

<ÉCOLE DE LA FILIÈRE NUMÉRIQUE/>

#NANTES #SAINT-NAZAIRE #ANGERS

#RENNES #LE MANS #LAVAL #CAEN #PARIS...

imie

Introduction

- Qu'est ce qu'un logiciel libre
- UNIX et les Unix-like
- GNU / Linux

La distribution Debian

- Premiers pas avec Debian
- Installation
- Présentation de l'architecture

Les commandes Linux

- Le Shell et le prompt
- Les commandes de base
- Les utilitaires

Administration Linux

- Gestion des paquets
- Gestion des utilisateurs et des groupes
- Gestion des droits

N° Fiche AT	Activités types	N° Fiche CP	Compétences professionnelles
1	Assister les utilisateurs en centre de services	1	Mettre en service un équipement numérique
		2	Assister les utilisateurs sur leurs équipements numériques
		3	Gérer les incidents et les problèmes
		4	Assister à l'utilisation des ressources collaboratives
2	Maintenir, exploiter et sécuriser une infrastructure centralisée	5	Maintenir et exploiter le réseau local et la téléphonie
		6	Sécuriser les accès à Internet
		7	Maintenir et exploiter un environnement virtualisé
		8	Maintenir et exploiter un domaine Active Directory et les serveurs Windows
		9	Maintenir et exploiter un serveur Linux
3	Maintenir et exploiter une infrastructure distribuée et contribuer à sa sécurisation	10	Configurer les services de déploiement et de terminaux clients légers
		11	Automatiser les tâches à l'aide de scripts
		12	Maintenir et sécuriser les accès réseaux distants
		13	Superviser l'infrastructure
		14	Intervenir dans un environnement de Cloud Computing
		15	Assurer sa veille technologique

- Installer et personnaliser le système Linux en fonction du rôle du serveur
- Gérer les utilisateurs et les droits
- Effectuer les tâches d'administration en ligne de commandes
- **Installer et configurer les services serveurs http, base de données et php (TP maison)**
- Échanger sur les forums en anglais
- Connaissance de la culture Unix/Linux
- Connaissance des familles de distributions

- Savoir-faire
 - Installer et personnaliser le système Linux en fonction du rôle du serveur
 - Gérer les utilisateurs et les droits
 - Effectuer les tâches d'administration en ligne de commandes
- Savoir-faire relationnels
 - Échanger sur les forums en anglais
- Savoirs
 - Connaissance de la culture Unix/Linux
 - Connaissance des familles de distributions

Introduction

- Qu'est ce qu'un logiciel libre :
 - Un programme informatique est constitué de nombreuses lignes de code, compilés en un fichier exécutable.
 - Un programme pourra s'exécuter sans connaissance du code source.
 - Il est difficile de remonter aux sources du programme à partir du binaire (exécutable).

- Les logiciels propriétaires sont des logiciels dont une licence, souvent payante, ne donne qu'un droit limité d'utilisation.
- Les logiciels libres sont les logiciels que l'on peut librement utiliser, échanger, étudier et redistribuer.
 - Recherche rapide logiciels freeware

Unix-Like

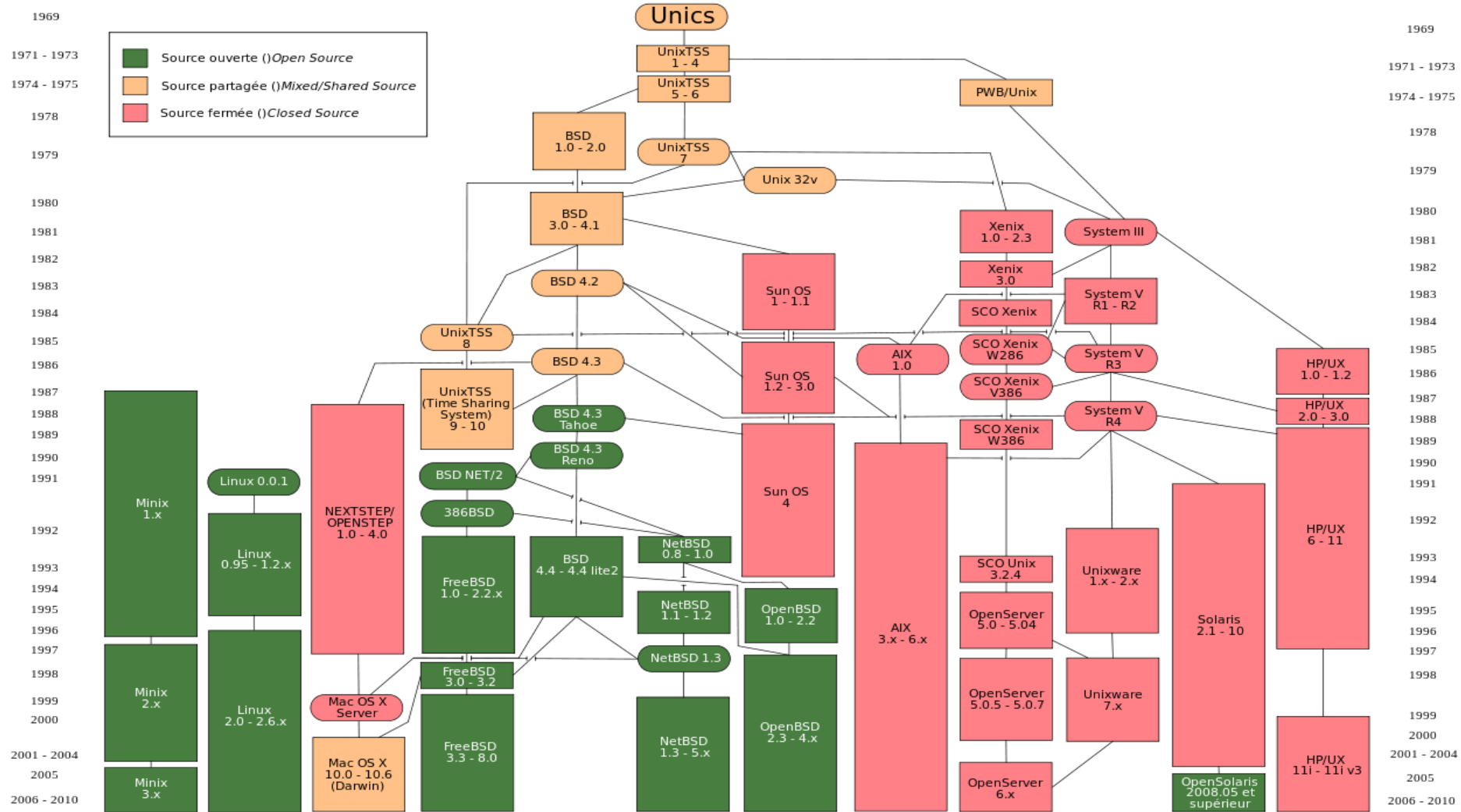
- UNIX et les Unix-like :
- UNIX est un système d'exploitation, créé en 1969 par Ken Thompson et Dennis Ritchie dans les laboratoires de la société AT&T.
- Premier système basé sur une API écrite dans un langage «évolué » et non en assembleur
- Distribué jusqu'en 1984
- Adapté par de nombreuses
- entreprises, qui ont développé des
- dérivés d'UNIX ...



- D'autres systèmes d'exploitation ont ensuite été conçus en s'inspirant des principes d'UNIX, voire même en recherchant une compatibilité complète avec UNIX.
- Le nom UNIX étant une marque de certification, on parle alors plutôt d'UNIX-like.
- Parmi ces systèmes, les plus connus sont :
 - GNU/Linux,
 - les systèmes BSD,
 - Minix ou encore Mac OS X.

- Qu'est ce qu'un logiciel libre :
 - Un programme informatique est constitué de nombreuses lignes de code, compilés en un fichier exécutable.
 - Un programme pourra s'exécuter sans connaissance du code source.
 - Il est difficile de remonter aux sources du programme à partir du binaire (exécutable).

UNIX-like



Projet GNU

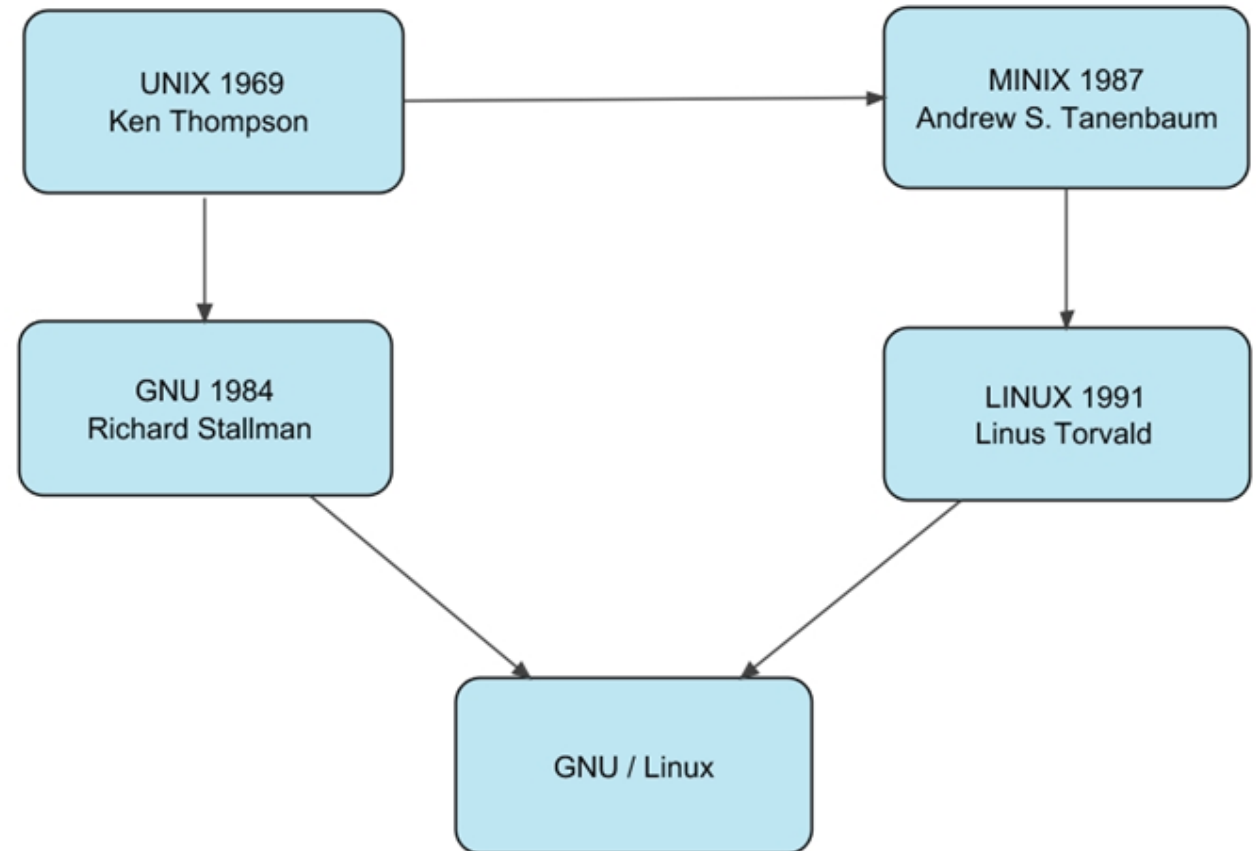
- Le projet GNU :
 - Scandalisé par les restrictions imposées par les logiciels propriétaires, **Richard Stallman** lance en 1983, le projet GNU.
 - Ce projet doit permettre de développer un système d'exploitation libre complet et inspiré d'UNIX.
- Le but étant de contrer le développement croissant des logiciels propriétaires.



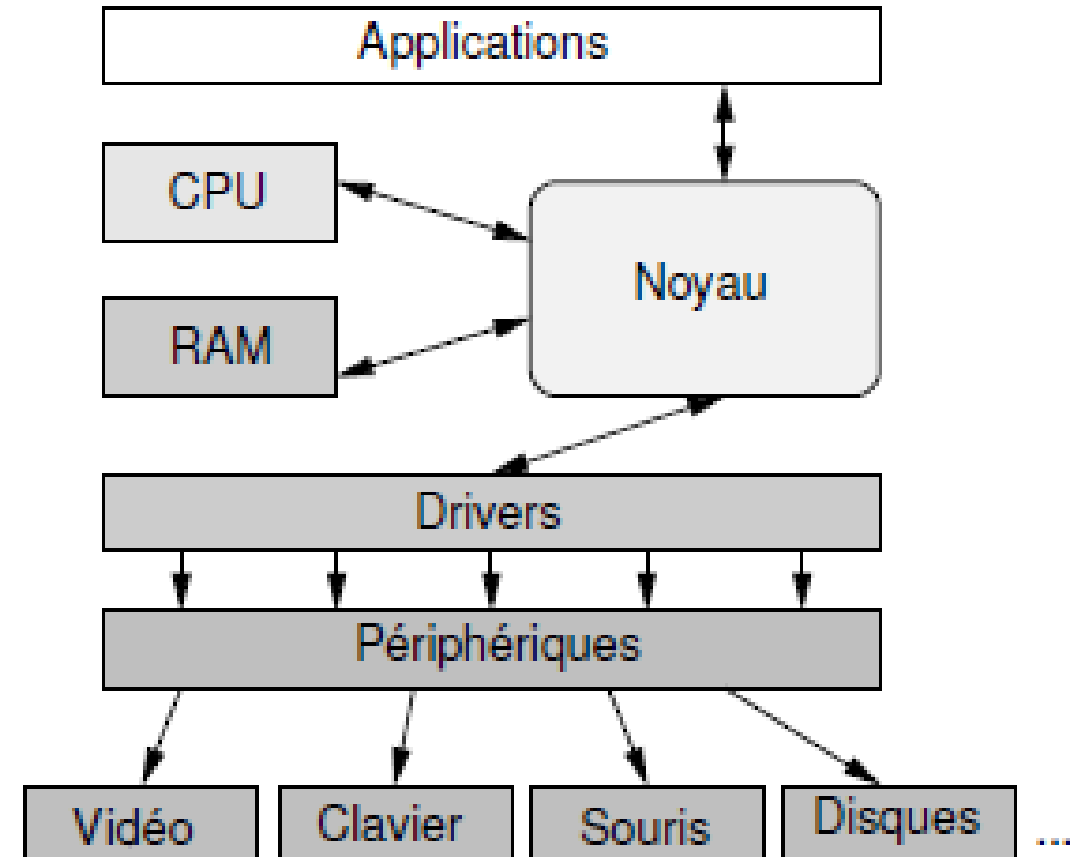
- Richard Stallman fonde l'association Free Software Foundation et écrit le GNU Manifesto.
- Libertés fondamentales du logiciel libre :
 - La liberté d'exécution,
 - La liberté de modification,
 - La liberté de redistribution,
 - La liberté d'amélioration.

Projet GNU / Linux

- Ce système peut se découper en 3 entités :
 - Le noyau
 - Les utilitaires
 - Les fichiers système.



- Le noyau :
 - Le noyau gère :
 - la mémoire,
 - l'accès aux périphériques (disque dur, cartes son, réseau, ...),
 - la circulation des données sur le bus,
 - les droits d'accès,
 - les multiples processus qui correspondent aux multiples tâches que l'ordinateur doit exécuter en même temps



- Les utilitaires :
 - Des interpréteurs de commande, appelés Shells ;
 - Des commandes de manipulation de fichiers (copie, déplacement, effacement, etc.) ;
 - Des commandes de gestion d'activités du système (processus) ;
 - Des commandes de communication entre utilisateurs ;
 - Des outils de développement (éditeurs de texte, compilateurs, débogueurs...) ;
 - Des outils de traitement de texte.

- Les fichiers système :
 - Des informations sur la configuration des services ;
 - Des scripts de changement d'état du système (démarrage, arrêt...).
 - Un système hiérarchisé de processus et une « génétique des processus » ;
 - Des points d'accès protégés aux services offerts par le noyau.

- Aspect multitâche renforcé ;
- Un système de fichier hiérarchisé (fichiers, répertoires, liens physiques et symboliques) ;
- Utilisation du concept de filtrage et de redirections.
- Les filtrages et les redirections constituent les principes de base permettant d'appliquer la philosophie de la boîte à outils d'UNIX : écrire des commandes élémentaires et homogènes, pouvant communiquer entre elles.

Les premiers pas ;)

Debian



PLATINUM MEMBERS

- Not
- l

et



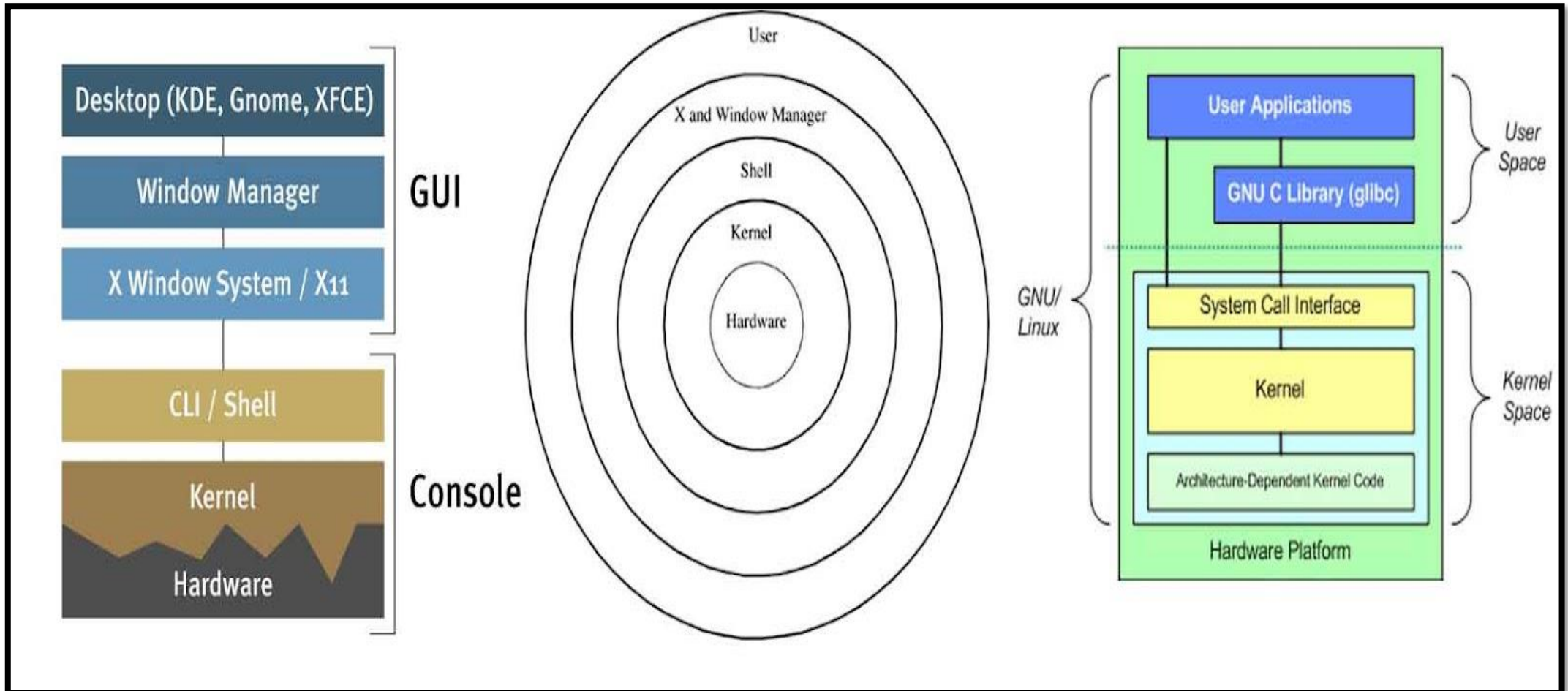
- Ok, mais pourquoi Debian ?
 - Pour ses qualités techniques ;
 - Debian GNU/Linux est utilisé par la plupart des fournisseurs d'accès à Internet, comme Free ;
 - Debian est reconnu pour son sérieux et ses fortes prises de positions dans le monde libre.

- Installation de Debian :
 - Il existe plusieurs types d'installations :
 - Amorçage par le réseau (en utilisant les protocoles PXE ou TFTP),
 - A partir de fichiers image, qui font jusqu'à 40 Mo.
 - A partir des images CD « netinstall » de taille réduite (jusqu'à 180 Mo) donc rapidement téléchargeable.
 - A partir des images complètes de CD ou DVD : utile sans connexion internet.

- Quelques bonnes pratiques :
 - Ne pas installer la partie graphique sur un serveur (No Desktop).
 - Spécifier le domaine dans lequel se trouvera le serveur.
 - Choisir ou définir le site miroir /!\
 - Ne pas sélectionner de pré-installation de services, ceux-ci seront installés par la suite.

- Quelques bonnes pratiques :
 - Ne pas installer la partie graphique sur un serveur (No Desktop).
 - Spécifier le domaine dans lequel se trouvera le serveur.
 - Choisir ou définir le site miroir /!\
 - Ne pas sélectionner de pré-installation de services, ceux-ci seront installés par la suite.

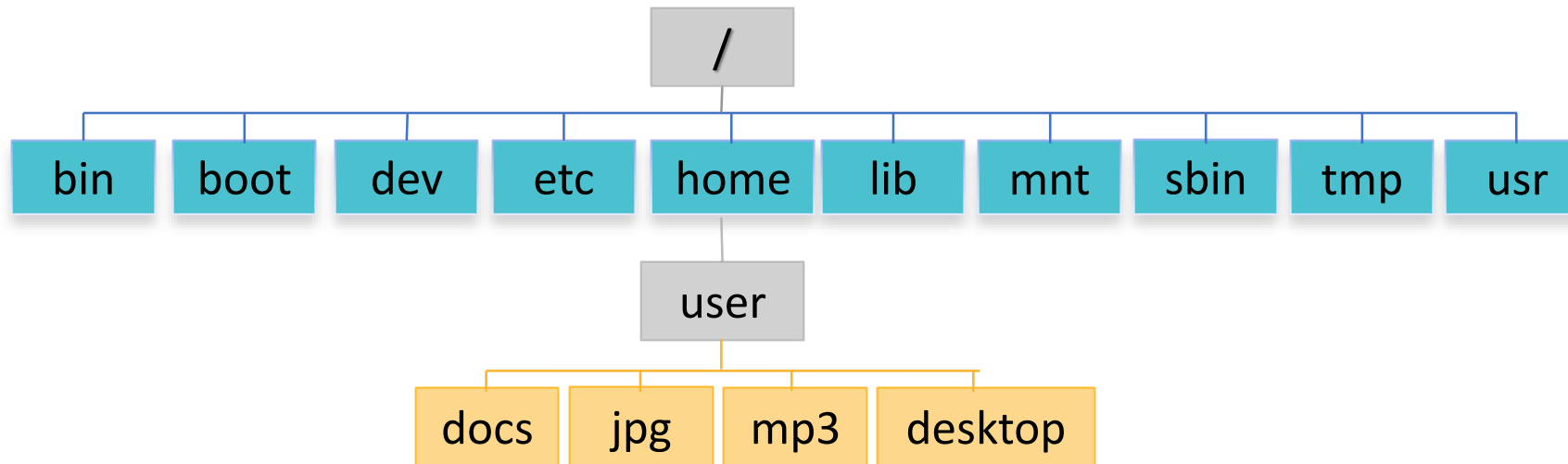
- Architecture:



- Particularités de ces OS :
 - UNIX/Linux est sensible à la casse, attention à vos commandes...
 - De même, les noms de fichiers et de répertoires sont également sensibles à la casse.
 - Les chemins sont séparés par des slash (et non des back-slashes) :
~~etc\rc.d\init.d\xfs~~ doit s'écrire `/etc/rc.d/init.d/xfs`

- Arborescence :
 - Les répertoires spéciaux :
 - . : représente le répertoire courant,
 - .. : représente le répertoire parent,
 - ~ : représente le répertoire maison de l'utilisateur
 - Les fichiers cachés :

Leur nom commence par un « . ». Exemple :
~/.bashrc est un fichier caché !



- **/** : Répertoire racine du système de fichiers. Tout le système, tous les fichiers sont situés en dessous (hiérarchiquement) de cette racine.
- **bin** : Répertoire contenant les principales commandes essentielles.
- **dev** : Dossier contenant tous les périphériques branchés sur l'ordinateur :
 - les lecteurs, les dispositifs branchés en USB,
 - les disques durs, ...

- **etc** : Dossier dans lequel sont rangés quasiment tous les fichiers de configurations. Il contient également les fichiers sources de dépôts pour les logiciels de gestions de paquets comme APT, RPM ou YUM.
- **home** : Contient les données de chaque utilisateur. Bien sûr, chaque utilisateur est « rangé » dans un répertoire associé à son login.
- **lib** : Répertoire contenant les bibliothèques.

- **mnt** : Ce dossier sert de répertoire pour monter les périphériques se trouvant dans /dev (hdd, cd/dvd, floppy etc...).
- **root** : Le répertoire personnel de root.
- **usr** : Un des répertoires les plus volumineux du système car il contient les fichiers liés aux utilisateurs, tels que :
 - tous les exécutables ainsi que les bibliothèques,
 - les documentations et les fichiers d'en-tête de ces exécutables.

- Notions de chemin :

- **Chemin absolu**

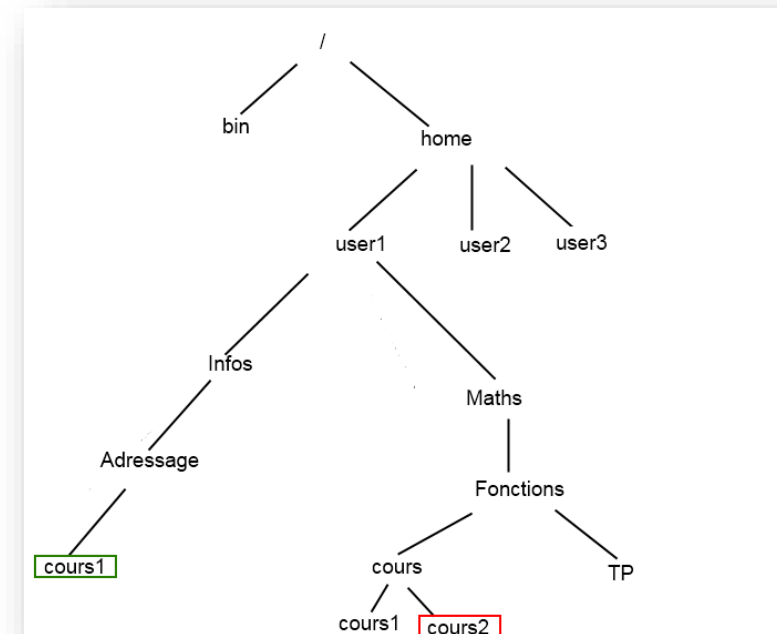
- Chemin complet depuis la racine de l'arborescence :

Ex. : /home/user1/infos

- **Chemin relatif**

- Chemin à partir d'un point quelconque de l'arborescence :

Ex. : en étant dans /home
./user1/infos



09/10/2018

35

Les commandes

Linux

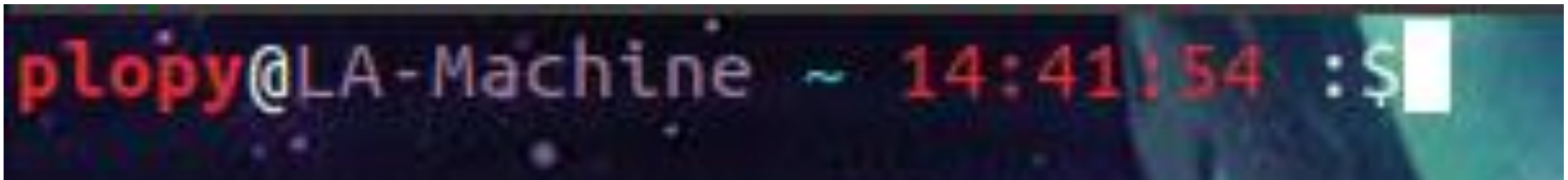
Les commandes



- Le Shell :
 - Le « Shell » est l'interpréteur de commande, il crée l'interface entre l'utilisateur et le système d'exploitation.
 - Son rôle consiste à :
 - lire la ligne de commande,
 - interpréter sa signification,
 - exécuter la commande,
 - retourner le résultat sur les sorties.
 - Le Shell utilisé sous Debian par défaut s'appelle BASH.

- Les commandes internes :
 - Il s'agit des commandes internes du Shell,
 - « help » affiche la liste des commandes internes et leur syntaxe.
- Les commandes externes :
 - **/bin** contient la liste des commandes externes usuelles communes à tous les utilisateurs.
 - **/sbin** contient la liste des commandes externes usuelles réservées à l'administrateur (root).

- Le prompt :
 - Au Login entrez l'utilisateur puis le mot de passe afin d'ouvrir une session.
 - Le prompt apparait suite à l'ouverture de session, il se compose de :
 - nom de l'utilisateur connecté
 - puis du nom de la machine
 - Puis du dossier courant
 - et enfin le \$ ou le # spécifient le type de l'utilisateur connecté.



```
pLopy@LA-Machine ~ 14:41:54 :$
```

- Changer d'utilisateur : ~# su
- Avec quel utilisateur suis-je connecté ? ~# whoami
- Fermer sa session : ~# exit
- Informations du système : ~# uname
(trouver une autre commande)
- Utilisation du manuel : ~# man [cmde]
 - **ENTREE ou ESPACE pour faire défiler le texte**
 - **touche « q » pour sortir de l'aide.**
- Afficher les utilisateurs connectés : ~# who
- Informations utilisateur : ~# id
- Afficher la date et l'heure : ~# date

- Déplacement dans l'arborescence :
~# cd
- Lister les fichiers et répertoires :
~# ls
- Créer un répertoire :
~# mkdir [rep]
- Effacer des fichiers :
~# rm
- Déplacer des fichiers et répertoires :
~# mv [src] [dst]
- Copier des fichiers et répertoires :
~# cp [src] [dst]

- Modifier l'horodatage d'un fichier : ~# touch
- Afficher le contenu d'un fichier texte : ~# less
~# more
~# cat
- Afficher le début d'un fichier : ~# head
- Afficher la fin d'un fichier : ~# tail

- Effacer l’affichage courant :
~# clear
- Historique des commandes :
~# history
- Recherche de mots dans un fichier :
~# grep
- Rechercher un fichier :
~# find
~# locate
~# whereis

- Afficher le répertoire en cours :
`~# pwd`
- Trier le contenu d'un fichier texte :
`~# sort`
- Lancer un script :
`~# ./[nom_script]`
- Voir les services en cours :
`~#service --status-all`
- Lister les processus en cours :
`~# ps`
- Afficher les proc sous arborescence :
`~# pstree`
- Tuer un processus :
`~# kill PID`
- Arrêt violent d'un processus :
`~# kill -9 PID`

- Les commandes peuvent être combinées sur une même ligne de commande via un « pipe » : |.
- Exemple :
 - `ps -ef | grep mysql`
 - Cette commande liste les processus mysql en cours d'exécution.
- Il est possible de rediriger la sortie standard des commandes dans un fichier :
 - Exemple : `ls -l /home/plopy > files_plopy.txt`

- > Écrit la sortie dans un fichier, **en écrasant celui-ci s'il existe déjà.**
- >> **Ajoute** la sortie de la commande à la fin du fichier (ou, crée le fichier s'il n'existe pas, **mais il n'écrase jamais de données**).
- 2> Écrit la sortie d'erreur dans un fichier, **en écrasant celui-ci s'il existe déjà.**
- 2>> **Ajoute** la sortie d'erreur à la fin du fichier (ou, crée le fichier s'il n'existe pas, **mais il n'écrase jamais de données**).

; Séquencement : La seconde opération ne commence qu'après la fin de la première.

```
#mkdir dossier ; touch dossier/test
```

& Parallélisme : La première opération est lancée à l'arrière-plan, et la seconde démarre simultanément.

```
#mkdir dossier1 & mkdir dossier2
```

&& Dépendance : La seconde opération n'est exécutée que si la première a renvoyé un code retour 0.

```
#mkdir dossier3 && echo "dossier créé "
```

|| Alternative : La seconde opération n'est exécutée que si la première a renvoyé un code retour différent de 0.

```
#mkdir dossier3 || echo "dossier non créé"
```

- Les REGEX = REGular Expression (expressions rationnelles courantes)

Caractère	Type	Explication
.	Point	n'importe quel caractère
[...]	crochets	classe de caractères : tous les caractères énumérés dans la classe
[^...]	crochets et circonflexe	classe complémentée : tous les caractères sauf ceux énumérés
	barre verticale	alternative - ou reconnaît l'un ou l'autre
*	astérisque	0, 1 ou plusieurs occurrences
?	interrogation	0 ou 1 occurrence

- Arrêt / relance du système :
 - Plusieurs commandes d'arrêt :
 - init 0
 - Halt
 - shutdown -h now
 - Plusieurs commandes de redémarrage :
 - init 6
 - reboot

09/10/2018

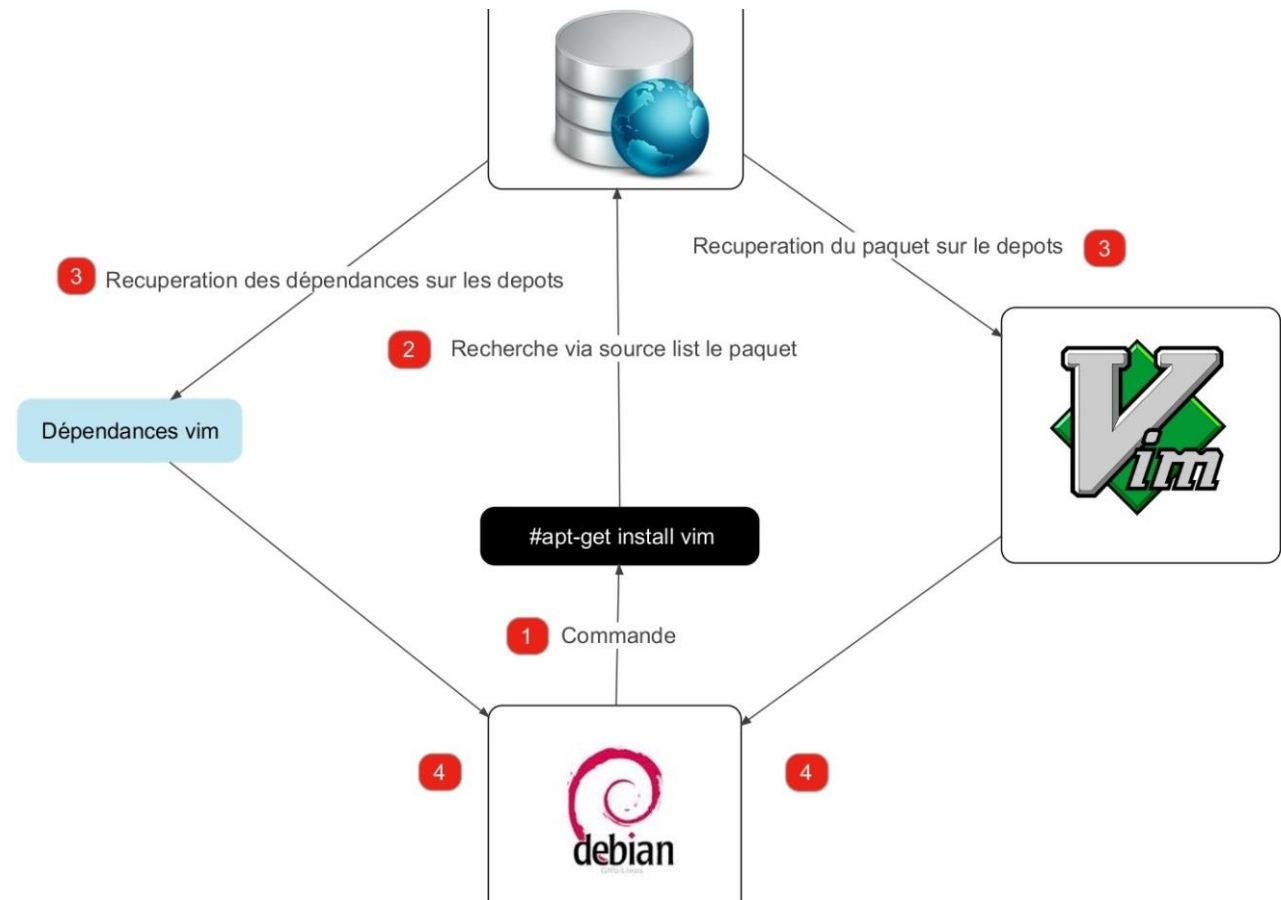
49

Administration du système

Linux

- La gestion des paquets :
 - Il s'agit d'un des fondements du système,
 - Procédé quasiment identique sur toutes les distributions, seules les commandes et le format des paquets changent.
- Le principe de base est de télécharger puis installer n'importe quel logiciel à partir d'une base de données qui indexe les paquets et redirige vers des serveurs (dépôts) où sont stockés les binaires officiels associés à la distribution.

- **1** la commande envoie la requête en se référant au fichier /etc/apt/sources.list
- **2** apt-get vérifie si le paquet est disponible sur le serveur de BDD index (Source List)
- **3** Source List redirige vers les dépôts du logiciel et les dépendances nécessaires à l'installation
- **4** les binaires sont rapatriés puis installés en local



- Mettre à jour la liste des paquets :
`~# apt-get update`
- Mettre à jour les paquets installés :
`~# apt-get upgrade`
- Installation de paquets :
`~# apt-get install [pkg]`
- Suppression de paquets :
`~# apt-get remove [pkg]`
- Purge des paquets téléchargés :
`~# apt-get clean [pkg]`
- Recherche de paquets :
`~# apt-cache search [pkg]`
- Installer un .deb :
`~# dpkg -i [pkg.deb]`

`Apt-get update -y && apt-get dist-upgrade -y`

09/10/2018

53

Droits et utilisateurs

Linux

- La gestion des utilisateurs :

- Un fichier de stockage des utilisateurs : /etc/passwd
- Un fichier de stockage des mots de passe : /etc/shadow

¹ ² ³ ⁴ ⁵ ⁶ ⁷
root : x : 0 : 0 : root : /root : /bin/bash

- | | |
|---|---|
| 1 | Identifiant (login) de l'utilisateur |
| 2 | Le caractère X remplaçant le mot de passe |
| 3 | User ID entifier (UID) identifiant de l'utilisateur |
| 4 | Group Identifier (GID) identifiant du groupe |
| 5 | Commentaire |
| 6 | Répertoire /Home de l'utilisateur |
| 7 | Emplacement de l'interpréteur Shell par défaut pour l'utilisateur |

```
plopy@plopy-virtual-machine ~ 15:28:07 :$cat /etc/passwd | grep plopy  
plopy:x:1000:1000:c'est copain plopy:/home/plopy:/bin/bash
```

- Création d'un utilisateur :
`~# useradd`
- Spécifier un mot de passe :
`~# passwd`
- Supprimer un utilisateur :
`~# userdel`
- Verrouiller un utilisateur :
`~# passwd -l`
- déverrouiller un utilisateur :
`~# passwd -u`
- Ajouter un commentaire :
`~# usermod -c`

- Un fichier de stockage des groupes: `/etc/group`

1	Nom du group
2	Mot de passe, non utilisé par défaut
3	Identifiant du groupe (GID)
4	Liste des utilisateurs dans le groupes. Séparés par une virgules

```
plopy@plopy-virtual-machine 15:35:51 :$cat /etc/group | grep plopy
adm:x:4:syslog,plopy
cdrom:x:24:plopy
sudo:x:27:plopy
dip:x:30:plopy
plugdev:x:46:plopy
lpadmin:x:116:plopy
plopy:x:1000:
smbashare:x:126:plopy
```


- Créer un groupe :
- Supprimer un groupe :
- Modifier un groupe :
- Modifier le groupe principal d'un utilisateur :
- Ajouter un utilisateur dans des groupes :
- Retirer un utilisateur d'un groupe :

```
~# groupadd nom_groupe  
~# groupdel nom_groupe  
~# groupmod [options] nom_groupe  
~# usermod -g nom_group nom_user  
~# usermod -aG groupe nom_user  
~# deluser nom_utilisateur nom_group
```

09/10/2018

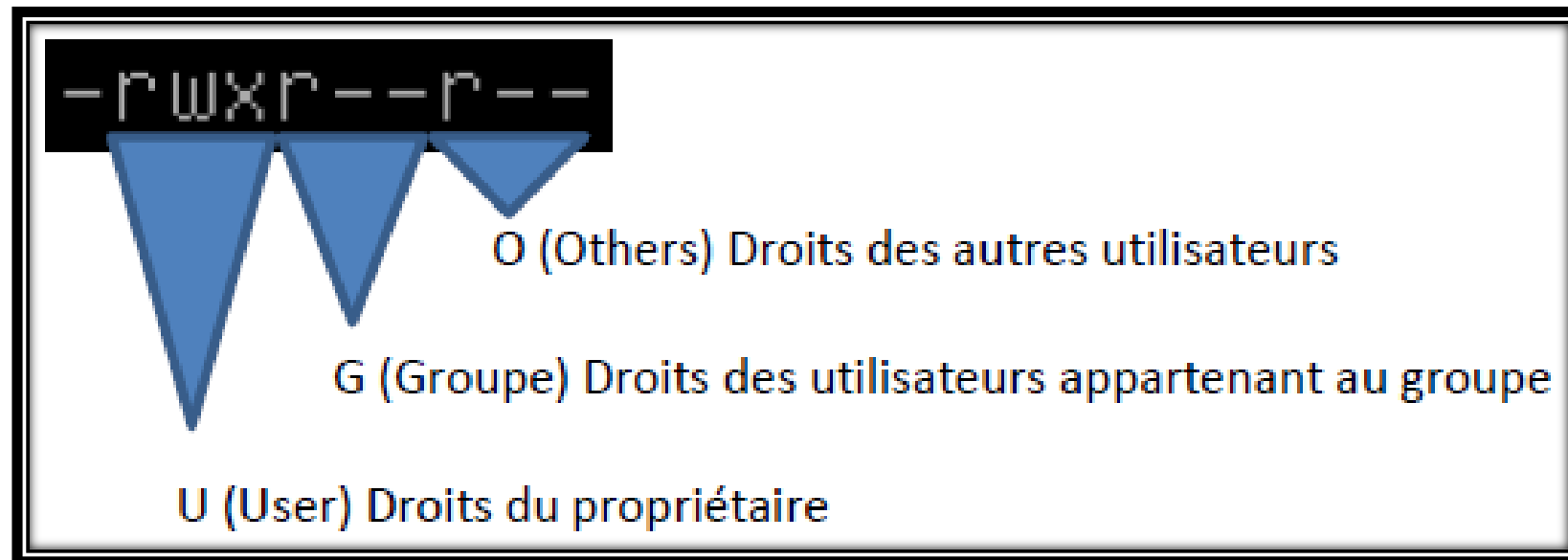
58

Droits et utilisateurs

Linux

- Il existe 3 différents droits possibles :
 - Droits en lecture (r) :
 - Autorisation de lire un fichier ou de consulter le contenu d'un répertoire.
 - Droits en écriture (w) :
 - Autorisation de modifier un fichier ou de créer/supprimer des fichiers dans un répertoire.
 - Droits en exécution (x) :
 - Autorisation d'exécuter un fichier (script).

- Seul root et le propriétaire du fichier peuvent changer les permissions d'accès sur ce même fichier.
- Affichage des permissions : `ls -l`
- La représentation est la suivante :



- Exemple :
~# ls -l plopy
-rwxr-xr-- 1 user group 12345 Nov 15 09:19 plopy
- Ce qui se traduit par :
 - L'utilisateur user a les droits en lecture, écriture et exécution sur le fichier plopy,
 - Le groupe group a les droits en lecture et exécution sur le fichier plopy,
 - Les autres utilisateurs n'ont qu'un accès en lecture sur ce fichier.

- Changer les droits d'un fichier/répertoire : ~# chmod
- changer le propriétaire (user : group) d'un fichier ou dossier :
~# chown
- Il existe trois notations possibles de changement de droits :
 - ~# chmod u+x *nom_fichier* (*u:user,g:group,o:other*)
 - ~# chmod 755 *nom_fichier* (*notation octale*)
 - ~# chmod u=rw,g=r,o= *nom_fichier* (*user = RW-*)

- La notation octale sur chmod :

Binaire	-----	Logique	-----	Décimal
000	-----	(---)	-----	0
001	-----	(--x)	-----	1
010	-----	(-w-)	-----	2
011	-----	(-wx)	-----	3
100	-----	(r--)	-----	4
101	-----	(r-x)	-----	5
110	-----	(rw-)	-----	6
111	-----	(rwx)	-----	7

r (4) : autorisation de lecture
w (2) : autorisation d'écriture
x (1) : autorisation d'exécution.

- Il suffit de coder chaque droit désiré avec un « 1 » et de convertir le bloc binaire de 3 digits en décimal.

- Pour changer le propriétaire d'un fichier :
~# chown new_user fichier
- Pour changer le groupe d'un fichier :
~# chown :new_group fichier
- Pour changer le propriétaire et le groupe :
~# chown new_user:new_group fichier
- L'option -R permet une mise à jour récursive.

- Pour changer le propriétaire d'un fichier :
~# chown new_user fichier
- Pour changer le groupe d'un fichier :
~# chown :new_group fichier
- Pour changer le propriétaire et le groupe :
~# chown new_user:new_group fichier
- L'option -R permet une mise à jour récursive.

09/10/2018

66

TP ! =)

Linux

- Gestion droit et utilisateurs
- Surveillance du système avec top
- Installation LAMP
- Planifier une tâches (apparitions d'une vache toutes les 2min) = apt-get install moo
- Archiver le /home de votre utilisateur

- Afficher les ressources utilisées : ~# top
- Exemple de sortie :

```
top - 22:38:53 up 42 days, 5:13, 0 users, load average: 0.38, 0.58, 0.66
Tasks: 15 total, 2 running, 13 sleeping, 0 stopped, 0 zombie
Cpu(s): 9.0%us, 1.3%sy, 0.0%ni, 89.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1554840k total, 1372092k used, 182748k free, 292528k buffers
Swap: 979924k total, 143012k used, 836912k free, 440040k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        15   0   1848   208   172  S   0.0   0.0   0:49.76  init
  2590 root        18   0   1624   592   476  S   0.0   0.0   0:02.64  syslogd
  2953 root        18   0   2196   664   600  S   0.0   0.0   0:00.80  cron
  4415 root        18   0 11652 4272 2128  S   0.0   0.3   0:01.12  apache2
15083 nagios      15   0   3396   728   688  S   0.0   0.0   0:07.34  nrpe
17509 root        15   0   4928   860   764  S   0.0   0.1   0:00.02  sshd
22321 www-data   15   0 37236 30m 4036  S   0.0   2.0   0:18.70  apache2
23085 www-data   15   0 36724 29m 4040  S   0.0   1.9   0:10.15  apache2
23329 www-data   15   0 37120 29m 4024  S   0.0   2.0   0:07.89  apache2
23825 ikipou     15   0   4024 1716 1300  S   0.0   0.1   0:00.01  bash
23855 www-data   15   0 31412 24m 4032  S   0.0   1.6   0:02.94  apache2
23858 www-data   15   0 31324 23m 4016  S   0.0   1.6   0:02.61  apache2
23917 www-data   15   0 11788 3480 1296  S   0.0   0.2   0:00.00  apache2
23918 www-data   15   0 11788 3480 1296  S   0.0   0.2   0:00.00  apache2
23939 ikipou     15   0   2228 1076   860  R   0.0   0.1   0:00.00  top
```

- Le planificateur de tâches :
 - Utilitaire : cron.
 - Il s'agit d'un service qui tourne en tâche de fond (daemon).
 - Pour afficher la liste des tâches planifiées :
~# crontab -l
 - Pour modifier la liste des tâches :
 - ~# vim /etc/crontab

- Exemple de planification :

```
# m h dom mon dow  command
0,30 9-17 * * 1-5 /var/scripts/script1.sh
```

- M : minute
- h : heure (hour)
- Dom : jour du mois (day of month)
- Mon : mois (month)
- dow : jour de la semaine (day of week)

Dans notre exemple, le script « script1.sh » sera exécuté toutes les 30 minutes de 9h à 17h tous les jours de la semaine, du lundi au vendredi.

Commande pour exécuter backup.sh deux fois par jour (à 2h30 et 14h30) du lundi au vendredi
30 2,14 * * 1-5 root /usr/local/sbin/backup.sh

30	2,14	*	*	1-5	root	/usr/local/sbin/backup.sh
0-59	0-23	1-31	1-12	0-7	Commande à exécuter	
					Utilisateur qui exécute la commande	
					Jour de la semaine	
					Dimanche = 0 ou 7; lundi = 1; mardi = 2; mercredi = 3; jeudi = 4; vendredi = 5; samedi = 6	
					Mois	
					Janvier = 1 (...) Décembre = 12	
					Jour	
					Heure	
					Minute	

<http://jhroy.ca>

<http://jhroy.ca>

- Archivage :
 - L'archivage consiste à regrouper un certain nombre de fichiers dans un seul et même fichier.
 - Utilitaire conseillé : tar
 - Pour créer une archive :
 - `~# tar -cvf [nom_archive] [fichiers]`
 - Pour extraire les fichiers d'une archive :
 - `~# tar -xvf [nom_archive]`
 - Pour consulter le contenu d'une archive :
 - `~# tar -tf [file]`

- Exemple de création d'une archive compressée avec chemins complets des fichiers :
~#tar -cPzf archive.tar.gz /home
- Extraction des fichiers à l'endroit initial :
~#tar -xPzf archive.tar.gz