

SAÉ S1.03 – Cours 2

Installation d'outils de développement

Département Informatique

IUT2, UGA

2022/2023

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak
- 7 Résumé

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak
- 7 Résumé

Outils de développement

Pour cette SAÉ

- Compilateur Java : JDK (*Java Development Kit*)
- Machine virtuelle Java (JVM) : JRE (*Java Runtime Environment*)
- IDE pour Java : NetBeans, IntelliJ IDEA
- Gestionnaire de code source : `git`

Autres outils et langages courants

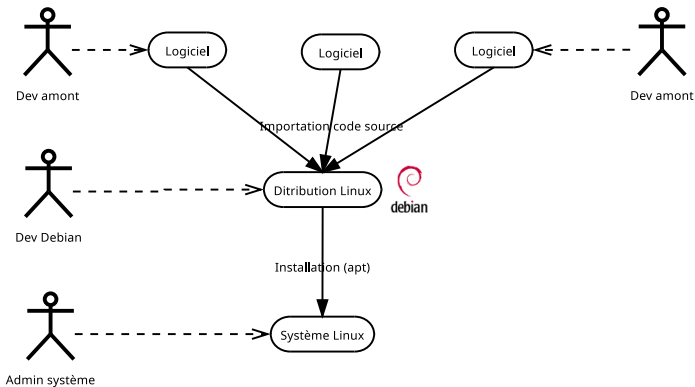
- IDE pour Android : Android Studio
- Autres IDE : CodeBlocks, CodeLite, Geany, Anjuta, KDevelop, ...
- Compilateur C : `gcc` et `clang`
- Compilateur C++ : `g++` et `clang++`
- Langage C# : `mono`, compilateur `mcs`
- Interpréteur Perl : installé en standard
- Interpréteur Python : installé en standard
- Interpréteur Ruby : utilisé par certains logiciels de Debian
- Interpréteur PHP
- ...

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels**
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak
- 7 Résumé

Schéma de base d'une distribution Linux

- Une distribution Linux est aussi un magasin d'applications



Avantages en terme de sécurité

- Logiciels libres avec code source donc inspectables
→ pas de *malware* caché
- Signature du code à chaque étape (GnuPG)
→ garantie de l'authenticité du code
- Màj de sécurité rapides et depuis une source unique
→ facilité de màj

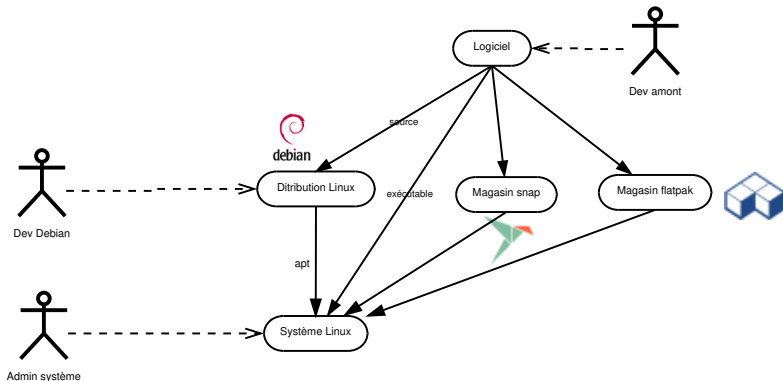
Distribution directe par les auteurs

- Sous forme de code source
 - archive
 - dépôt `git`
- Sous forme de code exécutable (ou "binaire")
 - installateur : attention aux scripts qui "en mettent partout" !
 - archive
 - package Debian/Ubuntu à télécharger et à installer "à la main"
 - packages Debian/Ubuntu dans un dépôt de packages
installation et m à j automatiques avec `apt`
 - format `ApplImage`

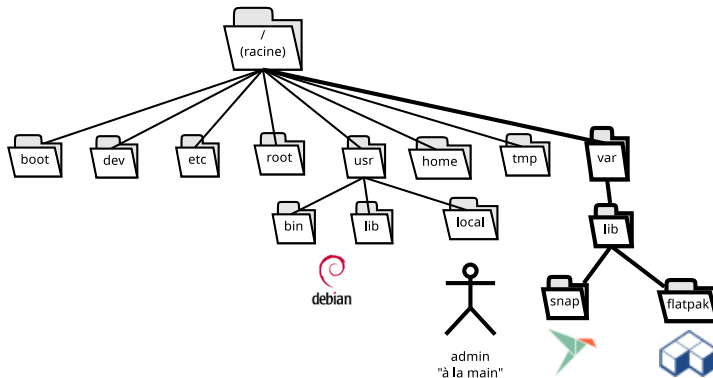
Autres magasins d'applications

- Deux magasins : `snap` et `flatpak`
- Fonctionnent sur (presque) toutes les distributions
- Applications qui embarquent leurs propres bibliothèques
- Utilise les mêmes techniques que les conteneurs
- Les logiciels tournent souvent dans une *sandbox*
→ isolation du logiciel pour une meilleure sécurité du système en cas de faille de sécurité ou de *malware*
- Occupent plus d'espace disque

Schéma complet



Emplacements réservés



Plan du cours

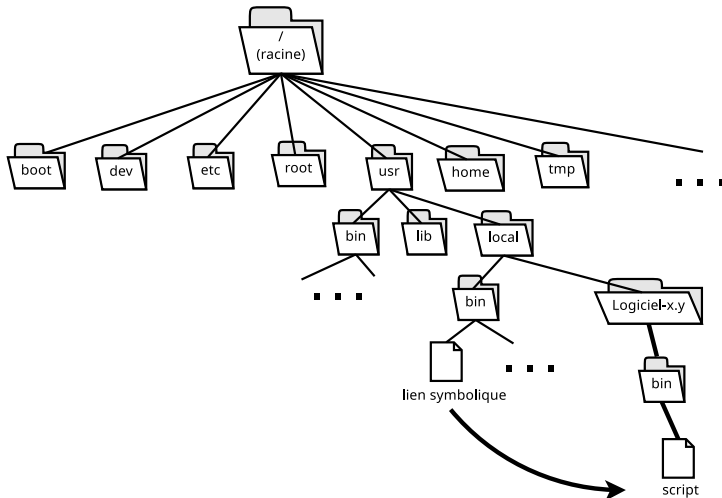
- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives**
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak
- 7 Résumé

Rappels sur les archives

- Types d'archives
 - Archives ZIP
 - Archives TAR + compression gzip, bzip2, xz, zstd
 - Archives 7ZIP
 - ...
- Extraction
 - gestionnaire de fichiers (Dolphin)
 - unzip
 - tar avec des paramètres

Emplacement d'installation

Répertoire `/usr/local/`



Cycle de vie du logiciel

- Lancement

- Les archives contiennent un script de lancement
- On peut lancer ce script en tapant son chemin absolu
`/usr/local/Logiciel-x.y/bin/script`
- On peut créer un lien symbolique dans `/usr/local/bin/`

- Installation d'une autre version en parallèle

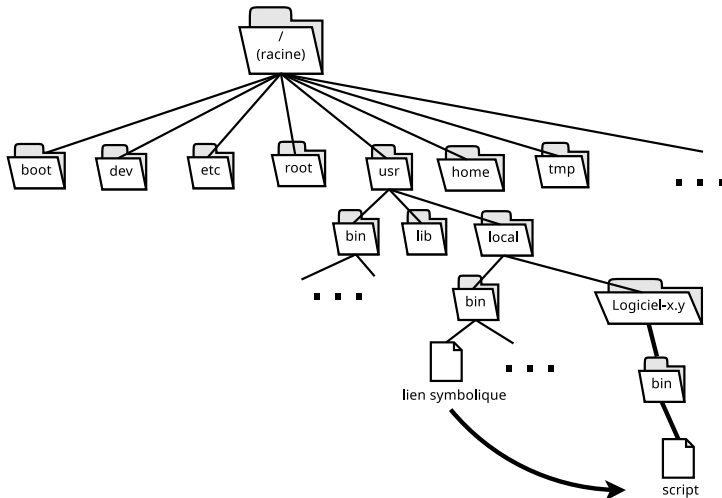
- autre répertoire à côté du 1er
- autre lien symbolique de lancement (avec n° de version)

- Désinstallation

- supprimer un répertoire et un lien symbolique

Script de lancement et lien symbolique

Répertoire `/usr/local/bin/`



Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak**
- 5 Snap
- 6 Flatpak
- 7 Résumé

Problèmes avec les magasins des distributions Linux

- Une distribution Linux, même très grosse, n'inclut pas tous les logiciels existants
- Un package pour une distribution X ne peut en général pas être installé sur une distribution Y
- Problème de compatibilité des logiciels avec une distribution
 - Langage comme Java, Python : pas (trop) de problèmes
 - Langage comme C, C++ : problèmes de compatibilité binaire avec les bibliothèques du système
démonstrations avec `ldd` et `ls, cp, evince`
- Un éditeur souhaitant diffuser un logiciel pour Linux doit le faire
 - pour plusieurs distributions
 - pour chaque version de distribution
 - ou inclure les bibliothèques utilisées dans le logiciel

Principe des magasins d'application snap et flatpak

- Quand on installe le 1er logiciel, une mini-distribution Linux est téléchargée et installée
- Les logiciels présents dans ces magasins sont prévus pour fonctionner sur cette mini-distribution
- Un éditeur n'a besoin de fournir son logiciel que pour le magasin snap et/ou flatpak
- Le logiciel fonctionnera sur (presque) toutes les distributions Linux
- snap et flatpak occupent plus d'espace de stockage qu'un package Debian/Ubuntu

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap**
- 6 Flatpak
- 7 Résumé

Présentation

- Projet développé par Canonical (éditeur de Ubuntu)
- Logiciels distribués à partir du site `snapcraft.io`
- Seul ce site, contrôlé par Canonical, peut être utilisé : centralisation
- Des interfaces graphiques existent pour gérer les packages
- Un logiciel tourne en tâche de fond : `snapd` (un "démon")
- Un logiciel sert d'interface en ligne de commande : `snap`
- Tout est installé dans le répertoire `/var/lib/snap/`

Commandes usuelles

- Chercher un package
`$ snap search MOT-CLÉ`
- Installer un package
`# snap install PACKAGE`
- Exécuter un logiciel
`$ /snap/bin/NOM-DU-LOGICIEL`
- Mettre à jour tous les packages snap
`# snap refresh`
- Lister les packages installés
`$ snap list`
- Afficher les métadonnées d'un package
`$ snap info PACKAGE`

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak**
- 7 Résumé

Présentation

- Projet sponsorisé par RedHat (éditeur de RedHat et Fedora)
- Logiciels distribués à partir du site `flathub.org`
- D'autres sites existent : décentralisation
- Des interfaces graphiques existent pour gérer les packages
- Aucun logiciel ne tourne en tâche de fond
- Un logiciel sert d'interface en ligne de commande : `flatpak`
- Tout est installé dans le répertoire `/var/lib/flatpak/`
- Les noms d'un package contiennent la provenance du logiciel

Commandes usuelles

- Chercher un package
`$ flatpak search MOT-CLÉ`
- Installer un package
`$ flatpak install PACKAGE`
- Exécuter un logiciel
`$ flatpak run PACKAGE`
- Mettre à jour tous les packages snap
`$ flatpak update`
- Lister les packages installés
`$ flatpak list`
- Afficher les métadonnées d'un package
`$ flatpak info PACKAGE`
- NB : pas besoin de *shell* root, flatpak utilise un autre mécanisme d'élévation de privilèges

Plan du cours

- 1 Introduction
- 2 Méthodes de distribution des logiciels
- 3 Archives
- 4 Magasins d'applications snap et flatpak
- 5 Snap
- 6 Flatpak
- 7 Résumé**

Résumé

- Il existe plusieurs façons d'installer un logiciel (au moins 4)
- La façon principale est depuis sa distribution Linux
- Les autres façons peuvent être intéressantes