utilisées
- L'installation d'un paquetage nécessite l'installation de ses dépendances
  - Pas de redondance au niveau des bibliothèques
  - S'il y a une issue de sécurité chez une bibliothèque, on fera la correction une seule fois
- Les bibliothèques partagées peuvent être utilisées par plusieurs programmes lors de leurs exécutions

# Bibliothèques et dépendances

## Les bibliothèques partagées

Lors de l'exécution d'un programme, s'il n'a pas besoin d'un code additionnelle on dit qu'il est lié statiquement. Sinon, ce sera un programme lié dynamiquement. Cela veut dire que ce genre cherche du code depuis des bibliothèques et l'utilise pour le bon déroulement de son exécution.

- Ces bibliothèques peuvent être partagées par plusieurs programmes
- Elles ont “.so” (shared object) comme extension suivit par la version
- Pour lister la dépendance d'un programme on lance la commande *ldd*

# Bibliothèques et dépendances

Pour l'installation d'un logiciel sur un SE GNU/Linux, il faut soit compiler son code source et installer le binaire soit télécharger directement le binaire et procéder à l'installation

- Nous avons deux formats principales de paquetages: RPM et DEB
- Le format RPM est utilisé chez Red Hat, Mandriva, Fedora
- Le format DEB est utilisé chez Debian, Ubuntu, Knoppix

Même si beaucoup de programmes sont disponibles en un format binaire, il est toujours bien intéressant de télécharger le code source, le compiler et puis installer le binaire

- Cette méthode marche sur toute distribution GNU/Linux
- Très pratique si on veut personnaliser la compilation

# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources**
- 3 RPM: Redhat paquetage Manager
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian paquetage
- 6 APT-GET

# Installation depuis les sources

Les sources sont emballés dans un fichier .tar qui peut être compressé par bzip2 ou bien gzip. L'installation d'un logiciel archivé suit toujours la même logique sur n'importe quelle distribution:

- Extraction des sources
- Exécution du script configure
- Compilation des sources (make)
- Installation du binaire (make install)

## NB

- L'installation du binaire doit être exécuté avec les droits du root
- Chaque .tar de logiciel contient un fichier texte nommé README ou bien INSTALL, pensez à le lire avant de penser à l'installation du logiciel



# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources
- 3 RPM: Redhat paquetage Manager**
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian paquetage
- 6 APT-GET

# Présentation

- Le RPM est un gestionnaire de paquetages publié sous la licence GPL
- Il maintient une base de données des paquetages et leurs fichiers
  - Cela permet des recherches et des vérifications flexibles sur le système concernant les paquetages installés
- L'exécution de programmes .rpm affecte le système
  - Nécessité de privilèges root
- Il vérifie la signature du paquetage avant son installation
  - Au cas où cette signature n'est pas la bonne, on aura un affichage d'un message du genre: "error: V3 DSA signature: BAD, key ID 0352860f"

# Modes d'opération

Le RPM comprend cinq modes d'opération de base:

- L'installation
- La mise à jour
- La désinstallation
- La recherche

## NB

Le nom de fichier rpm ressemble généralement à xsnow-1.41-1.i386.rpm

- xsnow: le nom du paquetage
- 1.41: la version
- 1: l'édition
- i386: l'architecture qui supportera son installation

# Modes d'opération

## L'installation

Pour installer un paquetage, on lance la commande:

- `rpm -ivh xsnow-1.41-1.i386.rpm`
  - `-i` définit le mode "installation"
  - `-v` l'option verbose (bavard)
  - `-h` affiche l'état d'avancement d'une installation en cours

# Modes d'opération

## L'installation

Les cas possibles après le lancement de l'installation d'un paquetage:

- Installation réussie: dans ce cas, on verra sur l'écran
  - Preparing ... ##### [100%]
- paquetage déjà installé:
  - Preparing ... ##### [100%]  
paquetage is already installed
- Conflits de fichiers:
  - Preparing ... ##### [100%]  
Conflits with file from paquetage ...
- Dépendance non résolue:
  - Preparing ... ##### [100%]  
error: failed dependencies: pack.so.2 is needed by paquetage ...

# Modes d'opération

## L'installation

Les options suivantes permettent d'ignorer les cas où l'installation ne réussie pas:

- `-replacepks` : pour procéder à l'installation même si le paquetage est déjà installé
- `-replacefiles`: Pour procéder à l'installation même s'il y a un conflit de fichiers
- `-nodeps`: Pour procéder à l'installation même s'il y a une absence d'une dépendance

# Modes d'opération

## La désinstallation

Pour désinstaller un paquetage, on lance la commande

- `rpm -e paquetage-1.0-2`

Si le paquetage qu'on veut désinstaller est utilisé par d'autres paquetages, on aura une erreur de dépendance.

- Comment peut-on désinstaller un tel paquetage?

# Modes d'opération

## La mise à jour

Pour la mise à jour d'un paquetage, on lance la commande

- `rpm -Uvh paquetage-1.0-2`

Si le paquetage qu'on veut mettre à jour est déjà installé et ancienne, le RPM le désinstallera pour le remplacer avec le paquetage indiqué sur la commande



# Modes d'opération

## La recherche

Pour la recherche d'un paquetage et la vérification de son installation sur le système, on lance la commande:

- `rpm -q paquetage-1.0-2`

Dans ce mode, on peut utiliser les options:

- `-a` : pour afficher tous les paquetages installés
- `-f` : pour afficher les paquetages qui possèdent le fichier indiqué avec cette option

# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources
- 3 RPM: Redhat paquetage Manager
- 4 YUM: Yellow dog Updater Modified**
- 5 DPKG: Debian paquetage
- 6 APT-GET

# YUM: Yellow dog Updater Modified

## Présentation

YUM est un outil permettant l'installation, la mise à jour et la suppression des paquetages rpm.

## Caractéristiques

- Il gère les dépendances entre les paquetages
- Il est configurable depuis le fichier `/etc/yum.conf`

# YUM: Yellow dog Updater Modified

## Options

- `yum install app(s)` : installe la dernière version d'un paquetage ou d'un groupe de paquetages en s'assurant que toutes les dépendances sont satisfaites
- `yum update [app(s)]` : met à jour les paquetages indiqués en argument, si ces derniers ne sont pas fournis, la commande mettra à jour tous les paquetages se trouvant sur le système
- `yum remove/erase app(s)` : supprime `app(s)` ainsi que tous les paquetages qui en dépendent
- `yum search chaine` : cherche des paquetages dont la description, le résumé, le nom, ou le nom de l'empaqueteur, contiennent la chaîne indiquée.
- `yum list` : affiche diverses informations sur les paquetages.

# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources
- 3 RPM: Redhat packaging Manager
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian packaging**
- 6 APT-GET

# DPKG: Debian paquetage

## Présentation

DPKG est l'outil de gestion de paquetage pour la distribution Debian. Il permet d'installer, de désinstaller, de visualiser, de configurer et de construire des paquetages Debian.

## Options

- `-i app.deb` : installe le paquetage `app.deb`
- `-r app.deb` : désinstalle le paquetage `app.deb`
- `-l | grep app.deb` : vérifie si le paquetage `app.deb` est installé déjà
- `-L app.deb` : cherche les fichiers du paquetage `app.deb` et leur emplacement
- `-unpack app.deb` : permet d'extraire l'archive sans le configurer

## NB

- `dpkg-reconfigure` : permet de reconfigurer un paquetage déjà installé
- `/etc/dpkg/dpkg.conf` est le fichier de configuration
- `/var/lib/dpkg` contient la base de données des paquetages Debian sous forme

# Plan

- 1 Bibliotheques et dependances
- 2 Installation depuis les sources
- 3 RPM: Redhat paquetage Manager
- 4 YUM: Yellow dog Updater Modified
- 5 DPKG: Debian paquetage
- 6 APT-GET**

# APT-GET

## Présentation

Le système de gestion de paquetages Debian est doté d'un autre outil de gestion avancé appelé APT (Advanced paquetage Tool).

## Caractéristiques

De la même manière avec laquelle YUM se caractérise pour RPM, APT-GET ajoute des fonctionnalités supplémentaires :

- la définition de la source des applications à installer (disques locaux, cédérom ou sur Internet par le protocole HTTP ou FTP)
- la gestion des dépendances.
- les sources se trouvent sur `/etc/apt/sources.list`



# APT-GET

## Options

- `apt-get install app` : installe le paquetage `app`
- `apt-get remove app` : désinstalle le paquetage `app`
- `apt-get --purge remove app` : désinstalle le paquetage `app` et supprime ses fichiers de configuration
- `apt-get install app1 app2-` : installe `app1` et désinstalle `app2`
- `apt-get remove app1 app2+` : désinstalle `app1` et installe `app2`
- `apt-get --reinstall app` : réinstalle le paquetage `app`
- `apt-get update` : met à jour la liste des paquetages disponibles
- `apt-get upgrade` : met à jour tous les paquetages pouvant être mis à jour
- `apt-get dist-upgrade` : met à jour le système tout entier