R4.08 : Introduction à la virtualisation

1. Rappels et concepts

Franck Butelle

IUT Villetaneuse,
Département Informatique, S4

01/04/2024

R4.A.08 : Introduction à la Virtualisation

Plan du cours

* Objectif du cours (PN) :
  comprendre les principes et enjeux de la virtualisation en informatique.
* Prérequis :
  Notions d'Administration Système et de réseaux.

- Un seul cours d'amphi
- (presque) Que des TPS ! Plan initial (à adapter suivant avancement) :
  - Rappels Admin Syst. : Systemes de (Gestion de) Fichier et utilisateurs
  - NFS / Overlayfs
  - VPN
  - VirtualBox, chroot, Qemu, Docker, (à adapter suivant avancement)
- Evaluation : ctrl final sur papier

Plan cours

Introduction et définitions

- *Virtualisation* : *ens. des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine/un seul réseau, plusieurs systèmes d'exploitation/réseaux et/ou plusieurs applications, séparément les uns de autres comme s'ils fonctionnaient sur des machines/réseaux physiques distincts.*
- Les bases de la virtualisation ont été données par Popek et Goldberg en 1974.
- Des usages divers !
  - Mémoire virtuelle (non étudiée dans ce cours)
  - Virtualisation du stockage
  - Virtualisation des réseaux (VLAN, VPN, ...)
  - Virtualisation d'applications par des «isolateurs» (chroot, docker,...)
  - Virtualisation de systèmes complets : Hyperviseur de type 1 ou 2.

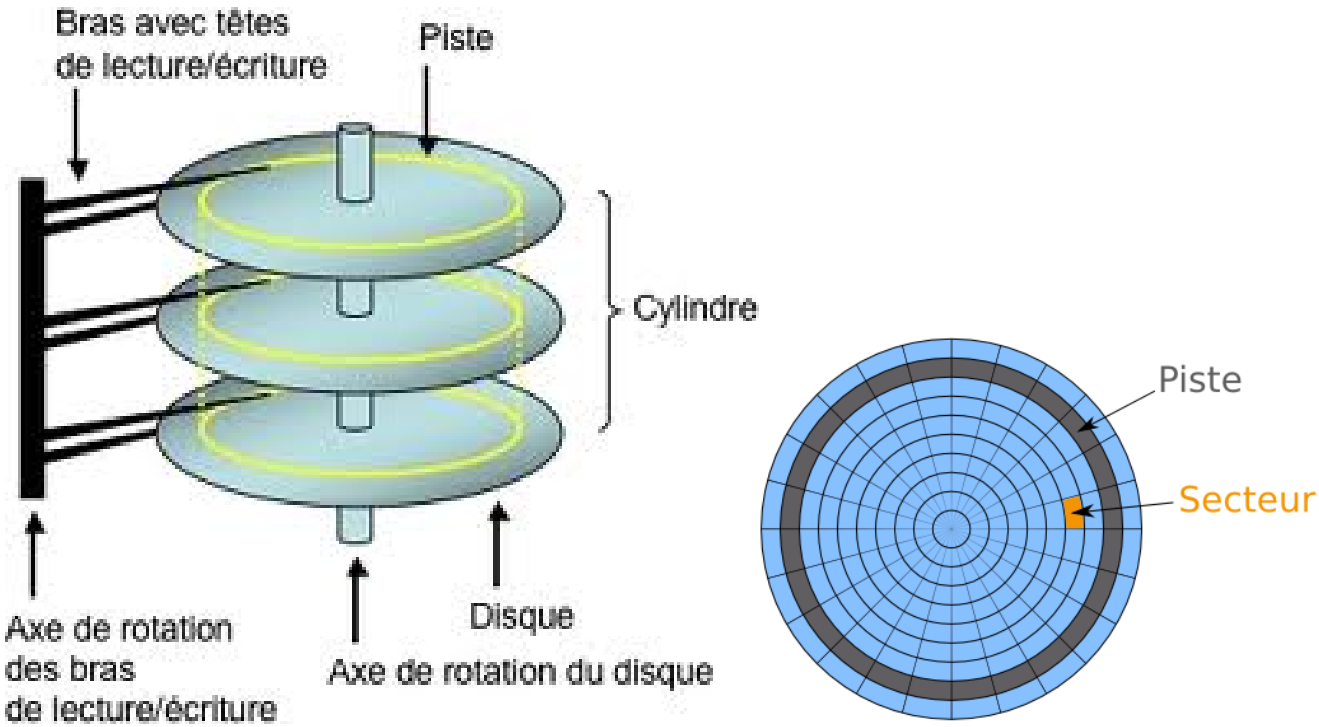
Plan de la section

Définitions

- Un *disque* est un support non volatile de données.
- L'unité d'allocation du disque s'appelle un *secteur* (souvent 512 o).
- Le disque est organisé en secteurs, têtes et cylindres (mais obsolète)
- Un disque est découpé en une ou plusieurs *partitions* (physiques)
- Un *volume* (logique) est constitué d'une ou plusieurs partitions.
- Un volume comporte un *Système de Fichiers* (SF ou FS en Anglais)
- Le «montage» (la «greffe») : intégrer le répertoire d'un volume à un point désiré du répertoire général, appelé le point de montage.

Rem. : La différence entre une partition et un volume, est que l'un est physique, et l'autre est logique. L'utilisateur ne voit que le volume, avec une racine, des répertoires, et des fichiers.

Disque dur



Rem. : d'où Adressage CHS *(Cylindre Head Sector) : historique mais trop limité.*
Le numéro de cylindre sur 10 bits, num. tête sur 8 bits, numéro de secteur sur 6 bits donne 8 Gio max !
Astuce : on code 254 ou 255 têtes pour dire qu'il faut utiliser l'adressage linéaire (LBA : Linear Block Adressing)

Découpage du disque en partitions

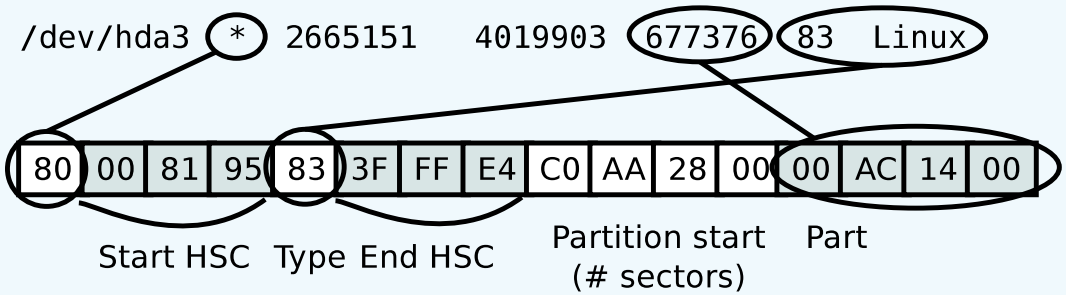
- On découpe le disque en plusieurs parties pour faciliter la gestion, autoriser plusieurs *Systèmes d'Exploitation (SE)*, etc.
- Il existe plusieurs formats de table de partitions
  - MBR (DOS) : PC un peu anciens (Windows, Linux), clé USB
    - Limite : 2,2 To par partition...
  - GPT : PC modernes (Mac OS X, Windows, Linux)
    - GUID Partition Table (GUID : Globally Unique Identifier)
    - min. 128 octets par descripteur de partition
    - Taille disque max :2^{64} secteurs. Si secteurs de 512 o, alors $\approx 9,4$Zo $= 9.4 \times 10^{21}$, soit 9.4 milliards de Téra octets.
  - Apple Partition Map : Vieux PowerPC Mac
  - BSD Disklabels (OpenBSD, FreeBSD)
  - ...

MBR : Master Boot Record

Au tout début d'un disque dur : le premier secteur

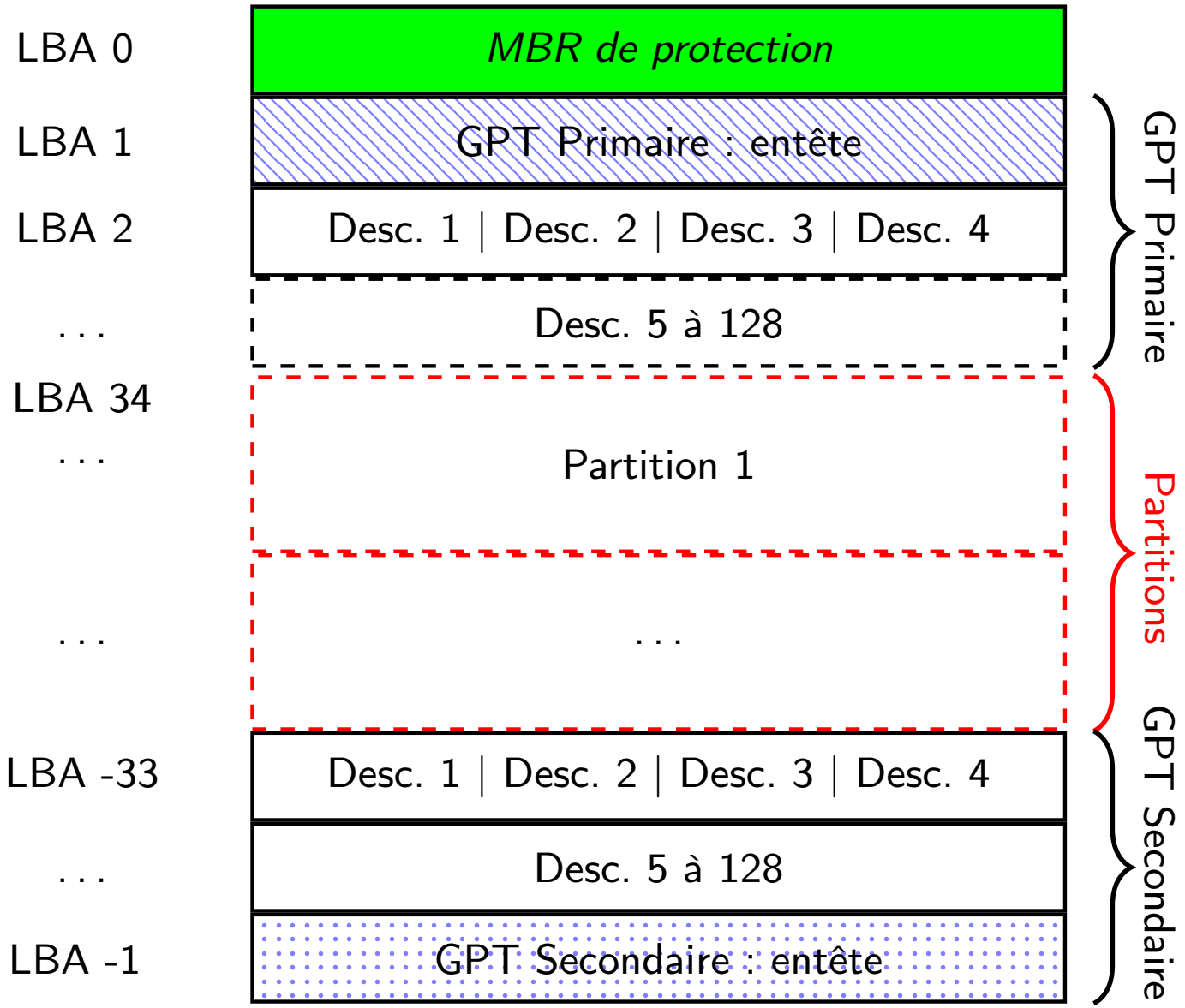
Début	Fin	Contenu
0	445	Programme vérifiant la table des partitions et lançant le secteur d'amorce de la partition active
446	509	Table des partitions
510	511	Signature "magic number" (=AA55h)

Détail partition :



Donc seulement 64 octets pour la table des partitions, 16 oct. par partition : donc 4 partitions (primaires) !
Astuce : partitions étendues dans les partitions primaires qui contiennent des partitions logiques. . .

GPT : GUID Partition Table (ici secteur de 512o, Desc. de 128o)



Entête GPT (pour les curieux)

Off.	Taille en oct.	Commentaires
0	8	Signature EFI : 45 46 49 20 50 41 52 54
8	4	Version (1.0 codée 00 00 01 00)
12	4	Taille de l'entête en octets souvent 92
16	4	CRC 32 bits de l'entête
20	4	Réservé, doit être à zero
24	8	Adresse de cette entête GPT (doit être 1 pour la primaire)
32	8	Adresse entête de l'autre GPT
40	8	Première adresse utilisable pour les partitions
48	8	Dernière adresse utilisable pour les partitions
56	16	GUID du disque (aussi appelé UUID)
72	8	Adresse du tableau des partitions (2 pour le GPT primaire)
80	4	Nombre de descripteur de partitions
84	4	Taille en octets d'un descripteur de partition (gén. 128)
88	4	CRC 32 bits de la table des partitions
92	*	réservé, la fin du bloc doit être à zéro (420 octets sur un bloc de 512 octets)

Nommage des disques et partitions sous Unix/Linux

- /dev/hda : le *disque entier* (IDE) (a, b, c,...).

Exemples

Partitions primaires : /dev/hda1 , /dev/hdf4 , ...
Partitions logiques (num>4) : /dev/hdc5, /dev/hdd7

- /dev/sda : disque SCSI et SATA, clés et disque USB etc. (idem).

Exemple de partition : /dev/sdb1

- /dev/nvme0n1 : nouveaux disques SSD PCIe NVMe [1] (0,1,2,...)

Exemple de partition : /dev/nvme1n1p2

1. SSD :Solid-State Drive, NVME :Non-Volatile Memory Express

Le Système de Fichiers (SF) composant du SE

Le SF est la partie du Système d'Exploitation (*Operating System*) qui :

- Réalise la correspondance entre organisation logique (la vue utilisateur) et l'organisation physique.
- Fournit une interface à l'utilisateur.
- Optimise en espace et en temps l'utilisation des disques.
- Assure le stockage permanent des données.
- Assure l'intégrité (blocs endommagés, corruption, etc).
- Assure le partage et la protection des données (attributs, droits).

Rem. : le SF n'est pas que «ouvrir/écrire» dans un fichier, c'est une partie fondamentale du Système d'Exploitation.
Plusieurs «façons de faire», donc plusieurs types de SF...

Le Système (de Gestion) de Fichiers : vue conceptuelle

Le SF est la *méthode d'organisation des fichiers et répertoires* sur un volume.

- Découpe le volume en blocs (ou clusters), qui est l'unité d'adressage.
- Comporte des blocs *réservés au SF* pour la gestion : c'est le *formatage*
- Tous les autres blocs sont des blocs de données.
- Algorithmes d'organisation des données

Rem. : Exemple d'adressage : le numéro de maison dans une rue !
Si la taille des blocs est fixe, elle est décidée au formatage (liée à la taille de la partition : souvent grosse partition $\Longrightarrow$ gros blocs)

Algorithmes et Fragmentation

- Allocation contigüe.
- Allocation par liste chaînée.
- Allocation par liste chaînée avec table d'allocation (voir FAT : *File Allocation Table*)
- Allocation par *nœud d'index (index-node ou* Inode*)*

Deux types possibles de fragmentation :

- L'utilisation de bloc de taille fixe introduit de la *fragmentation interne* (ex : stocker «ok» dans un fichier : min. 4096 octets !)
- L'allocation de bloc de taille dyn. introduit de la *fragmentation externe*

Rem. : Les SF modernes réduisent/retardent la fragmentation mais ne la supprime pas tout à fait.

Rappel : Grands Préfixes

Décimal			Binaire		
Nom	Symb.	Valeur	Nom	Symb.	Valeur
kilo	k	10^3	kibi	Ki	$2^{10} = 1\,024$
méga	M	10^6	mébi	Mi	$2^{20} = 1\,048\,576$
giga	G	10^9	gibi	Gi	$2^{30} = 1\,073\,741\,824$
téra	T	10^{12}	tébi	Ti	$2^{40} \approx 1{,}100 \times 10^{12}$
péta	P	10^{15}	pébi	Pi	$2^{50} \approx 1{,}126 \times 10^{15}$
exa	E	10^{18}	exbi	Ei	$2^{60} \approx 1{,}153 \times 10^{18}$
zetta	Z	10^{21}	zébi	Zi	$2^{70} \approx 1{,}181 \times 10^{21}$
yotta	Y	10^{24}	yobi	Yi	$2^{80} \approx 1{,}209 \times 10^{24}$

Quelques SF

Nom	Origine	Compatibilité	Max Vol.	Max fichier	Usage
VFAT	Microsoft	ok pour Linux	2 Tio	4 Gio	Petite clé USB
exFAT	Microsoft proprio →08/2019	Linux kernel 5.4+	64 Zio	16 Eio	SDcard
NTFS	Microsoft proprio windows NT→	ok pour Linux sauf compression table	256 Tio	16 Tio	
iso9660	Tous	ok	2 Gio	8 Tio	CDROM
HPFS+	Apple Mac	partiel+	8Eio	8 Eio	
ext2	Linux	possible	32 Tio	2 Tio	vieux !
ext3	Linux	possible	32 Tio	2 Tio	
ext4	Linux	possible	1 Eio	16 Tio	
btrfs	Oracle Linux	possible	16 Eio	16 Eio	
XFS	SGI Redhat Linux	possible	8 Eio	8 Eio	

Notion d'Inode (nœud d'index) (ext2,3,4)

Un fichier, en interne, est représenté par un *Inode/Inœud* (métadonnée), contenant :

- droits d'accès, type du fichier (-, d, l, c, p, b, s)
- numéro du propriétaire (UID), numéro de groupe proprio. (GID).
- taille du fichier.
- date et heure dernier accès en lecture (atime *Access*)
- date et heure dernier accès en écriture (mtime *Modify*)
- date et heure dernière modification de l'Inode (ctime *Change*)
- nombre de liens physiques sur cet Inode
- des pointeurs vers les blocs de données.
- ne contient pas le nom du fichier.

Les numéros d'Inodes sont spécifiques à un volume.

Voir `ls -i`, `stat` sous Linux.

stat et l'occupation disque

`stat` affiche des infos concernant l'Inode du fichier spécifié :

stat /bin/bash

```
  File: ‘/bin/bash’
Size: 646672          Blocks: 1272       IO Block: 4096   regular file
Device: 804h/2052d      Inode: 2031622     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2021-12-09 22:43:37.000000000 +0100
Modify: 2021-08-08 22:52:15.000000000 +0200
Change: 2021-12-09 21:40:02.000000000 +0100
```

- «Size» : la taille du fichier en octets
- «Blocks» : nbre de blocs de 512 octets
- «IO Block» : taille de bloc min. pour le SF,

Donc 1 272×512 = 651 264 o. sur le disque (au lieu de 646 672 o.)
1 fichier d'1 octet ⟶ 4 096 o. sur le disque (8 Blocks).
1 répertoire vide ⟶ 4 096 o. sur le disque.

Les fichiers sous UNIX

Tout est fichier dans UNIX

Les types de fichiers sous UNIX (vus par `ls -l`) :

- les fichiers ordinaires : notés **-**
- les répertoires : notés **d**
- les fichiers spéciaux :
  - périphériques en mode **b**lock ,**c**har
  - tubes nommés : **|** ou **p** (*linux*), **l**iens symb., **s**ockets Unix

Exemples de fichiers sous Linux

Exemple

```
butelle@ici> ls -l /tmp/testdir
total 0
-rw-r--r-- 1 butelle butelle    0 mars  29 16:24 fichierNormal
lrwxrwxrwx 1 butelle butelle   13 mars  29 16:24 lienS -> fichierNormal
brw-r--r-- 1 root    root    1, 3 mars  29 16:27 periphBlock
crw-r--r-- 1 root    root    4, 0 mars  29 16:27 periphChar
drwxr-xr-x 2 butelle butelle   40 avril 20  2015 repertoire
prw-r--r-- 1 butelle butelle    0 mars  29 16:28 tubeNommé
srwxr-xr-x 1 butelle butelle    0 mars  29 16:02 unixSocket
```

Répertoires usuels du système UNIX

- / racine du système
- /**dev** fichiers spéciaux liés aux périph.
- /**etc** fichiers de configuration
- /**var** fichiers dont le contenu varie
  - /**var**/**log** traces d'exécutions
  - /**var**/**spool** fichiers en file d'attente mail, impression,...
- /**usr** fichiers système en lecture seule
  - binaires utilisateur /**usr**/**bin** et /**bin** et /**usr**/**local**/**bin**
  - binaires super-utilisateur /**usr**/**sbin** et /**sbin**
  - librairies /**usr**/**lib** et /**lib**
- /**tmp** et /**var**/**tmp** : fichiers et répertoires temporaires
- /**boot** : fichiers de démarrage du système
- /**proc** : représentations des processus en cours d'exécution

Répertoires recommandés :

- /**mnt**, /**media** : montage de périphériques externes
- /**home** : répertoires de connexion des utilisateurs.

Liens

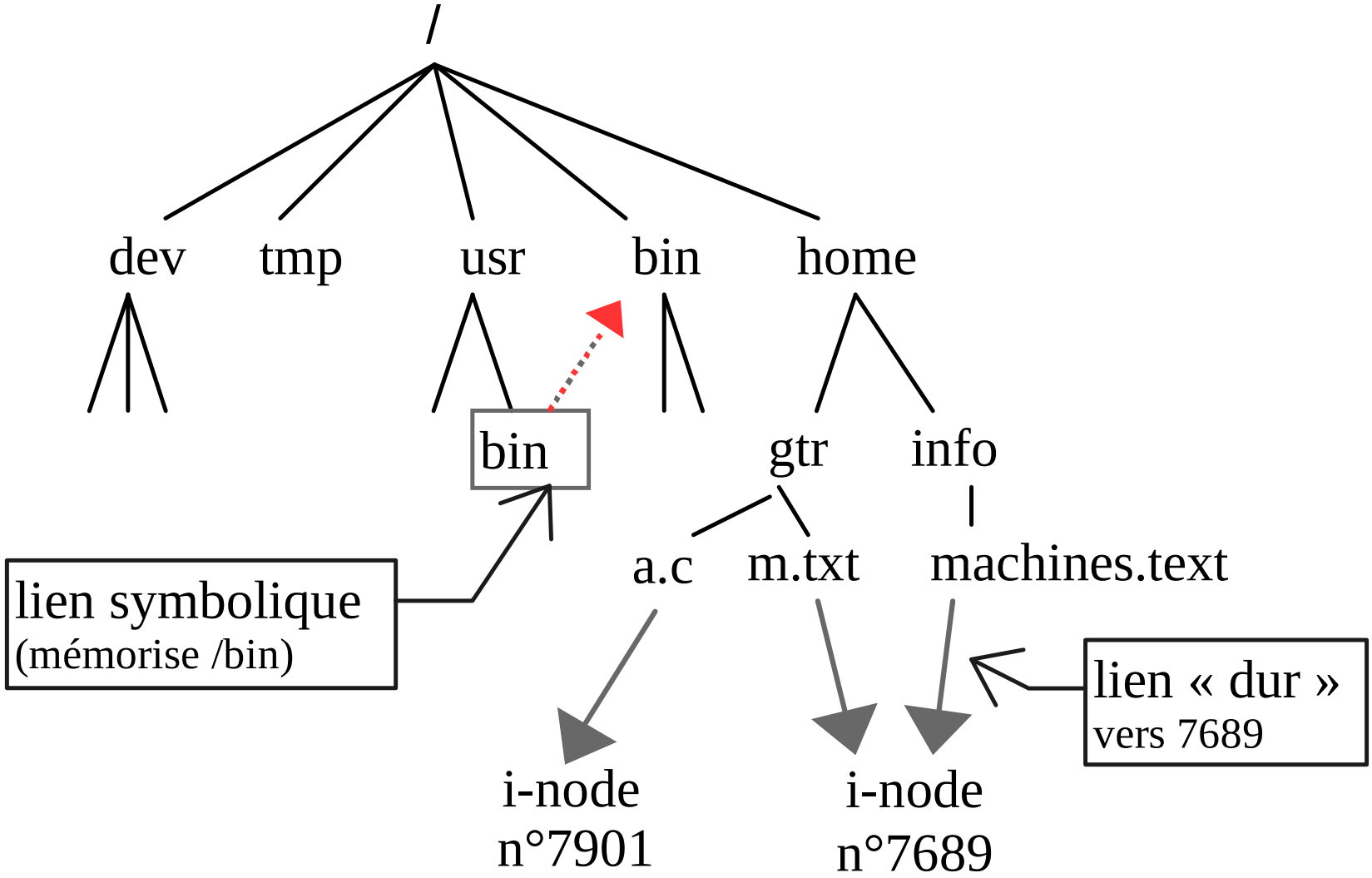
Les liens durs, ou physiques `ln`

- Un lien dur est une association nom de fichier => Inode.
- Une entrée de répertoire est donc un lien dur !
- Un Inode peut être référencé plusieurs fois !

Les liens symboliques `ln -s`

- Problème : les numéros d'Inode sont spécifiques à un volume.
- Les liens durs ne sont donc pas utilisables dans toute l'arborescence.
- Un lien symbolique est un *fichier spécial*, contenant le chemin du fichier référencé. *Les vrais droits sont ceux du fichier cible.*
- Il n'incrémente pas le compteur de référence d'un Inode.
- Si le fichier cible est effacé, le lien est *pendant (dangling)*.
- Similaires aux «raccourcis» de windows

Exemples de liens



La commande ls -la donne beaucoup d'informations...

```
$ ls -la
total 20
drwxr-xr-x  2 fb   users 4096 mars    6 11:19 ./
drwx------  4 fb   users 4096 mars    6 11:17 ../
drwxr-x--x  1 fb   users 4096 janv.   4  2021 Old.c
-rw-r--r--  2 fb   users    6 mars    6 11:19 fic2.txt
-rw-r--r--  2 fb   users    6 mars    6 11:19 lienDur
brw-r--r--  1 root root   1, 3 mars   29 16:27 sda1
crw-r--r--  1 root root   4, 0 mars   29 16:27 tty1
lrwxrwxrwx  1 fb   users    2 mars    6 11:19 toto -> /f
prw-r--r--  1 fb   users    0 mars    6 14:10 untube
```

type : droits

nb refs : proprio

groupe propri

taille sauf périph.

date

```
$ ls -i fic2.txt lienDur
6914053 fic2.txt  6914053 lienDur
```

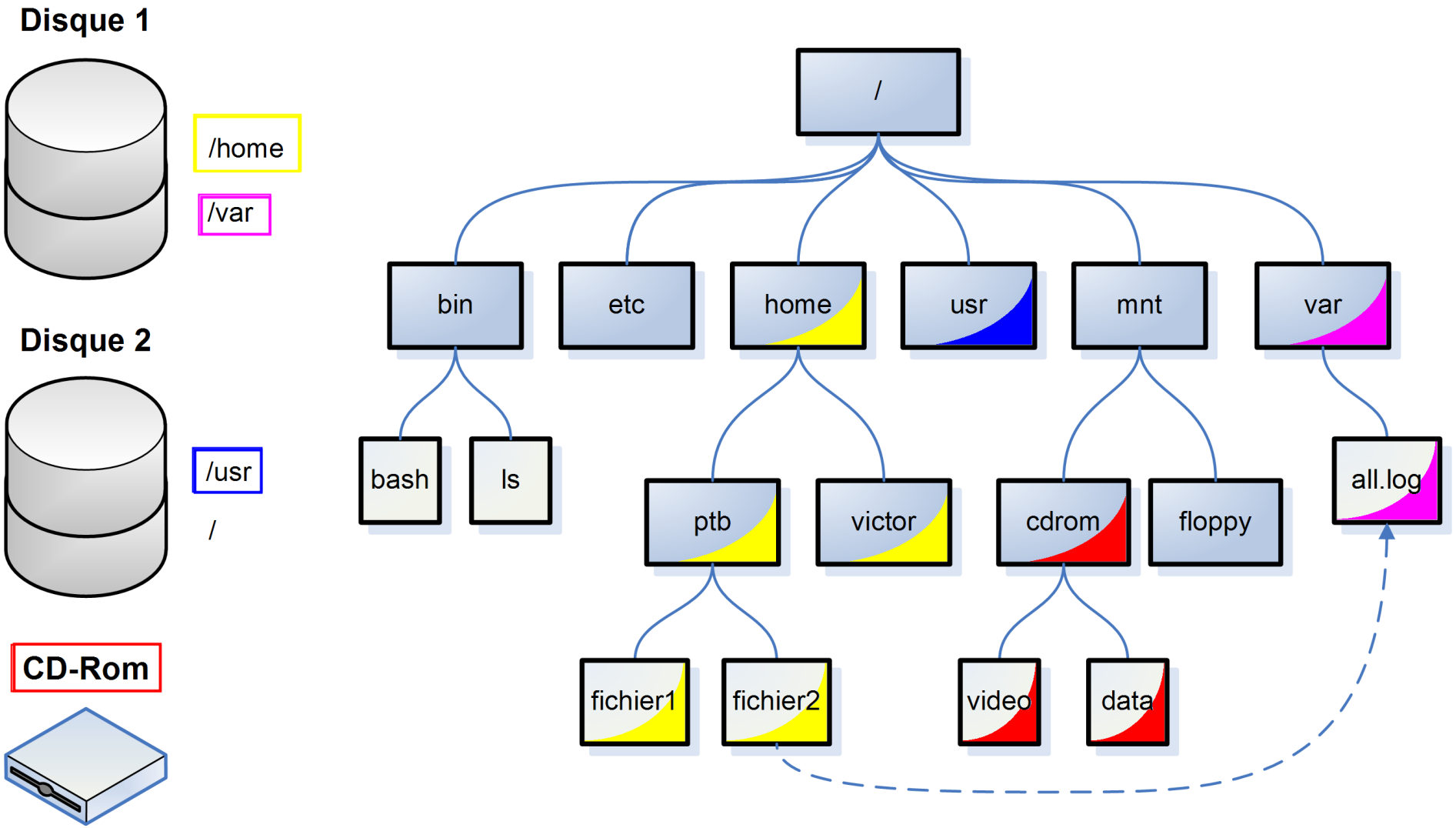
Le «total» est en nbre de Kio «montrés dans ce listing» (y compris ., .. et lienDur).

Que peut-on déduire de ces affichages ?

Déductions possibles de l'affichage précédent

- Le rép. courant (.) est à 4096 octets alors qu'il est petit (peu de fichiers) donc 4ko min pour les I/O Blocks,
- `Old.c` est en fait un répertoire (type d) !
- `lienDur` et `fic2.txt` ont le même inode d'où le compteur de réfs à 2 (lien "dur").
- `toto` est un lien symbolique (type `l`) vers `/f` de taille 2 car /f est codé sur 2 caractères.
  Notons que les *droits d'accès* sont les droits du fichier lié pas ceux du lien (qui sont toujours à rwxrwxrwx !).
- `untube` est un tube nommé (type `p` : *named pipe* voir `mkfifo`).

Exemple d'arborescence avec montage de SF



Outils Linux pour la gestion des SF

1. Partitionnement : `fdisk`, sfdisk (scripts), cfdisk, `gparted` (graphique)

Exemple : fdisk /dev/sda

2. Formatage : `mkfs`

Exemple : mkfs -t ext4 /dev/sda1

3. `mount` permet de monter/greffer un volume dans un point de montage, répertoire de l'arborescence globale

Exemple : mount -t vfat /dev/hda2 /home/moi/Mes_Documents

- `fsck` (*File System Check*) vérifie (et peut essayer de corriger) : consistance des blocs, consistance des fichiers, cohérence blocs alloués vs blocs libres, vérification des compteurs de réf. des Inodes.

/etc/fstab, df

`/etc/fstab` contient les paramètres permettant de monter les volumes automatiquement à *chaque démarrage* du système.

Exemple de contenu de /etc/fstab

```
# volume    mount-point  fs    options      dump/pass
/dev/sda1   /boot        ext2  noatime,ro   0 2
UUID=<ici identifiant hexa> swap swap defaults 0 0
/dev/sda4   /            ext3  noatime      0 1
```

`df` affiche la liste des volumes montés et l'espace libre/occupé sur chacun.

df -h

```
Filesystem  Size  Used Avail Use% Mounted on
/dev/sda4   107G   59G   43G  58% /
/dev/sda1    99M   25M   69M  27% /boot
```

UUID : Universally Unique IDentifier

Utilisateurs et groupes

Identifiés par un «login» (pseudonyme), accès protégé par un mot de passe.

Utilisateurs

- Identifié en interne par un numéro (*UID*, *user identifier*)
- Utilisateurs définis localement (`/etc/passwd` et `/etc/shadow`)
  ou par annuaire partagé en réseau (NIS, LDAP, ...).

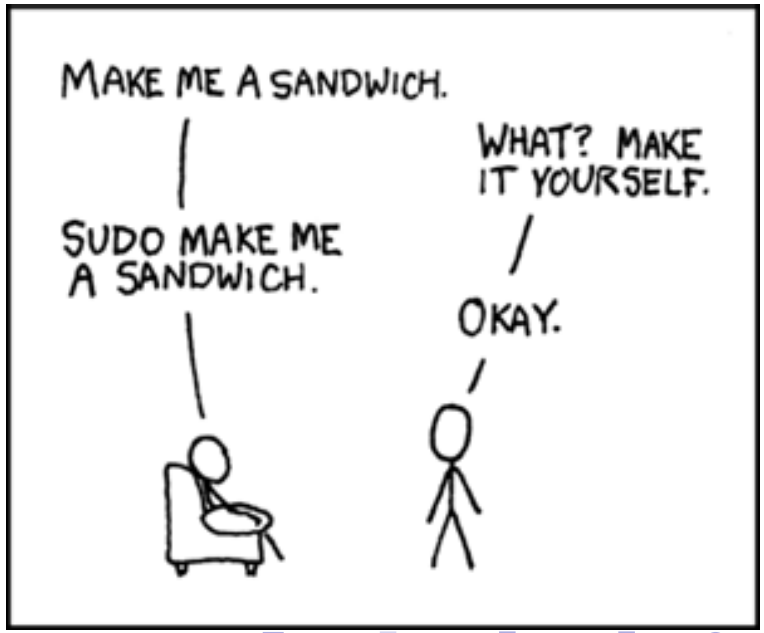
Groupes

- Utilisés pour définir des catégories d'utilisateurs
- Le GID du *groupe primaire* est défini dans /etc/passwd
- Les éventuels groupes secondaires dans `/etc/group` et /etc/gshadow
  (ou par annuaire réseau...)

Suite en TP...

sudo

- Objectif : donner le droit d'exécuter certaines commandes pour certains utilisateurs en tant que root ou autre utilisateur.
  - exemple : donner le droit de faire shutdown, graver des CDs,...
  - évite de donner le mot de passe de root à des non admins.
  - génère des traces de toutes les commandes lancées par sudo
  - mot de passe mémorisé 5 min par défaut
    - peut être changé par `timestamp_timeout=...`
- Configuration : fichier `/etc/sudoers`
- Utilisation : `sudo` *commande*
- ou encore `sudo -s` pour rester root... *sortie par exit ou CTRL D.*



https ://xkcd.com/149

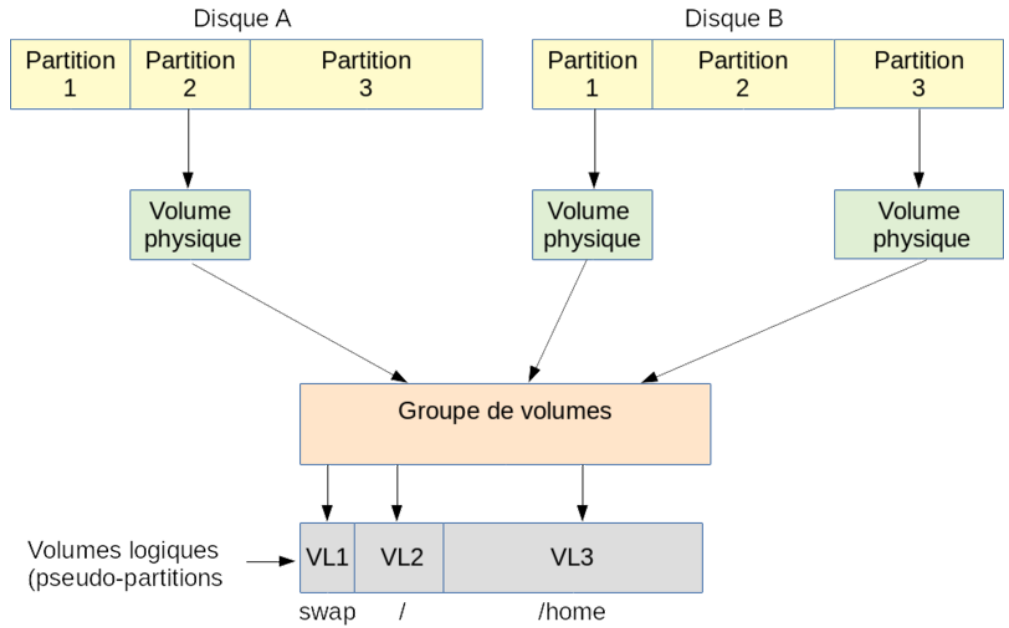
Plan de la section

Virtualisation du stockage

Idée : séparer la représentation logique et la représentation physique de l'espace de stockage.

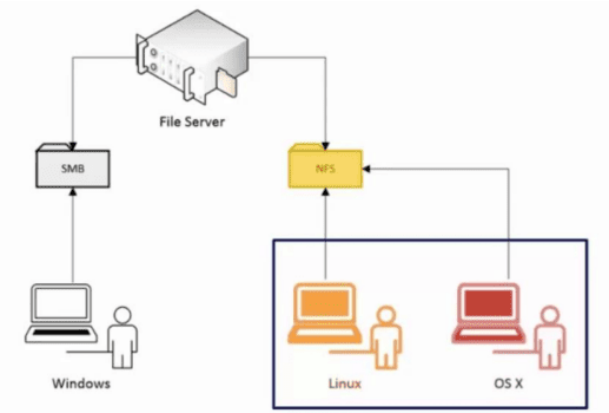
- Bas niveau (permet redondance, rapidité) :
  Redundant Array of Independent (ou Inexpensive) Disks (RAID)
- Système de fichiers virtuel : notion de volume(s) logique(s) regroupant plusieurs partitions (LVM)

https://doc.fedora-fr.org/wiki/Concepts_de_base_-_LVM



- Système de fichiers réseau (ex : NFS, SMB).
- Serveur NAS (Network Attached Storage) : fournit plusieurs services

NFS et SMB



- NFS (Sun Microsystem → Unix/linux) et SMB (IBM, Microsoft) sont des protocoles client/serveur
- Permettent l'accès à des fichiers centralisés distants
- Clients Linux utilisent la commande mount : `mount -t nfs` ; `mount -t smbfs`
- Désormais SMB est utilisable aussi par Linux (Samba) et Mac OS X (NFS théoriquement utilisable aussi sous windows)
- Mais le protocole NFS est incompatible avec SMB .

NFS : Avantages et inconvénients

- Avantages
  - L'utilisateur a l'impression d'avoir tous ses fichiers en local
  - Les modifications à un fichier sont faites par bloc (cache)
  - Dans la même connexion, l'accès simultané à plusieurs fichiers est possible
  - Accès simultané par plusieurs clients possible (lecture ok, écriture : lockd)
  - Résistance partielle aux pannes réseaux et pannes serveur (réessai auto)
  - Des clients et des serveurs open source sont disponibles
  - Bonnes performanances
- Inconvénients
  - échange des données en clair (jusque NFS v3)
  - fait confiance aux clients (pour les UID et les droits de montage) ! (jusque NFS v3)
  - La v4 corrige les pbs d'authentification et de confidentialité mais est plus difficile à configurer
  - dépend intensément du réseau $\Longrightarrow$ ralentissements possibles

Overlayfs

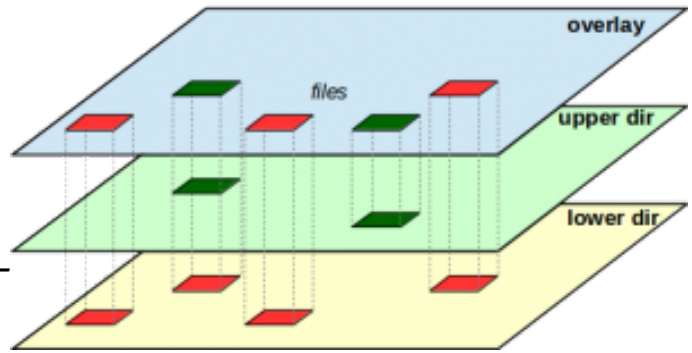
Overlay (ou overlayfs) permet de superposer des visions différentes du même FS.

- Techno fondamentale pour les systèmes «live» avec persistance, docker, etc.
- Idée de superposition de couches/niveaux de répertoires
- Un répertoire ou un fichier défini à un niveau est accessible au niveau le plus haut "upper"
- La vision par l'utilisateur correspond à une union des couches inférieures «overlay» = «lower» $\cup$ «upper»
- Seul le niveau «upper» permet des modifications/suppressions, le(s) niveau(x) inférieur(s) «lower» sont dit immuables (read only).

Sous linux :

```
mount -t overlay -o lowerdir=lower,\
 upperdir=upper,workdir=work overlay
```

Le répertoire work *doit être vide et sur le même système de fichiers que le répertoire upper.*



Plan de la section

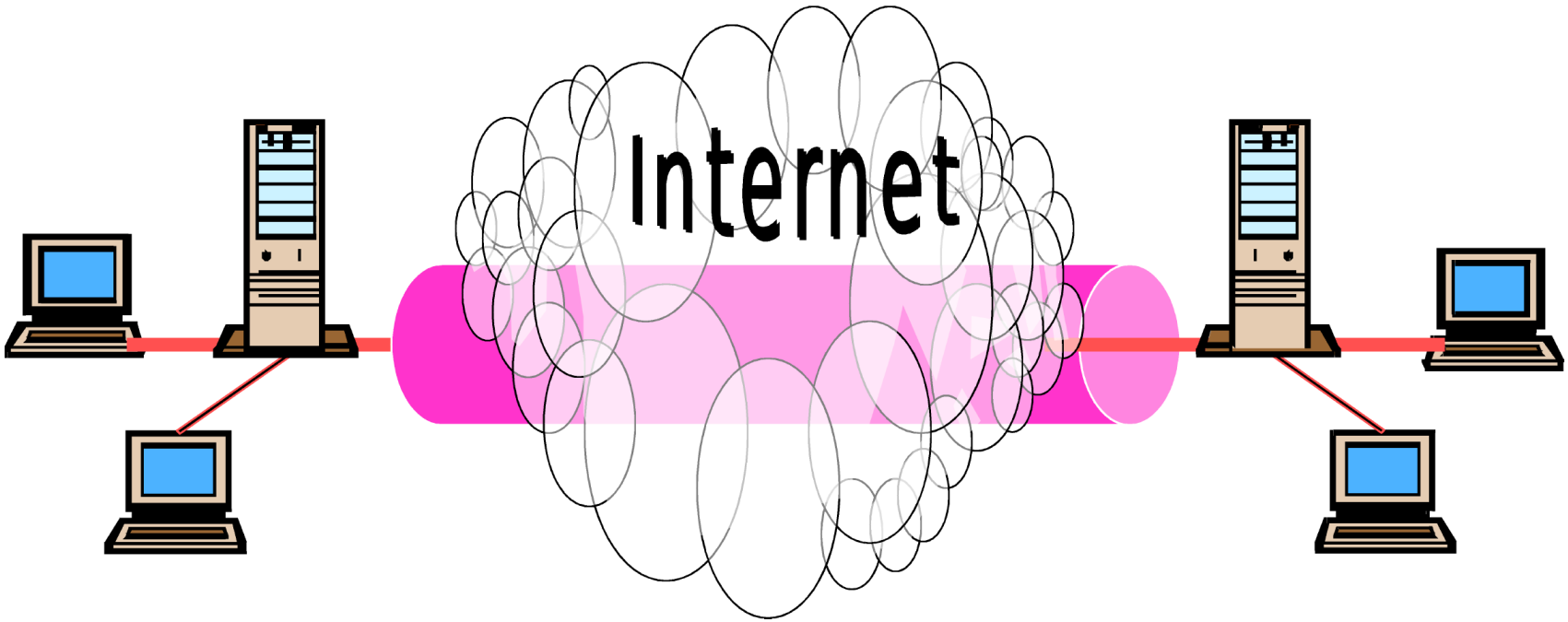
Virtualisation de réseaux

- VLAN : Virtual Local Area Network
  - Idée : cloisonner une partie d'un réseau local physique au niveau 2.
  - Nécessite des switchs
  - Plusieurs types de VLAN :
    - par port de switch
    - par adresse MAC
    - par protocole (dépend du proto de niv 3),
    - IEEE 802.1Q(2003) : ajoute des champs à l'entête de protocole de niveau 2 : «tags».
- VPN : Virtual Private Network
  - Essentiellement relier 2 ou plusieurs Réseaux Locaux physiques par un réseau publique (internet).
  - Le réseau publique n'étant pas sûr, il faut encapsuler les paquets dans une couche de crypto $\Longrightarrow$ notion de «tunnel»

Qu'est-ce qu'un «tunnel» ?

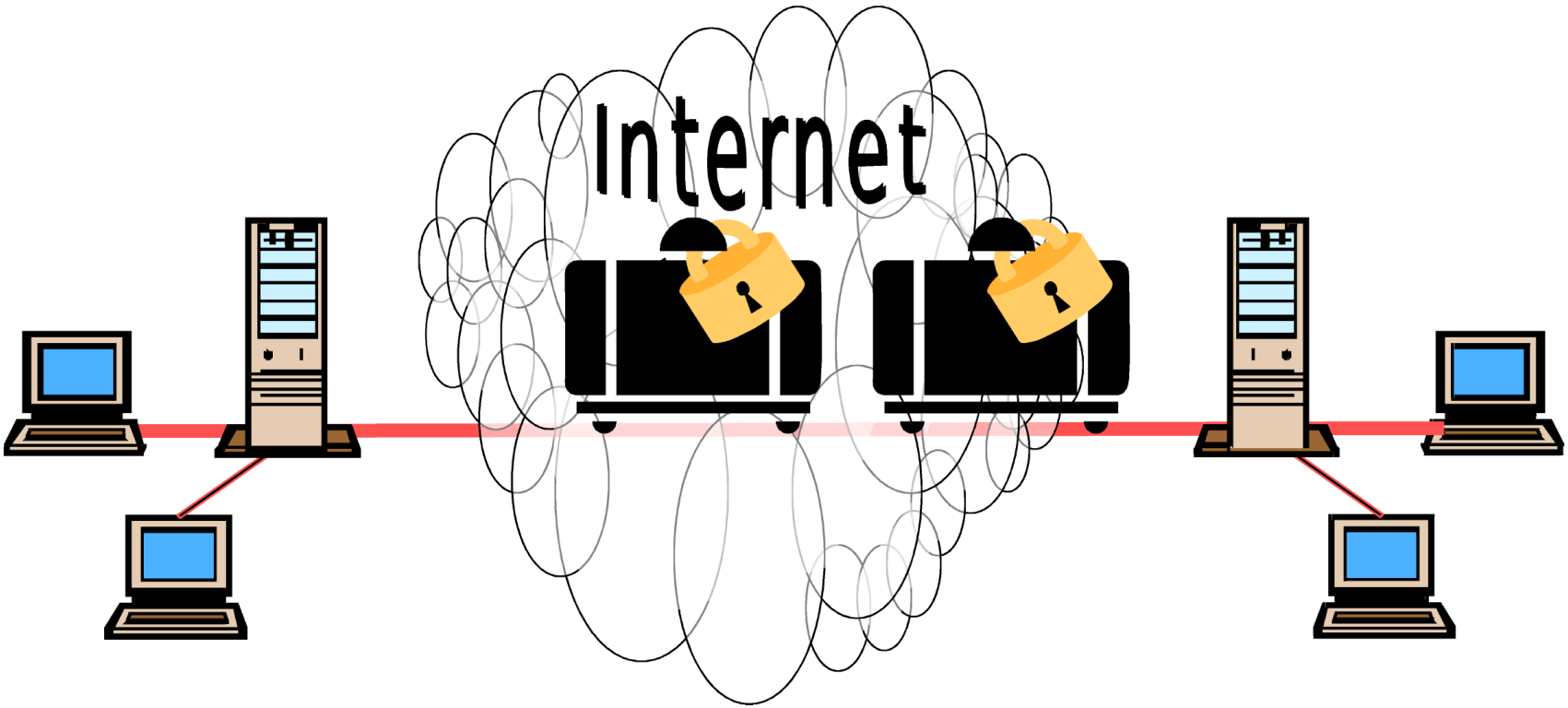
Un tunnel permet

- de cacher les adresses IP réelles (émetteur et récepteur)
- la confidentialité et l'intégrité des données (évent. auth.)
- établir une qualité de service si possible



Rem. : généralement le tunnel permet de «sécuriser» les paquets échangés entre deux routeurs d'extrémité.

Tunnel = paquets sécurisés



Suite en TP...

Plan de la section

Virtualisation d'applications / Isolation

Idée : encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné.
Motivation : sécurité et stabilité.
Exemples sous Linux :

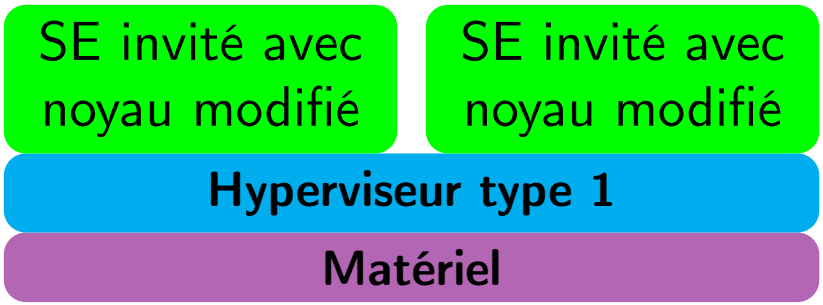
- wine (permet d'exécuter des applications Windows sur une plateforme Linux)
- chroot
- Conteneurs : docker, LXC

Virtualisation de système complet

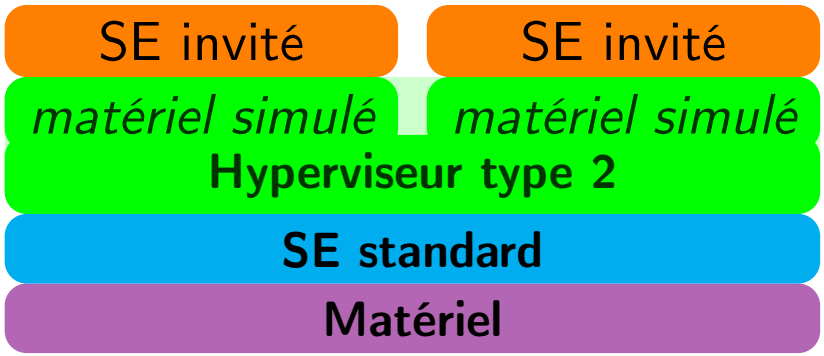
Idée : masquer les caractéristiques physiques pour pouvoir faire tourner un système d'exploitation complet.

Deux principales méthodes de virtualisation

- Paravirtualisation (Hyperviseur type 1) Xen, Hyper-V
  - Adapter le système pour communiquer avec un hyperviseur au lieu de communiquer directement avec la machine physique.
- Virtualisation totale (Hyperviseur type 2) : VirtualBox, Qemu, VMware
  - faire croire au SE qu'il s'exécute sur une machine physique : pas de modifs sur le système à part fournir des drivers pour du matériel générique



Paravirtualisation



Virtualisation Totale

Quelques Hyperviseurs

- Paravirtualisation : Xen, Hyper-V, ESX server (de VMware), ...
- Virtualisation totale : VMware server, VirtualBox, Microsoft Virtual PC, qemu, . . .

En pratique :

- Le disque de la machine virtuelle est souvent un (gros) fichier de la machine physique
- Les appels système sont redirigés vers le logiciel hyperviseur
  - simule une ou plusieurs cartes réseau,
  - une carte graphique de base,
  - réserver une partie de la mémoire vive etc.
  - Les entrées clavier et souris sont redirigées vers le système virtuel si dans la fenêtre concernée.

Les difficultés

- Architecture x86 donne des mauvaises habitudes : possède 18 instructions critiques : accessibles à partir des applications alors qu'elles permettent d'accéder aux ressources physiques...
- Un système virtuel ne doit pas avoir la possibilité de modifier les ressources physiques. L'hyperviseur doit intercepter ces instructions.
- L'adressage mémoire : une partie seulement de la mémoire physique doit être accessible
- La table des processus : le processus numéro 1 à un rôle spécifique (INIT, le premier ps et le dernier !), il faut donc que le ps n°1 de la MV devienne un autre numéro...

Avantages de la virtualisation

- Optimisation de l'infrastructure
  (charge moyenne d'un serveur est ≈ 10% (selon VMware)
  => récupérer les ressources restantes...)
- Réduction du nombre de machines physiques :
  => économies d'énergie, d'espace, de frais de ventilation,...
- Réduction des interruptions de service :
  => sauvegardes plus simples
- Facilité de migration :
  une machine virtuelle peut être exécutée sur une autre machine phy.
- Compatibilité :
  pas de dépendance entre serveur virtuel et machine physique.
- Sécurité par Cloisonnement

Inconvénients de la virtualisaion

- une machine physique unique... cela tombe en panne
- des performances moindres (suivant la puissance de la machine physique) mais les processeurs récents disposent de fonctions dédiées à la virtualisation.
- difficulté supplémentaire pour détecter et résoudre les problèmes : la couche de virtualisation vient s'ajouter aux autres.
- virtualisation parfois impossible : bases de données ont recours à de nombreux accès disques et le delai d'accès supplémentaire introduit par la couche de virtualisation peut être rédhibitoire.
- suivant l'outil de virtualisation, difficulté de virtualiser des systèmes d'exploitation prévus pour des archi. différentes (ex : nouveaux MACs à processeurs ARM)

Suite en TP.