

Introduction aux systèmes GNU/Linux

S20E03 inetdoc.net



Philippe Latu / Université Toulouse 3

Document sous licence GNU FDL v1.3
<http://www.gnu.org/licenses/fdl.html>

Plan séance 3

- Séance 3 - Environnements graphiques - gestion de paquets
 - Décrire les caractéristiques des serveurs graphiques
 - Utiliser l'environnement graphique KDE
 - Identifier les caractéristiques d'un gestionnaire de paquets
 - Utiliser l'Advanced Package Tool Debian → apt
 - Gérer une collection de paquets
- Manipulations réalisables sur machines virtuelles
 - Dupliquer des jeux de paquets lors d'un clonage
 - Sélectionner différentes tâches avec tasksel ou aptitude
 - Tester les environnements graphiques

Environnements graphiques

- Historique X.org

- Consortium X-Window

- Projet historique de développement des interfaces graphiques Unix

- 1986 : première version diffusée par le MIT

- 1992 : début du projet XFree86

- <http://www.xfree86.org>

- Initialement dédié aux processeurs Intel i386

- Étendu au catalogue des processeurs du projet (

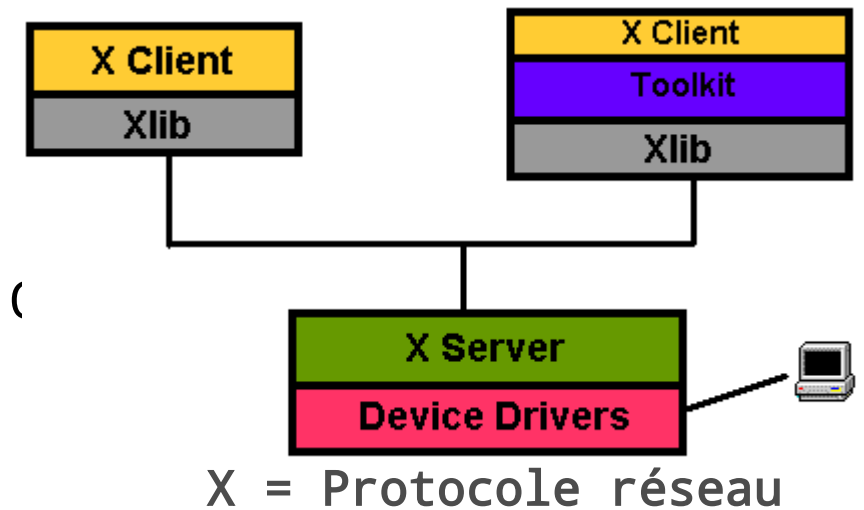
- 2004 : lancement de la fondation X.Org

- <http://www.x.org>

- Architecture modulaire

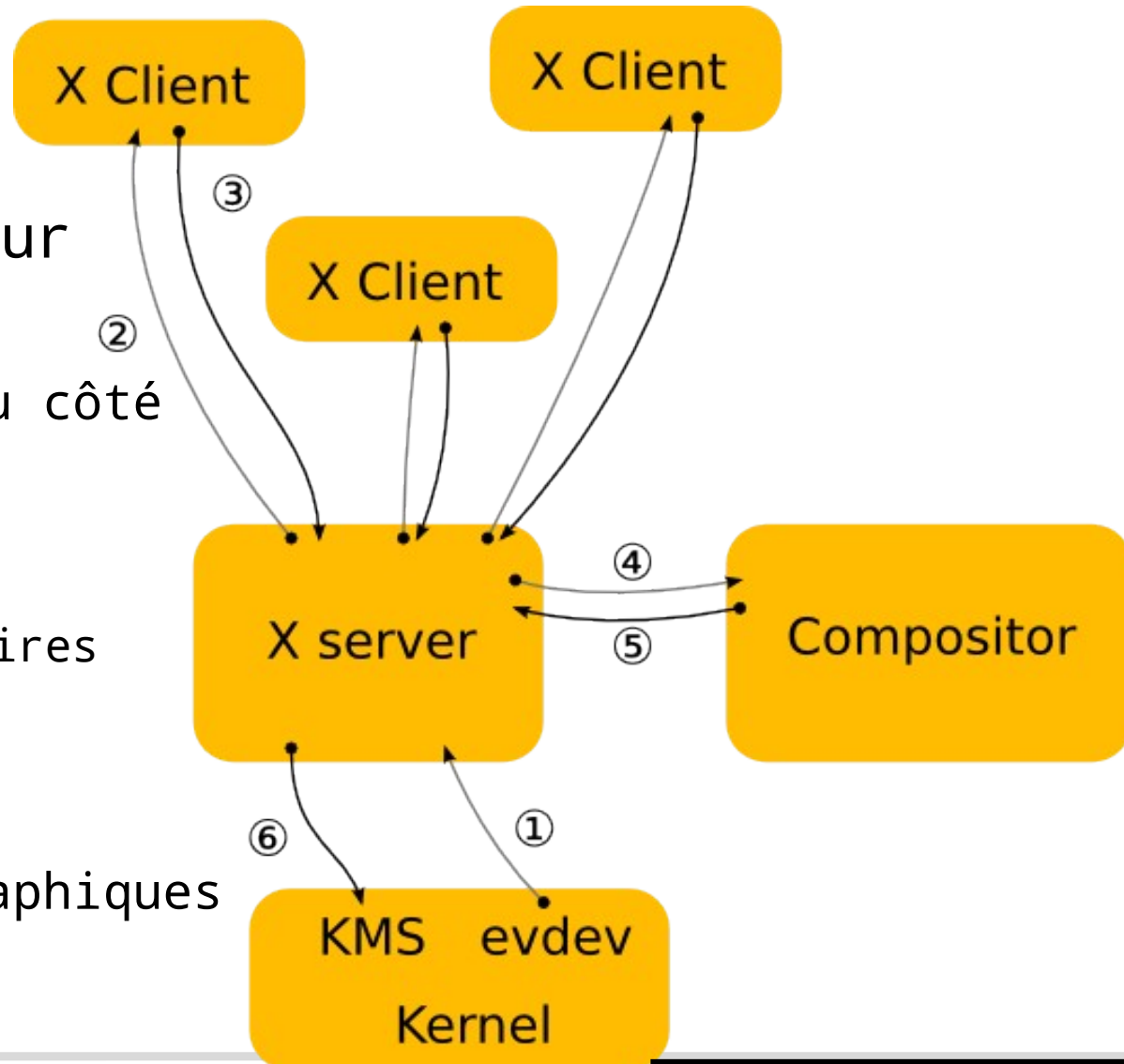
- 2008 : début du projet wayland

- Architecture «allégée» pour les outils mobiles



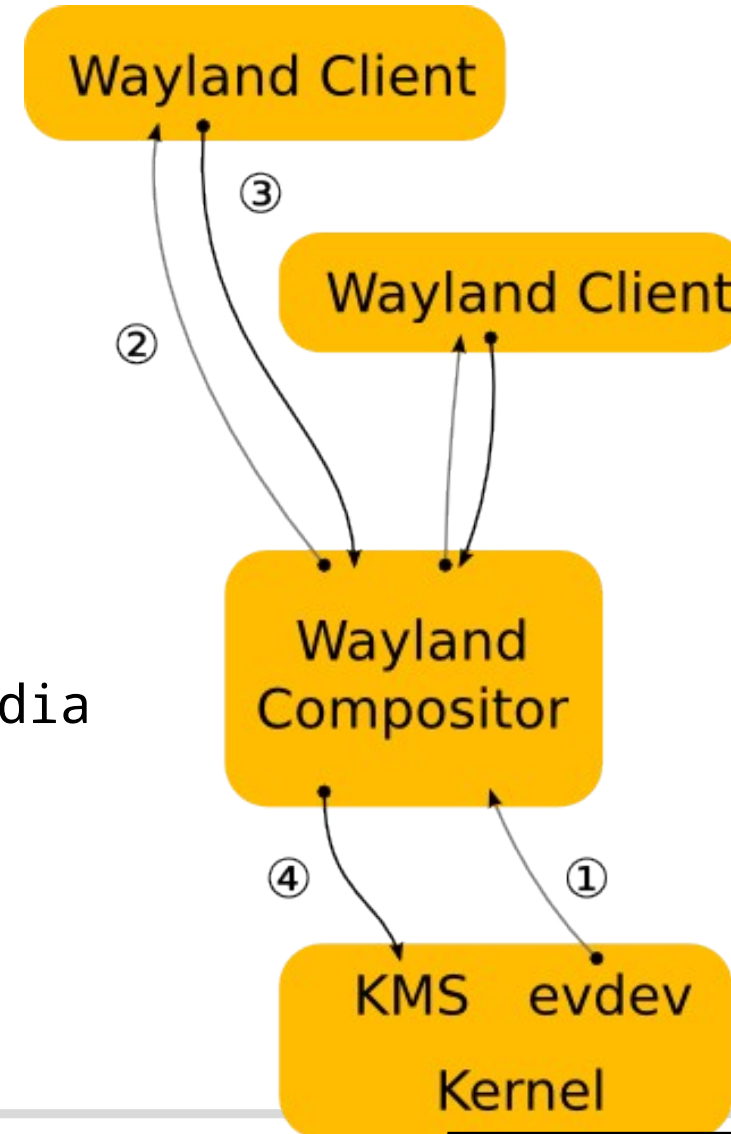
Environnements graphiques

- Architecture X.org
 - Source freedesktop.org
 - Architecture client/serveur
 - Serveur X = proxy
 - Tous les traitements ont lieu côté Compositor
- Solution modulaire
 - ⊕ Ajout de fonctions supplémentaires
 - ⊖ Accumulation dans le temps
- Fonctions réseau
 - Utilisation d'application graphiques à distance



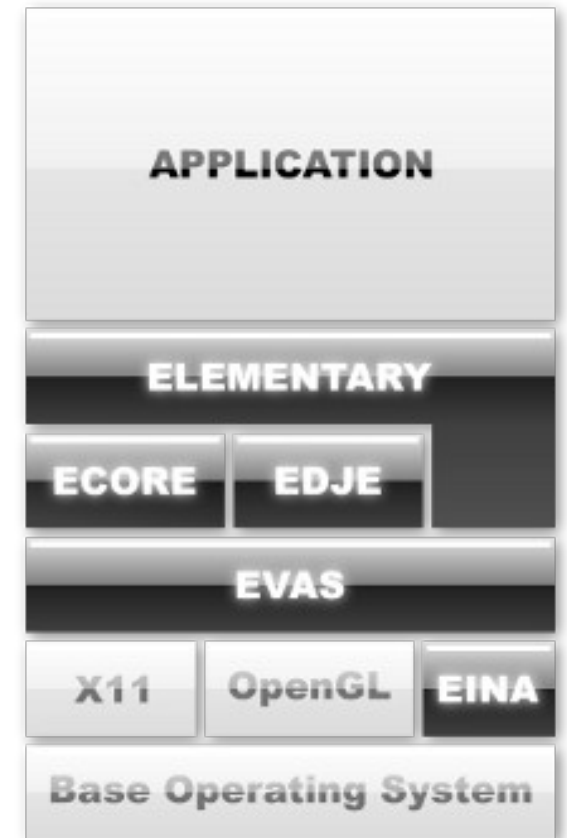
Environnements graphiques

- Architecture Wayland
 - Source freedesktop.org
 - Architecture «simplifiée»
 - Projet relativement jeune
 - Dispositifs mobiles
 - Solution adoptée par Ubuntu
 - Sponsors actifs : Qt, Intel
 - Pas encore de support matériel : NVidia



Environnements graphiques

- Processus de développement
 - Environnement graphique = chaîne de développement
 - Une architecture de base → Projets multiples
 - Modèle de développement OpenSource
 - ⊕ Introduction facile de nouvelles fonctionnalités
 - ⊖ Coordination difficile entre projets «concurrents»
- Quelques exemples
 - Enlightenment
 - <http://www.enlightenment.org/>
 - Bibliothèques Enlightenment Foundation Libraries (EFL)
 - Dispositifs mobiles et faible puissance de calcul



Environnements graphiques

- Quelques exemples (suite)

- LXDE & XFCE

- Solutions orientées faible puissance de calcul

- Systèmes embarqués, Live CD/DVD

- Xubuntu : <https://xubuntu.org/>

- Gnome & GTK

- Environnement : <http://www.gnome.org/>

- Bibliothèques : <http://www.gtk.org/>

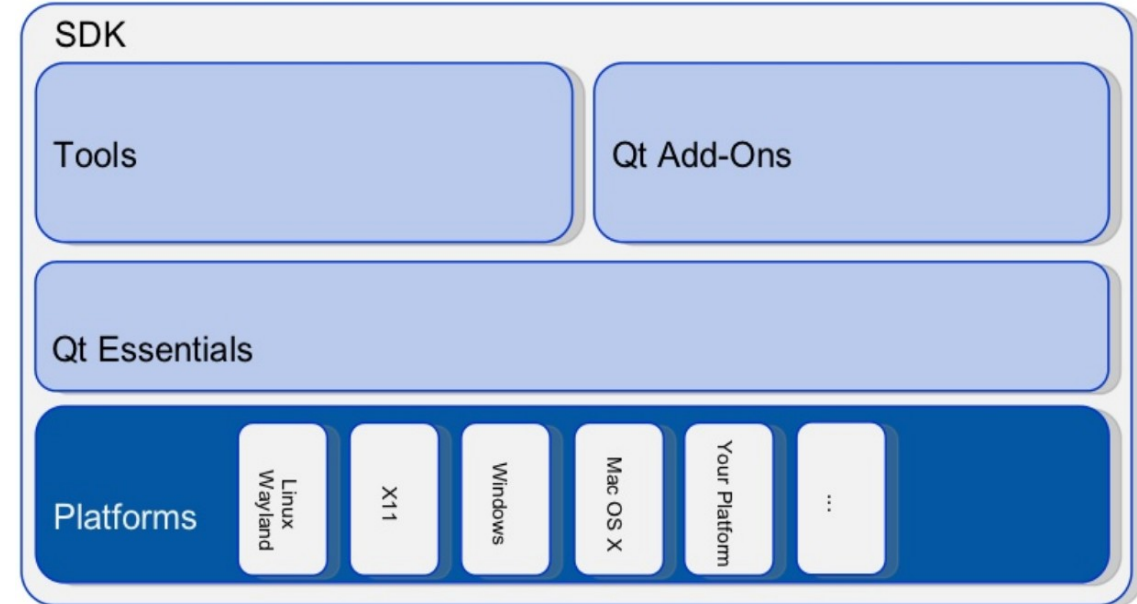
- RAD - glade : <http://glade.gnome.org/>

- KDE & Qt

- Environnement : <http://www.kde.org/>

- Bibliothèques : <http://qt.nokia.com/>

- RAD - qtcreeator : <http://qt-project.org/wiki/Category:Tools::QtCreator>



Gestion de paquets

- Distribution
 - **Noyau Linux + Shell(s) + collection de paquets**
- Gestionnaire de paquets
 - Construire le catalogue des paquets disponibles
 - Construire l'arbre de dépendances
 - Interroger l'index
 - Consulter les propriétés
 - Télécharger & installer depuis un **miroir ou dépôt**
- Bibliothèques partagées
 - Fonctions logicielles partagées entre applications
 - **Dépendance** → relation entre applications et bibliothèques

Gestion de paquets

- Paquets **binaires**

- Programmes compilés → exécution immédiate
- Formats principaux : rpm et **deb**
- Un paquet binaire par architecture supportée (i386|amd64)
- Code exécutable générique par famille de processeurs

- Paquets **sources**

- Programmes à compiler → temps d'installation plus long
- Formats principaux : ports, emerge et **deb-src**
- Code exécutable adapté au processeur cible

Gestion de paquets ou magasin d'applications ?

▪ Paquets

- Intégration au niveau système
 - Taille réduite
 - Dépendances → autres paquets
- Assurance qualité cohérente
 - Pilotage au niveau distribution
- Mises à jour contrôlées

▪ Snaps

- Indépendant du système
 - Taille plus importante
 - Dépendances → dans chaque snap
- Assurance qualité par snap
 - Responsable de publication
- Mises à jour automatiques

▪ Fragmentation de l'écosystème versus plate-forme

- Système d'exploitation
- SDK développeur
- Langage de conception
- Magasin d'applications

4 conditions à satisfaire
pour le développement d'une
plate-forme autonome

Advanced Package Tool

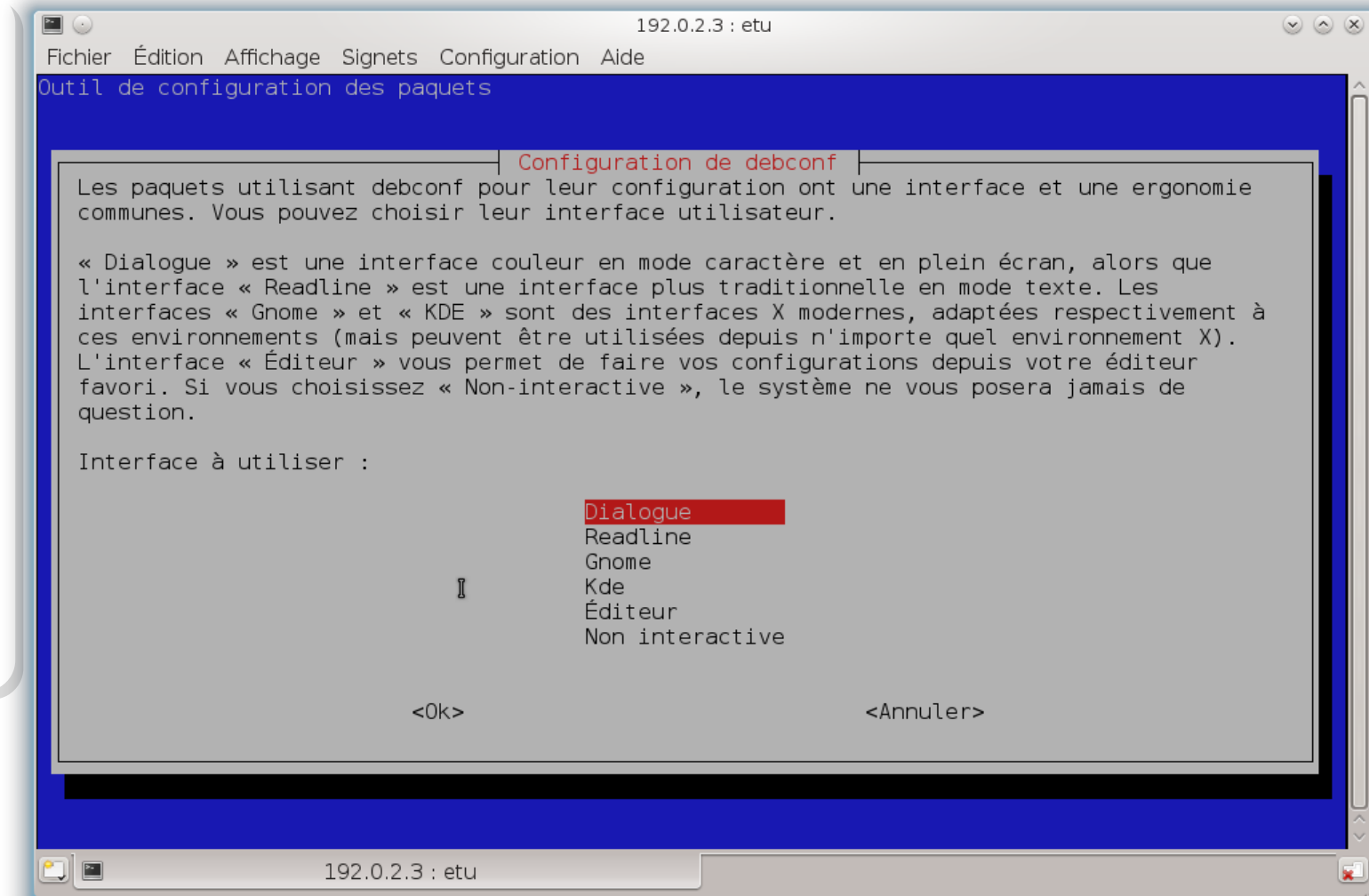
- APT : Advanced Package Tool
 - Gestion automatisée des dépendances depuis l'origine (1993)
 - APT → Bibliothèque utilisée par différentes applications
 - Concepts progressivement appliqués aux autres systèmes
 - Mises à jour continues & incrémentales
 - Installation unique pour toute la durée de vie d'un système
 - Stratégie établie suivant les branches de la distribution
 - Configuration préservée entre les mises à jour
 - Interface de configuration standardisée → `debconf`

Advanced Package Tool

debconf

- Gestionnaire de la configuration des paquets
- Plusieurs interfaces utilisateur disponibles
- Mémorisation des choix courants

```
# dpkg-reconfigure debconf
```



Advanced Package Tool

- Advanced Package Tool
 - Branches & catégories de paquets
 - 4 branches permanentes et indépendantes des versions principales
 - Branches
 - `stable` → paquets officiels
 - `testing` → paquets en attente d'intégration dans la version stable
 - `unstable` → paquets les plus récents en cours de test
 - `experimental` → paquets en cours de développement
 - Catégories
 - `main` → paquets conformes aux règles de définition du logiciel libre selon Debian
 - `contrib` → paquets de logiciels libres dépendant d'outils non libres
 - `non-free` → paquets avec des conditions de redistribution particulières

Advanced Package Tool

- Rythme des mises à jour suivant les branches
 - **stable**
 - Risque minimal → parfait pour les infrastructures critiques
 - Intégration des correctifs de sécurité sans évolution de version
 - Rythme d'évolution trop lent → <http://www.backports.org/>
 - **testing**
 - Risque moyen → convient bien pour les infrastructures de test
 - Intégration des correctifs de sécurité avec évolution de version
 - Rythme d'évolution satisfaisant pour couvrir tous les besoins
 - **unstable**
 - Risque élevé → «il faut assumer ses propres choix»
 - Pas de correctifs de sécurité
 - Rythme d'évolution le plus rapide

Advanced Package Tool

- Interfaces utilisateur de gestion des paquets
 - Console → `aptitude`
 - Utilisable dans tous les contextes
 - Ligne de commande → `apt` | `aptitude` | `apt-get` | `apt-cache`
 - Manipulations sur les paquets à partir du catalogue `réseau`
 - Ligne de commande → `dpkg` | `dpkg-reconfigure` | `dpkg-buildpackage`
 - Manipulations sur les paquets à partir du catalogue `local`

Advanced Package Tool

aptitude

- 'u' → update
- 'U' → marquer les paquets à mettre à jour
- 'g' 2 fois → valider la décision

```
Actions Annuler Paquet Solutions Rechercher Options Vues Aide
C-T: Menu ?: Help q: Quit u: Update g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.13 @ vm0
- \ Paquets pouvant être mis à jour (44)
--- admin      Utilitaires d'administration (installation de logiciels, gestion des utilisateurs, etc.) (3)
--- libs      Ensembles de routines logicielles (21)
--- localization  Language packs (3)
--- misc      Logiciels divers (3)
--- net       Programmes pour se connecter à différents services et les fournir (3)
--- perl      Interpréteur et bibliothèques Perl (1)
--- python    Interpréteur et bibliothèques Python (2)
--- utils     Divers utilitaires système (8)
--- Nouveaux paquets (75)
--- Paquets installés (462)
--- Paquets non installés (56829)
--- Paquets virtuels (18956)
--- Tâches (216)
```

Une version plus récente de ces paquets existe.

Ce groupe contient 44 paquets.

Advanced Package Tool

- Catalogue & Arbre des dépendances

- Fichier `/etc/apt/sources.list`

```
deb http://deb.debian.org/debian/ stable main contrib non-free
```

paquets
binaires

adresse
miroir

branche

catégories

```
deb-src http://ftp.fr.debian.org/debian/ stable main contrib non-free
```

paquets
sources

```
deb http://security.debian.org/ stable/updates main
```

adresse miroir
correctifs de sécurité

```
deb http://www.deb-multimedia.org stable main
```

adresse dépôt
non officiel

Advanced Package Tool

- Construction du catalogue
 - En mode console → `aptitude -u`
 - En ligne de commande → `aptitude update` | `apt update`

```
$ sudo apt update
Atteint :1 http://deb.debian.org/debian testing InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
```

Advanced Package Tool

- Mise à jour à l'échelle système
 - En ligne de commande → `aptitude safe-upgrade` | `apt upgrade`
 - Résolution complète des dépendances
 - Suppression de paquets installés interdite
 - En ligne de commande → `aptitude full-upgrade` | `apt full-upgrade`
 - Résolution complète des dépendances
 - Suppression de paquets installés en conflit avec un nouveau paquet possible
- Installation individuelle
 - En ligne de commande → `aptitude install <paquet>` | `apt install`
 - Proposition de solution en cas de conflit

Advanced Package Tool

- Recherches dans le catalogue des paquets
 - À partir du serveur web Debian
 - <http://packages.debian.org>
 - Recherche sur un nom de paquet
 - Recherche sur un nom de fichier appartenant à un paquet
 - Exemple : existe-t-il un paquet contenant le programme wireshark ?

Recherche dans les répertoires de paquets

Mot-clé :

Rechercher sur : Noms de paquets seulement Descriptions Noms de paquets-sources

N'afficher que les correspondances exactes:

Distribution : Section :

Voici quelques raccourcis pour certaines recherches :

- <http://packages.debian.org/nom> pour une recherche sur les noms des paquets.
- <http://packages.debian.org/src:nom> pour une recherche sur les noms des paquets source.

Recherche dans le contenu des paquets

Ce moteur de recherche ci vous permet de chercher dans le contenu de la distribution Debian en spe d'un paquet donné.

Mot-clé :

Afficher : les chemins se terminant par le mot-clé les paquets contenant un fichier de ce nom les paquets contenant un fichier dont le nom contient le mot-clé

Distribution : Architecture :

Voici un autre raccourci possible :

- <http://packages.debian.org/file:path> pour une recherche sur l'emplacement d'un fichier.

Advanced Package Tool

- Rechercher dans le catalogue des paquets
 - En ligne de commande → `aptitude search` | `apt search`
 - Exemple : existe-t-il un paquet ayant pour nom wireshark ?

```
$ aptitude -w 80 search ^wireshark
i   wireshark                - analyseur de trafic réseau – métapaquet
i   wireshark-common         - network traffic analyzer - common files
p   wireshark-dev            - analyseur de trafic réseau - outils de dev
...
```

Le nom débute
par wireshark

- Rechercher uniquement parmi les paquets installés
 - En ligne de commande → `aptitude search ~i`

```
$ aptitude -w 80 search ~iwireshark
i   libwireshark-data        - network packet dissection library -- data
i A libwireshark12          - network packet dissection library -- share
i   wireshark                - analyseur de trafic réseau – métapaquet
i   wireshark-common         - network traffic analyzer - common files
i A wireshark-qt             - analyseur de trafic réseau– versio
```

Affichage sur
80 caractères

Advanced Package Tool

- Rechercher dans le catalogue local des paquets
 - En ligne de commande → `dpkg -l`
 - Exemple : existe-t-il un paquet ayant pour nom wireshark ?

```
$ dpkg -l wireshark*
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder
| État=Non/Installé/fichier-Config/dépaqueté/échec-Config/H=semi-installé/W=attend-
traitement-déclenchements
|/ Err?=(aucune)/besoin Réinstallation (État,Err: majuscule=mauvais)
||/ Nom                Version             Architecture Description
+++-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
ii  wireshark            3.0.5-1            amd64        network traffic analyzer - meta-package
ii  wireshark-common    3.0.5-1            amd64        network traffic analyzer - common files
un  wireshark-doc       <aucune>           <aucune>     (aucune description n'est disponible)
un  wireshark-gtk       <aucune>           <aucune>     (aucune description n'est disponible)
ii  wireshark-gtk       3.0.5-1            amd64        network traffic analyzer - Qt version
```

État du
paquet

Version du
paquet
installé

Advanced Package Tool

- Rechercher dans le catalogue local des paquets
 - En ligne de commande → `dpkg -S`
 - Exemple : Quel est le paquet qui contient le programme wireshark ?

```
$ which wireshark
/usr/bin/wireshark
$ dpkg -S /usr/bin/wireshark
wireshark: /usr/bin/wireshark
```

Recherche du programme dans l'arborescence du système

Recherche dans les listes de fichiers des paquets installés

- Rechercher dans la liste des fichiers d'un paquet installé
 - En ligne de commande → `dpkg -L`
 - Exemple : Quel est le binaire contenu dans le paquet wireshark installé ?

```
$ dpkg -L wireshark | grep bin/
/usr/bin/wireshark
```

Advanced Package Tool

- Affichage des méta-données d'un paquet
 - En ligne de commande → `aptitude show` | `apt show`
 - Exemple : quelles sont les dépendances du paquet `apache2` ?

```
$ aptitude show apache2
Paquet : apache2
Version : 2.4.41-1
État: non installé
Priorité : optionnel
Section : httpd
Responsable : Debian Apache Maintainers <debian-apache@lists.debian.org>
Architecture : amd64
Taille décompressée : 617 k
Dépend: apache2-bin (= 2.4.41-1), apache2-data (= 2.4.41-1), apache2-utils (= 2.4.41-1), lsb-base, mime-support, perl:any, procps
Pré-dépend: dpkg (>= 1.17.14)
Recommande: ssl-cert
Suggère: apache2-doc, apache2-suexec-pristine | apache2-suexec-custom, www-browser
Est en conflit: apache2.2-bin, apache2.2-common
Casse: libapache2-mod-proxy-uwsgi (< 2.4.33)
Remplace: apache2.2-bin, apache2.2-common, libapache2-mod-proxy-uwsgi (< 2.4.33)
Fournit: httpd, httpd-cgi
```

État du paquet

Liste des dépendances

Advanced Package Tool

- Suppression d'un paquet installé
 - En ligne de commande → `aptitude remove` | `apt remove`
 - **Supprime** les fichiers binaires **mais conserve** la configuration locale
 - Exemple : Comment supprimer le paquet apache2 ?

```
$ sudo aptitude remove apache2
Les paquets suivants seront ENLEVÉS :
  apache2 apache2-bin{u} apache2-data{u} apache2-utils{u} libapr1{u} libaprutil1{u}
  libaprutil1-dbd-sqlite3{u} libaprutil1-ldap{u}
0 paquets mis à jour, 0 nouvellement installés, 8 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 0 o d'archives. Après dépaquetage, 7 607 ko seront libérés.
Voulez-vous continuer ? [Y/n/?]
```

Suppression des paquets
installés automatiquement

```
$ sudo aptitude search ^apache2 | egrep -e '^(p|c)'
```

c	apache2	-	Serveur HTTP Apache
p	apache2-bin	-	Serveur HTTP Apache (modules et autres fichiers binaires)
p	apache2-data	-	Serveur HTTP Apache - fichiers communs

État du paquet après
suppression

Advanced Package Tool

- Purge d'un paquet installé
 - En ligne de commande → `aptitude purge` | `apt purge`
 - **Supprime** les fichiers binaires **et** la configuration locale
 - Exemple : Comment purger le paquet wireshark ?

```
$ sudo aptitude purge wireshark
Les paquets suivants seront ENLEVÉS :
  wireshark{p} wireshark-qt{u}
0 paquets mis à jour, 0 nouvellement installés, 2 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 0 o d'archives. Après dépaquetage, 8 289 ko seront libérés.
```

Suppression des paquets installés automatiquement

```
$ sudo aptitude search ^wireshark
p   wireshark                - analyseur de trafic réseau – métapaquet
i   wireshark-common        - network traffic analyzer - common files
p   wireshark-dev           - analyseur de trafic réseau - outils de développement
p   wireshark-doc           - analyseur de trafic réseau - documentation
p   wireshark-gtk           - transitional dummy package
p   wireshark-qt            - analyseur de trafic réseau– version Qt
```

État du paquet après suppression

Advanced Package Tool

- Suppression & purge d'un paquet individuel avec dpkg
 - Suppression en ligne de commande → `dpkg --remove`
 - Purge en ligne de commande → `dpkg --purge`

Installé &
configuré

```
$ sudo dpkg -l nano | egrep '^(i|r|u)'  
ii nano          4.4-1          amd64          small, friendly text editor inspired by  
Pico
```

```
$ sudo dpkg --remove nano  
Suppression de nano (4.4-1) ...
```

Supprimé &
encore
configuré

```
$ sudo dpkg -l nano | egrep '^(i|r|u)'  
rc nano          4.4-1          amd64          small, friendly text editor inspired by  
Pico
```

```
$ sudo dpkg --purge nano  
Purge des fichiers de configuration de nano (4.4-1) ...
```

Ni installé
Ni configuré

```
$ sudo dpkg -l nano | egrep '^(i|r|u)'  
dpkg-query: aucun paquet ne correspond à nano
```

Advanced Package Tool

- Duplication du jeu de paquets d'un système à l'autre
 - Installation «optimale» → processus long
 - Duplication de la liste des paquets installés sans les configurations
 - Évolution/Migration d'une architecture à l'autre (ex. i386 → amd64)
- Sur le système **source**

```
$ aptitude search "?installed?not(?automatic)" -F %p > package-list.txt
```

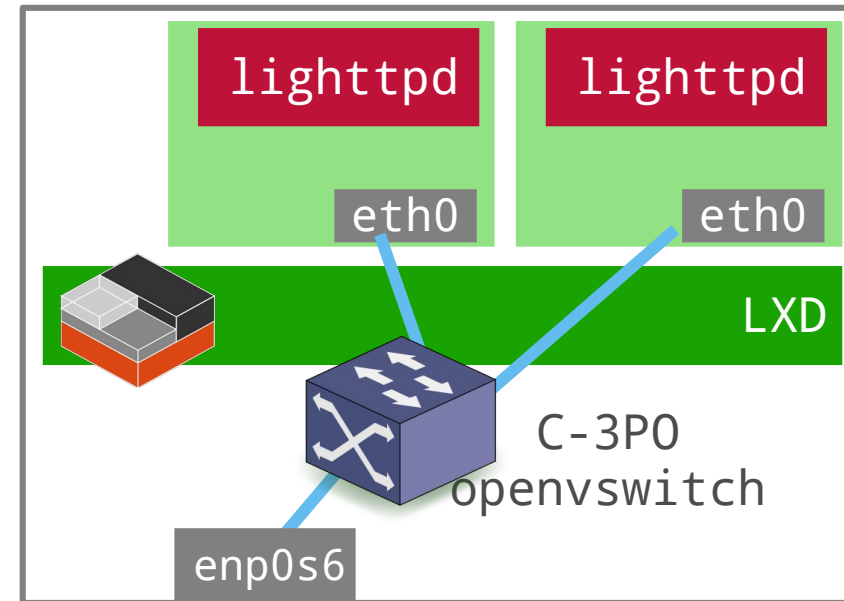
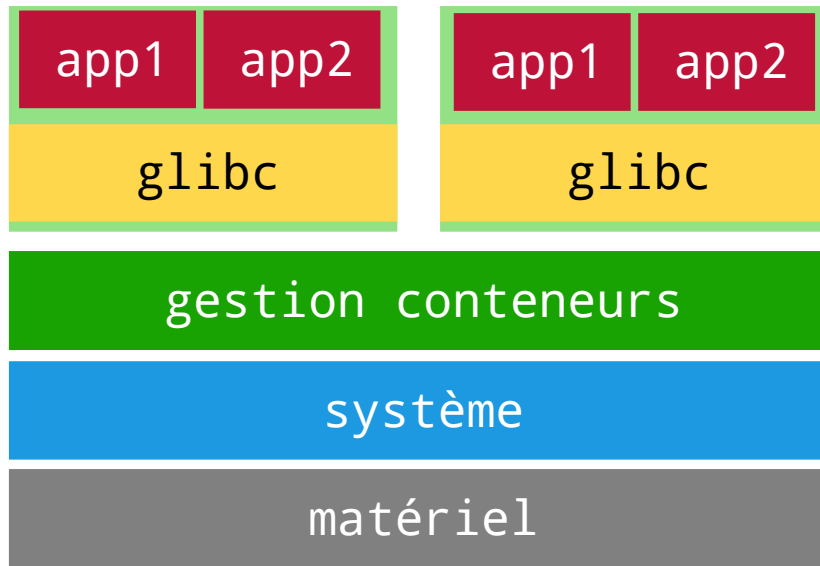
- Sur le système **cible**

```
# aptitude install $(cat package-list.txt | tr '\n' ' ')
```

Application → Opération C-3PO

▪ Objectifs

- Transformer le système hôte en commutateur/routeur réseau
- Installer de gestionnaire de conteneurs LXD
- Créer plusieurs conteneurs



Application → Opération C-3P0

- Installation des paquets, du service snapd et de LXD
 - Installer les paquets openvswitch-switch et snapd
 - Installer de gestionnaire de conteneurs LXD
 - Attention !
 - L'utilisateur 'etu' doit appartenir au groupe système 'lxd'
 - La liste des chemins de recherche d'applications doit contenir '/snap/bin'

```
etu@vm0:~$ sudo apt install openvswitch-switch snapd
etu@vm0:~$ sudo snap install lxd
etu@vm0:~$ sudo adduser etu lxd
Déconnexion / Reconnexion

etu@vm0:~$ id | grep -o lxd
lxd
etu@vm0:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/snap/bin
```

Application → Opération C-3P0

▪ Interconnexion réseau

1. Éditer le fichier
/etc/network/interfaces
2. Redémarrer la machine virtuelle
3. Vérifier l'état des interfaces

```
$ ip addr ls
```

```
etu@vm0:~$ cat /etc/network/interfaces
# This file describes the network
# interfaces available on your system
# and how to activate them.
# For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto C-3P0
iface C-3P0 inet dhcp
        ovs_type OVSBridge
        ovs_ports enp0s6

allow-C-3P0 enp0s6
iface enp0s6 inet manual
        ovs_bridge C-3P0
        ovs_type OVSPort
        up ip link set dev $IFACE up
        down ip link set dev $IFACE down
```

Application → Opération C-3P0

- Gestion des conteneurs
 - Configuration initiale
 - Raccordement au commutateur C-3P0

```
etu@vm0:~$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (btrfs, dir, lvm, ceph) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=13GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]: no
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: C-3P0
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:

etu@vm0:~$ lxc profile device set default eth0 nictype bridged
```


Application → Opération C-3P0

- Gestion des conteneurs
 - Téléchargement des images et lancement des conteneurs
 - Raccordement au commutateur C-3P0

```
etu@vm0:~$ lxc launch images:debian/bullseye c0
```

```
Creating c0
```

```
Starting c0
```

```
etu@vm0:~$ lxc launch images:ubuntu/focal c1
```

```
Creating c1
```

```
Starting c1
```

```
etu@vm0:~$ lxc ls
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
c0	RUNNING	172.16.100.192 (eth0)	2001:678:3fc:da:216:3eff:fe8f:34df (eth0)	CONTAINER	0
c1	RUNNING	172.16.100.193 (eth0)	2001:678:3fc:da:216:3eff:fe12:7501 (eth0)	CONTAINER	0

```
etu@vm0:~$ for num in {0..1}; do lxc config set c$num security.nesting true; done
```

```
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- apt update; done
```

Application → Opération C-3P0

- Configuration des conteneurs
 - Création d'un compte utilisateur
 - Ouverture de l'accès SSH

```
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- adduser --disabled-password --gecos "" etu ; done
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- /bin/bash -c "echo 'etu:-etu-' | chpasswd"; done
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- apt install ssh; done
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- sed -i 's/#Port 22/Port 2222/g' /etc/ssh/sshd_config; done
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- grep Port /etc/ssh/sshd_config; done
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- systemctl restart ssh; done
```

```
phil@Ubuntu4STRI:~$ ssh -p 2222 etu@172.16.100.192
The authenticity of host '[172.16.100.192]:2222 ([172.16.100.192]:2222)' can't be established.
ECDSA key fingerprint is SHA256:nwu/hATe0rYxq4rcVT6Ll17PnaGs1l9Dp0Qd+i3XZvQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.16.100.192]:2222' (ECDSA) to the list of known hosts.
etu@172.16.100.192's password:
Linux c0 5.9.0-4-amd64 #1 SMP Debian 5.9.11-1 (2020-11-27) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
etu@c0:~$
```

Application → Opération C-3P0

- Ajout d'un service Web
 - Installation du paquet `lighttpd`
 - Accès à la page Web de chaque conteneur via un tunnel `SSH`

```
etu@vm0:~$ for num in {0..1}; do lxc exec c$num -- apt install lighttpd; done
```

```
phil@Ubuntu4STRI:~$ ssh -p 2222 -L 8000:localhost:80 etu@172.16.100.192
```

```
phil@Ubuntu4STRI:~$ wget -O /dev/null http://localhost:8000
--2020-12-06 11:13:31-- http://localhost:8000/
Résolution de localhost (localhost)... 127.0.0.1
Connexion à localhost (localhost)|127.0.0.1|:8000... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 3388 (3,3K) [text/html]
Enregistre : «/dev/null»
```

```
/dev/null 100%[=====>] 3,31K --.-KB/s ds 0s
```

```
2020-12-06 11:13:32 (141 MB/s) - «/dev/null» enregistré [3388/3388]
```

Bilan séance 3

- Environnements Graphiques
 - Interfaces utilisateur + chaînes de développement
 - Évolutions importantes côté dispositifs mobiles
 - Social Desktop → nextcloud.com
- Gestionnaire de paquets → Advanced Package Tool
 - Base de l'administration système
 - Bibliothèques partagées entre applications
 - Notion de dépendance
 - Gestion automatisée des relations entre applications et bibliothèques
 - Gestion autonome des configurations
 - Gestion automatisée des correctifs de sécurité

