Administration Réseaux et Systèmes

II. Installation système

Cas : OS Debian

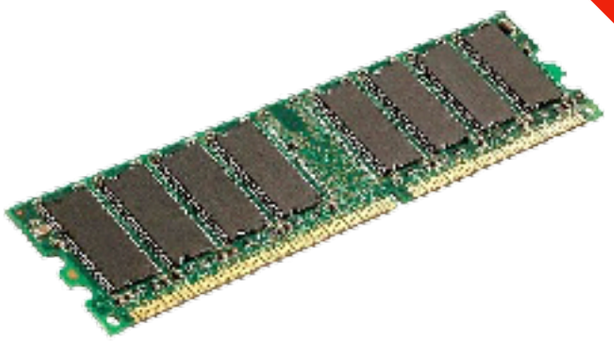
Sommaire

Machine Linux

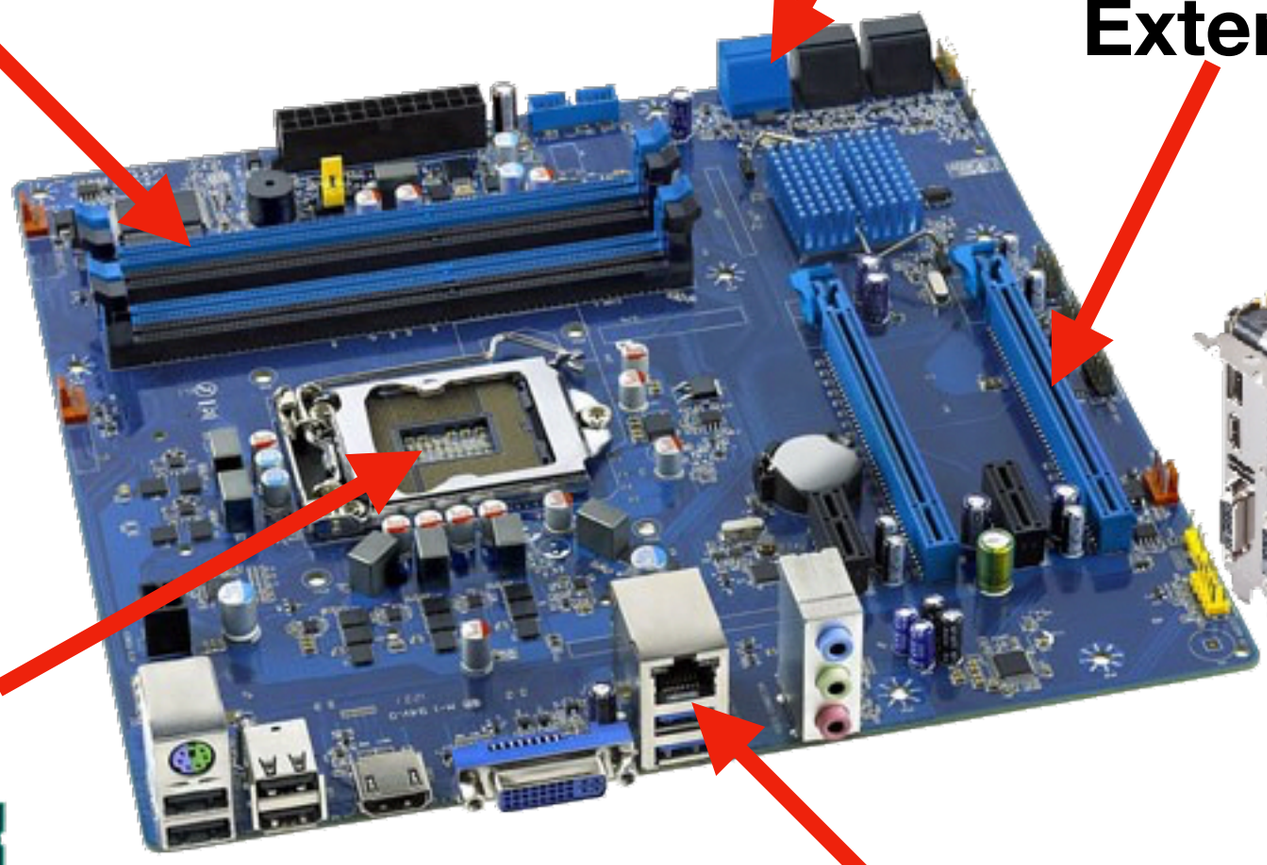
- Unité de calcul
    - Processeur + mémoire + carte mère
- Stockage
    - Disque dur, CD/DVD, USB
- Communication
    - Carte réseau, carte graphique, clavier, souris

Composants internes

Mémoire vive (RAM)



Disques SATA



Extension PCI

Disques SATA

Connecteurs externes

Unité de calcul

- Carte mère + processeur + mémoire
  - ➡ Serveurs professionnels : spécifique
  - ➡ Ordinateurs personnels : Intel ou AMD en 32 ou 64 bits mono ou multi-coeur
  - ➡ Architectures atypiques : Raspberry Pi, Arduino, …

Stockage

- Catégories pour les disques
    - ➡ En interne fixe, externe amovible, en réseau
    - ➡ Disque dur ou mémoire flash (solid state : SSD)
    - ➡ En réseau (NAS,
    - ➡ Interface
        - IDE (PATA) : en voie de disparition
        - SATA, SCSI (ce dernier peu répandu)
        - eSATA ou USB3 (disque externe)
        - iSCSI (disque en réseau)
        - NFS
        - WebDAV (disque sur le cloud)
    - ➡ Critères
        - Vitesse d’accès, de lecture et d’écritu

Séquence de démarrage

1. Lancement du BIOS (Basic Input/Output System) ou UEFI (BIOS amélioré)
    - Test rapide de la machine : mémoire, disques…
2. Recherche d'un système à charger
    - Parcours des disques démarrables : cd, disque, usb
    - Lecture du chargeur de système (bootLoader) situé sur le premier disque démarrable
        - NTLDR ou Winload sur Windows
        - Grub, Lilo ou syslinux sur Linux
3. Chargement et lancement du système

Partition de démarrage

➡ Le système se trouve sur le « disque » désigné pour le démarrage

➡ En réalité,

- Un disque physique est vu comme un ensemble de parties ou ***volumes*** : C: D: ... sur Windows, sda1, sda2 sur Linux
- Ces volumes sont appelés *partitions*
- Le système démarre sur l'une des partitions, celle qui a reçu le chargeur de système et qui est marquée comme « démarrable » (bootable)

Préparation à l'installation

- ➡ Disque vide ou contenant
  - Au moins une partition pour le système /
  - Éventuellement une partition pour swap (espace d'échange
  - Éventuellement une partition /home
  - Si UEFI, alors une partition spécifique
- ➡ Le CD, DVD ou Clé USB d'installation
  - Demander au bios de démarrer dessus
  - Ce périphérique contient le système à installer

Installation

➡ Le CD d'installation:

- Efface (formatte) les partitions
- Copie les fichiers du système
- Installe le bootLoader (grub)
- Créer et configure les comptes
- Configure l'accès au réseau (à minima)

➡ Au prochain démarrage, le système est installé

Être admin

- Les commandes d'administration se lancent avec un terminal en mode super utilisateur root ou en mode sudo (**s**uper **u**ser **do**)
- Exemple :

  sudo commande : donne tous les droits à la commande, mais seulement durant son exécution

  % sudo mkdir /temp

- Toute erreur peuvent avoir des conséquences irrémédiables

  % sudo rm -rf / temp

Examen du système

Information sur le système

- ➡ Une des sources d'informations principales sur le système se trouve dans le dossier **/proc**
- ➡ C'est un dossier dynamique : son contenu est généré à la volée par le système, il n'est pas stocké sur les disques
- ➡ Les fichiers qu'on y voit correspondent à des listes dans le système : ex : liste des processus

Examen du système

Information sur le système

- **Information sur la machine**

  ➡ Le fichier ***/proc/cpuinfo*** contient les caractéristiques du processeur : vitesse, nombre de cœurs...

  % cat /proc/cpuinfo

  ➡ Le fichier ***/proc/meminfo*** contient des informations sur la mémoire : présente (MemTotal), disponible (MemFree), utilisée par des copies des fichiers en mémoire (Cached).

  % cat /proc/meminfo

Examen du système

Information sur le système

- **Activité de la machine**

  ➡ **/proc/uptime** : nombre de secondes allumée et nombre de secondes à ne rien faire

  *La commande **uptime** donne la même information*

  ➡ **/proc/loadavg** : affiche la charge CPU actuelle, celle d'il y a 5 et 10 minutes, le nombre de processus actifs/le nombre total et enfin le PID du dernier processus créé

Périphériques

- ➡ Ils sont constitués, pour le système, d'une « puce » appelée ***chipset***, en fait une sorte de processeur spécialisé (NB : la carte mère possède également un chipset).
- ➡ Chaque chipset a un numéro d'identification :
  - [fabricant:numéro] (vendor et device)
  - 4 chiffres hexadécimaux normalisés
  - Ex : nVidia = vendor 10DE, ATI = 1002, ...
- ➡ Distinguer la marque qui est sur l'étiquette, du chipset qui est réellement dedans !

Périphériques

Périphériques internes

➡ La commande ***lspci*** affiche la liste des périphériques connectés sur le bus PCI (interne au PC)

- L'option -nn affiche les codes [fabricant:produit]

```
user@host$ lspci -nn
00:00.0 Host bridge [0600]: Intel Corporation Core Processor DRAM
Controller [8086:0044] (rev 02)
01:00.0 VGA compatible controller [0300]: NVIDIA Corporation GT218M
[10de:0a6c] (rev a2)
04:00.4 FireWire (IEEE 1394) [0c00]: Ricoh PCIe IEEE 1394 Controller
[1180:e832] (rev 03)
...
user@host$
```

Périphériques
Périphériques externes

➡ La commande ***lsusb*** affiche la liste des périphériques connectés sur le bus USB

- Elle affiche directement les identifiants du chipset
- Ex : 046D = Logitech, C526 = Wireless Mouse

```
user@host$ lsusb
Bus 002 Device 004: ID 0a5c:5800 Broadcom Corp. BCM5880 Secure
Applications Processor
Bus 002 Device 003: ID 046d:c526 Logitech, Inc. Nano Receiver
Bus 001 Device 002: ID 8087:0020 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

...
user@host$
```

Pilote de périphériques

- ➡ Un pilote de périphérique (device driver) c'est le module logiciel qui gère le périphérique
  - ▸ Dans Linux, on les appelle ***modules du noyau***
- ➡ Les pilotes reconnaissent les chipsets :
  - ▸ Au lancement, ils comparent les identifiants de ce qu'il y a sur la machine avec ceux qu'ils savent gérer
    - le pilote peut ou non gérer le périphérique
    - seuls les pilotes compétents sont actifs

Pilote de périphériques

Liste des pilotes

- La commande ***lsmod*** affiche les pilotes actifs: ce qui peuvent gérer les périphériques
- Certains pilotes sont utilisés par d'autres

```
user@host$ lsmod
Module                  Size  Used by
nvidia              11239877  44
bnep                   17535  2
rfcomm                 33471  0
bluetooth.            170002.  10, bnep, rfcomm

...
user@host$
```

Pilote de périphériques

Informations sur un pilote

➡ La commande ***modinfo*** affiche les informations sur un pilote :

- Fichier .ko de son code exécutable, dépendances
- Licence, n° de version, description, auteur
- Paramètres et options de lancement

```
user@host$ sudo modinfo bluetooth
filename:     /lib/modules/3.10-3-amd64/kernel/net/bluetooth/bluetooth.ko
license:      GPL
version:      2.16
description: Bluetooth Core ver 2.16
author:       Marcel Holtmann <marcel@holtmann.org>
depends:      rfkill,crc16
parm:         enable_hs:Enable High Speed support (bool)
...
user@host$
```

Périphériques
Utilisation d'un périphérie

- ➡ Une fois pris en charge par un pilote (module), certains périphériques peuvent devenir visibles dans le système sous la forme de fichiers spéciaux dans le dossier **/dev**
  - Exemple le 1er disque dur SATA : /dev/sda, ses partitions sont /dev/sda1, /dev/sda2, etc.
  - Un deuxième disque (ou DVD) SATA : /dev/sdb

Gestion des disques

Notion de volume

Nom donné à un espace de stockage de fichiers sur un disque dur (ou assimilé) : c'est généralement une ***partition*** dans un disque

Gestion des disques

Structure d'un disque

Vu du BIOS ou du pilote,

- ➡ un disque dur est composé de **secteurs** = tableaux de N octets avec N=512 en général, sauf SSD : 1024 octets
  - PB : confusion entre secteurs et blocs, voir plus loin
  - Les secteurs sont regroupés en « cylindres » : accès plus rapide sur le même cylindre
- ➡ Ces secteurs ont un numéro appelé **LBA** (logic block address)
- ➡ Le BIOS est capable de lire ou écrire n'importe lequel de ces secteurs : accès « aléatoire »

Gestion des disques

Structure d'un disque

Vu de Unix,

- ➡ Un disque dur (ou SSD, ou clé USB, ou CD-ROM, etc.) est représenté par un *fichier spécial* dans **/dev** :
  - Disques IDE (PATA), sur les vieilles machines :
    - /dev/hda pour le disque primaire maître,
    - /dev/hdb pour le disque primaire esclave,
    - /dev/hdc pour le disque secondaire maître,
    - /dev/hdb pour le disque secondaire esclave
  - CD-ROM IDE (PATA aussi) : /dev/cdrom0, /dev/cdrom1

Gestion des disques

Structure d'un disque

Vu de Unix,

➡ Depuis quelques années, on a :

- Disques et CD/DVD SATA (Serial ATA)
- Disques SCSI (ancien mais très performant)
- Clés ou disques amovibles USB

➡ Tous sont représentés par la même famille de noms :

- /dev/sda pour le premier disque
- /dev/sdb pour le deuxième disque
- etc

Gestion des disques

Structure d'un disque

Vu de Unix,

- ➡ Un disque entier est donc composé de secteurs de 512 octets et vu sous la forme d'un fichier spécial,
  ex : ***/dev/sda***
  - ▸ Lire ou écrire ce fichier = lire ou écrire le disque
  - ▸ Le fichier est continu : pas de frontière entre secteurs
- ➡ La commande ***dd*** permet de lire l'un ou l'autre des secteurs donné par son LBA, exemple :

  % dd if=/dev/sda ibs=512 skip=LBA ...

Voir détail plus loin avec un exemple complet

Gestion des disques

Structure d'un disque

MBR

1) Recopie du MBR dans un fichier

% *dd if=/dev/sda ibs=512 skip=0 count=1 of=mbr.bin*

- dd : copie directe d'octets entre fichiers
- if = fichier d'entrée à lire, ibs = taille des blocs à lire
- skip = LBA du secteur qu'on veut lire
- count = nombre de blocs de ibs octets à lire
- of = fichier de sortie, qui reçoit les count*ibs octets

Gestion des disques

Structure d’un disque

MBR

2) Affichage du contenu binaire

% od -Ax -tx1z -v mbr.bin

Les codes qu'on voit sont un programme en langage machine : EB 48 = JMP 0x4A, FA = CLI, ...

- Le site http://onlinedisassembler.com/ offre un désassembleur en ligne, on lui donne les codes et il indique quelles sont les instructions correspondantes
- C'est ultra technique (réservé aux hackers)

Les deux derniers octets sont 55 AA = MBR correcte

Les 4*16 derniers octets = table des partitions...

Gestion des disques

Partitions primaires

- ➡ Avant, le MBR spécifiait aussi le découpage du disque en 4 zones appelées partitions primaires
- ➡ Une partition = « de tel secteur à tel secteur » Table des partitions dans le MBR (4 partitions)

Gestion des disques

Partitions logiques

- ➡ Comme le MBR limitait à 4 partitions, ce qui est trop peu pour un système d'exploitation, on a rajouté la possibilité d'étendre la table :
  - ▸ Partitions dites « logiques » ou « étendues »
    - ▸ Placées dans le premier bloc de la dernière partition, c'est comme un second MBR
- ➡ **Problème** : si le disque est déjà occupé par 4 partitions primaires et que la dernière n'a pas été prévue en étendue, alors c'est fichu (ou c'est fait exprès).

Gestion des disques

Partitions avec UEFI

➡ Dans la norme UEFI, ce n'est plus le MBR qui contient la table des partitions (lui ne contient plus seulement qu'une seule partition pour représenter tout le disque), mais les 33 premiers secteurs du disque :

- La table est stockée sur les secteurs LBA=2 à 34 et est aussi recopiée à la fin du disque
- Le secteur LBA=1 est un bloc d'en-tête appelé GPT Header : GUID Partition Table

  => Il peut y avoir 128 partitions au lieu de seulement 4 (ça semble suffisant)

Gestion des disques

Interêt des partitions

- ➡ Comme pour les dossiers : pour séparer les choses :
  - Partitions pour le système (/ et /boot)
  - Partition pour les comptes (/home)
  - Partition pour la mémoire virtuelle (swap)
  - Partition pour les sites web hébergés (/var/www)
  - Partition pour les fichiers temporaires (/tmp)
  - Partitions pour archiver les comptes
  - Partitions pour un autre système d'exploitation...

Gestion des disques

Pourquoi séparer

➡ En cas de panne ou de virus

- Seules les partitions du système seront à effacer et réinstaller
- Panne : défaillance du disque, les fichiers de la partition sont perdus mais pas ceux des autres partitions

➡ En cas de débordement :

- Ex : serveur FTP : si des utilisateurs déposent trop de fichiers, ça bloque seulement cette partition

Gestion des disques

Création des partitions

➡ Plusieurs logiciels permettent de définir les partitions :

- **fdisk** : outil de base, en ligne de commande et avec un menu simpliste
- **cfdisk** : interface améliorée pour fdisk
- **gparted** : logiciel très complet et ergonomique
-

Gestion des disques
Procédure d'installation

- ➡ L'installation de Linux est automatisée à l'exception du partitionnement des disques :
  - ▸ Préserver Windows ou pas
  - ▸ Découper l'espace libre
- ➡ Ne pas choisir l'installation automatique mais passer en mode manuel => lancement de gparted
- ➡ Linux = 3 partitions au moins
  - ▸ / contiendra le système, les logiciels
  - ▸ /home contiendra les comptes
  - ▸ swap pour la mémoire virtuelle, taille ≳ RAM

Gestion des disques
Partition vue de Unix

➡ Chaque partition d'un disque est associée à un fichier spécial : /dev/disqueN (N : 1, 2...)

- /dev/sda1 pour la première partition du 1er disque
- /dev/sdc2 pour la deuxième partition du 3e disque etc.

➡ Ça pose un problème si on échange les disques (branchement sur la carte mère) : le nom de leurs partitions change (sda1 → sdb1)

Gestion des disques
Nommage par GUID ou UUID

➡ Au lieu de nommer les partitions ainsi, on préfère maintenant les identifier par un numéro unique écrit en hexadécimal :

- un GUID (global identifier) sur Windows, ex : B0AA-B27F
- Ou ***UUID*** dans le monde Unix, ex : 2F0A84C0-B92F-41D1-671A-6466554876F1
- Ce n° est inscrit au début de chaque partition

➡ La commande ***uuidgen*** affiche un nouvel UUID

Gestion des disques

La commande blkid

➡ La commande blkid affiche les UUID et le type de formatage des disques connectés au système :

```
user@host$ sudo blkid
/dev/sda1: LABEL="live-rw" UUID="e1314788-3f9e-41fe-
852f-9e2f42de7c9a" TYPE="ext4"
/dev/sr0: LABEL="LILI LiveCD" TYPE="iso9660" /dev/loop0:
TYPE="squashfs"

...
user@host$
```

Gestion des disques
Noms UUID des partitions

➡ Dans ce système de nommage, les volumes sont dans le dossier /dev/disk/by-uuid/ : (ce sont des liens vers les /dev/sdXN)

```
user@host$ ls -l /dev/disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 nov.  25 08:52 e1314788-3f9e-
41fe-852f-9e2f42de7c9a -> ../../sda1


...
user@host$
```

Gestion des disques
Label des partitions

- ➡ Il y a également la possibilité d'utiliser une étiquette (ou *volume label*) pour nommer les partitions : un mot de 16 lettres au maximum
  - Ex : ROOT, HOME, DATA…
- ➡ On le définit quand on formate la partition

Gestion des disques

Formatage des partitions

- ➡ Format d'une partition défini son contenu (dossiers et fichiers)
- ➡ Le format spécifie la manière d'écrire sur les secteurs d'une partition
  - Windows : FAT32, exFAT, NTFS
  - Unix : ext (ext2, ext3, ext4), reiserfs, xfs, btrfs

  - ext3 est la version journalisée de ext2
  - ext4 est la version améliorée de ext3 (très grands disques et gros fichiers...)

Gestion des disques

Formatage des partitions

- ➡ Formater => tout effacer et préparer pour de nouveau fichiers
- ➡ Commande:

% sudo mkfs.type -L label /dev/partition

Ex : sudo mkfs.ext4 -L HOME /dev/sdb1

Ex : sudo mkfs.xfs -L DATA /dev/sdc3

- Option -L : définit le nom (label) de la partition.
  NB : l'UUID ne peut pas être choisi, il est généré automatiquement.

Gestion des disques
Vérification d’une partition

➡ Il arrive de temps en temps qu'un disque dur commette une erreur de lecture ou d'écriture

- Des fichiers peuvent être corrompus : contenu altéré (du texte aléatoire à la place)
- Des dossiers peuvent être incomplets : certains fichiers sont tronqués ou perdus

➡ Il faut employer un outil de vérification

- Automatique : tous les 23 démarrages de l'ordinateur ou tous les 30 jours.

Gestion des disques
Vérification d'une partition

➡ La commande fsck vérifie la cohérence d'une partition

% sudo fsck -f /dev/hdb1

- ▸ Ne pas employer sur une partition montée

➡ Les fichiers égarés en cas de corruption de leur dossier sont mis dans lost+found : ils viennent d'i-nodes qui n'étaient plus dans aucun dossier (leur ancien dossier ayant été corrompu)

Gestion des disques

Montage d’une partition

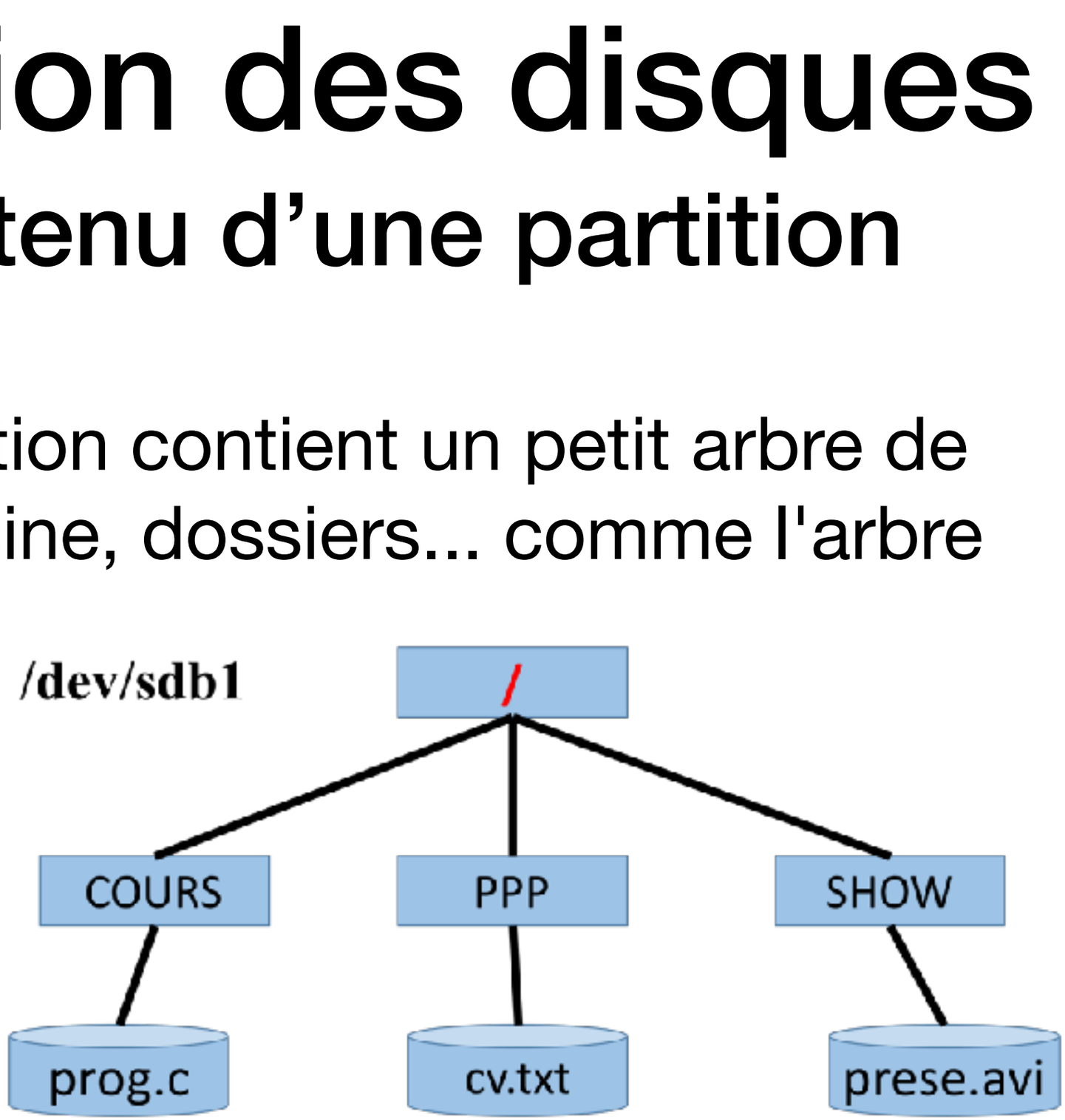
CONCEPTS:

➡ Pour accéder aux fichiers d'une partition, on doit la « monter » (**mount**)

➡ Cela fait apparaître son contenu dans l'arbre des fichiers Unix

Gestion des disques
Contenu d'une partition

➡ Toute partition contient un petit arbre de fichier : racine, dossiers... comme l'arbre Unix

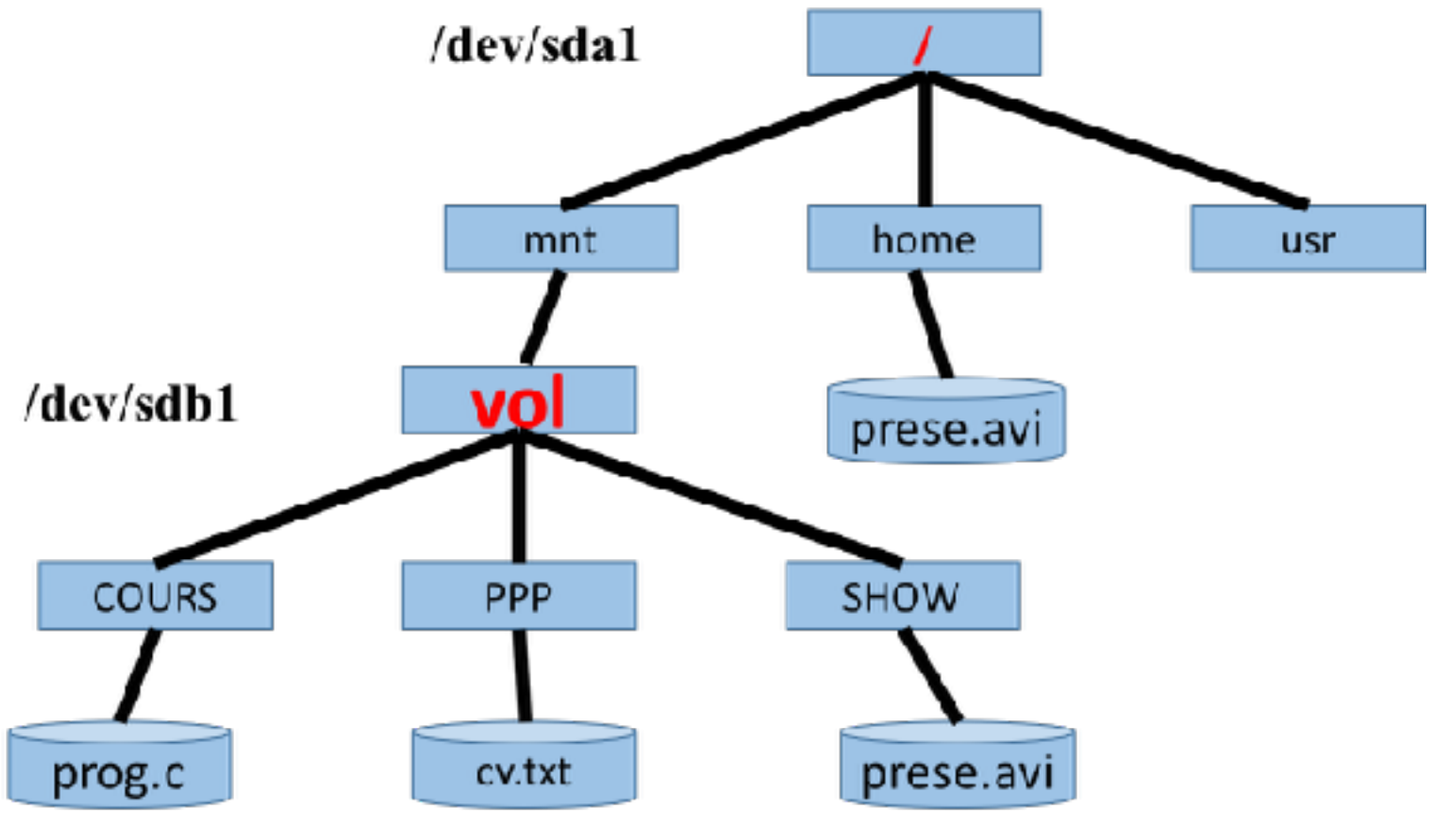


Gestion des disques

Point de montage

➡ Créer un dossier vide généralement dans /mnt ou /media. Ex: /mnt/vol

➡ Monter la partion /dev/sdb1 sur /mnt/vol



Gestion des disques

Accès au partition

- ➡ L'accès au volume monté est transparent : tout se passe comme si tous les fichiers faisaient partie d'un seul et même arbre
- ➡ Le système gère les accès aux partitions :
  - /home/moi/edt.txt accès à /dev/sda1
  - /mnt/vol/COURS/prog.c accès à /dev/sdb1
- ➡ Attention : on n'a plus accès à l'ancien contenu du point de montage !

Gestion des disques

Montage d'une partition: mise en pratique

➡ Pour monter une partition, il faut :

- Créer ou avoir un dossier vide, ex : /mnt/dossier qui définit le point de montage
- Connaître le fichier spécial /dev/partition correspondant à la partition
- Employer la commande en tant qu'administrateur :

  mount -t format /dev/partition /mnt/dossier

➡ Ex :

% sudo mount -t xfs /dev/sdb2 /mnt/vol

Gestion des disques

Liste des montages

➡ Commande **mount** affiche les montages actuels sur le système

```
user@host$ mount
proc on /proc type proc (rw)
/dev/sdb2 on /home type ext4 (rw)
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
```

➡ Le fichier **/etc/mtab** et le fichier **/proc/mounts** affichent aussi la liste des montages actifs

➡ Pour démonter une portion on utilise la commande **umount**

% sudo umount /mnt/vol

% sudo umount /dev/sdb1

Gestion des disques

Info sur les partions

➡ La commande **df** affiche la liste des partitions et leur taux de remplissage

- L'option -h les affiche en Mo ou Go

```
user@host$ mount
Sys. de fichiers blocks de     1K       Utilisé     Disponible     Uti%    Monté sur
/dev/sda1          19553560 11138404     7398836     61%     /
udev                  10240        0       10240     0%      /dev
tmpfs                398500     1080      397420     1%      /run
tmpfs                  5120        0        5120     0%      /run/lock
tmpfs                1597540      420     1597120     1%      /run/shm
/dev/sdb1           53359108 46587712     4037828     93%     /home
```

Gestion des disques

Montage définitif et auto

➡ Le fichier **/etc/fstab** contient la liste des montages qui doivent être faits au démarrage du système (option auto) ou qui peuvent être faits ultérieurement (options noauto,user)

➡ La commande ***mount -a*** est lancée au démarrage et monte tous les volumes notés auto

Gestion des disques

Montage définitif et auto

Syntaxe de /etc/fstab
Les lignes de /etc/fstab contiennent 6 « mots » :

Filesystem	Point de montage	type	options	Dump	Pass
/dev/sdb1	/	Ext4	Defaults	0	0

1. Le périphérique associé à la partition, ex : /dev/sdb1
2. Le point de montage, ex : /
3. Le format de la partition, ex : ext4
4. Les options, ex : defaults, noauto, user
5. Option pour l'archivage avec dump 0 pas de sauvegarde, 1 avec sauvegarde
6. Un 1 pour /, 2 pour les autres : ordre de vérification par fsck, 0 empêche la vérification

FIN COURS 2