Cours Linux

Architecture PC pour linux

alan.simon@free.fr

historique

PC : Personal Computer

IBM PC en 1981(pc-g) (boot sur disquette)

Basé sur les microprocesseur x86 d'Intel

Composant standard de divers fabriquants

- Carte mère
  - Processeur
  - Mémoire
  - Bus pour cartes filles (graphique, son, FC, SAS/Raid, etc...)
  - Interfaces : rs232, usb, etc..
  - Bios

POST : Power On Self Test

Vérifie l'initialisation du CPU et stabilité d'alim

Vérifie le check sum du microcode

Vérifie la RAM intégrée à la carte mère (640Ko)

Vérifie l'intégrité de la carte mère (controleur d'interruptions)

Initialisation des I/O minimum (clavier et carte graphique)

Affichage à l'écran ou beep si erreur

lancement du BIOS

Bios

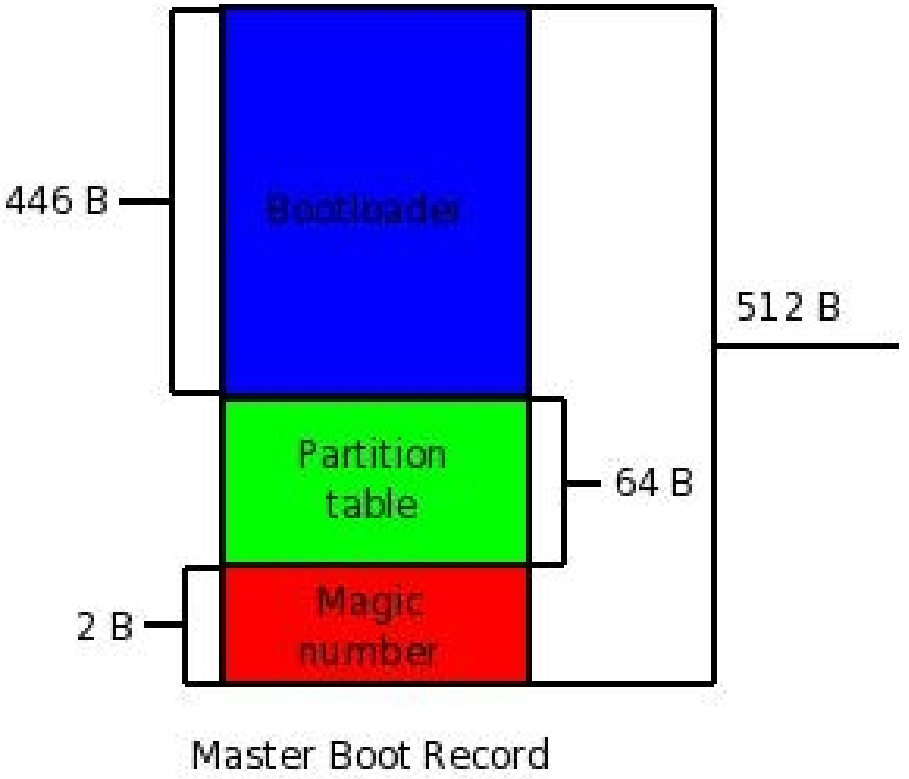
Basic Input Output System

- Microcode de la carte mère sur ROM ou eeprom
- Identification des périphériques
- Initialisation bas niveau des composants
  - POST et bios secondaire
- Selectionne le périphérique de boot
- Charge et execute le MBR

MBR : Master Boot Record

Le MBR : 512 octets sur le premier secteur d'un disque

- Le boot loader
- Table de partition du disque

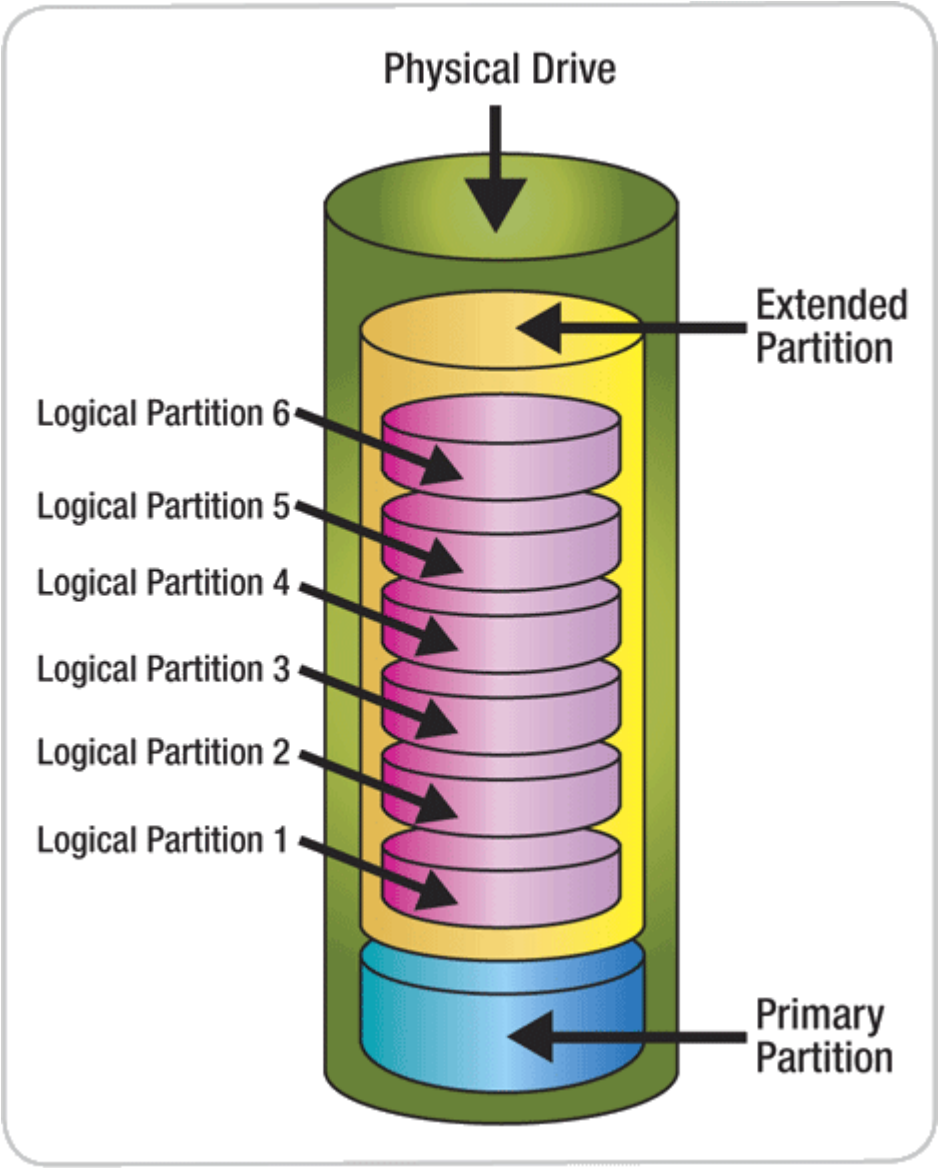


Master Boot Record

Table de partition

Découpage bas niveau du disque

- 4 partitions primaires :
  - reconnue par le bios
  - Définies sur 16 octets
- Une Partition étandue
  - 1 MBR vide puis
  - Série de partitions logiques chainés par des EBR (64octets)
    - Pointeur sur la partition
    - Pointeur sur l'EBR suivant



Lecture de la table de partitions

La table primaire : “sudo dd if=/dev/sda bs=1 count=64 skip=446 | hexdump”

0000000 2080 0021 fe**83** ffff **0800 0000** 0000 08f6

- Type **83** commence au bloc 0x000000**8**

0000010 fe00 ffff fe**05** ffff **0ffe 08f6** e802 005a

- Type **05** (étendue) commence au bloc **0x8f60ffe**

La table secondaire, elle commence au bloc 0x8f60ffe soit 150343678 en décimal “echo $((0x8F60FFE))”. “sudo dd if=/dev/sda bs=512 count=1 skip=150343678 | dd bs=1 count=64 skip=446 | hexdump”

0000000 fe00 ffff fe**82** ffff **0002 0000** e800 005a

- Partition logique de type 82 (linux swap) qui commence au bloc : 0x000002 de la partition étendue donc : 0x8f61000

0000010 0000 0000 0000 0000 0000 0000 0000 0000

- Pas d'EBR suivant, il n'existe qu'une partition logique

EFI / UEFI

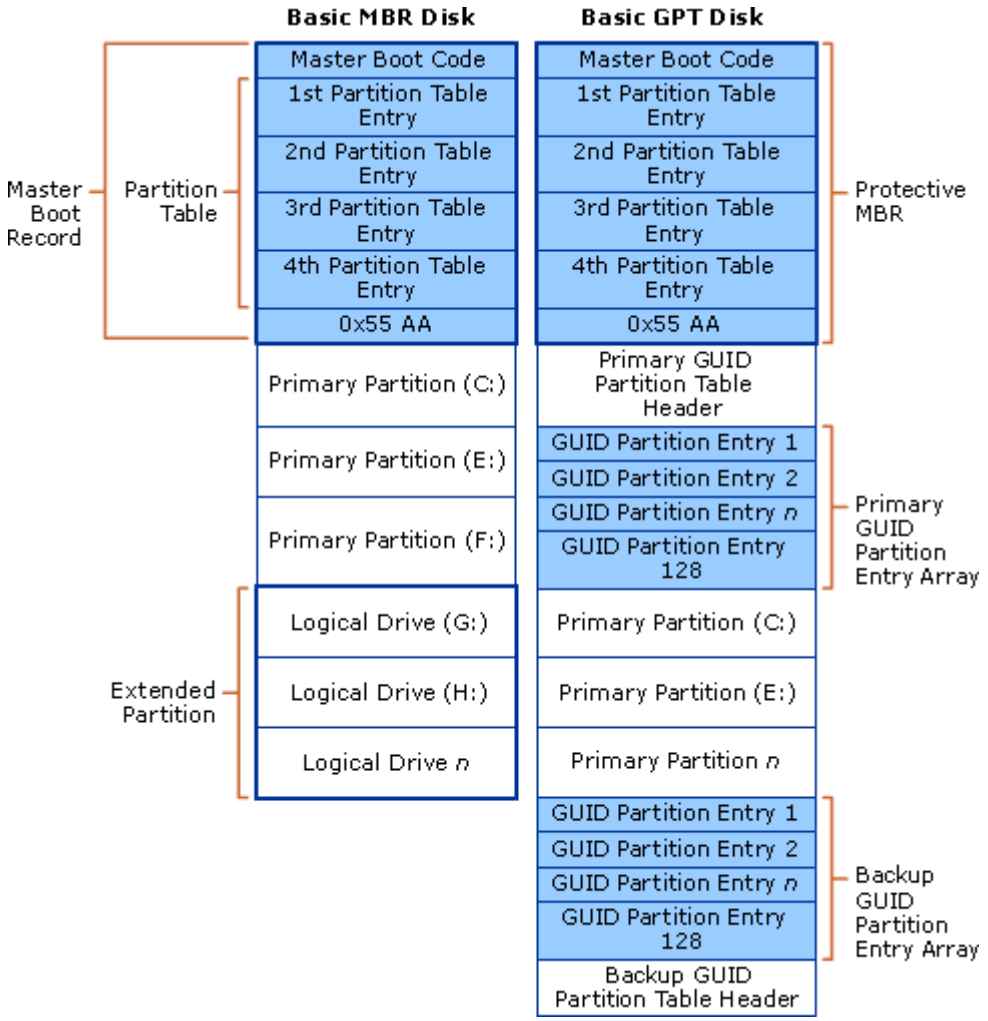
Unified Extensible Firmware Interface

- Remplace le Bios pour certaine architecture de CPU (Intel Itanium, ARM)
- Palie aux limitation du bios (cpu acceptant un mode 16bits, et 1M de mémoire adressable)
- Codé en C et non en assembleur
- Micro système d'exploitation
- Permet l'utilisation du partitionement GPT (GUID Partition Table)

!!! Offre la fonction Secure boot !!! qui empèche le chargement d'OS et driver non signé numériquement. (Microsoft est l'entité qui vend ces signatures)

Guid Partition Table

- Concerve le MBR
  - Le MBR protecteur
  - Première partition :
    - Tout le reste du disque
    - Dispose de l'entête des partition primaire
- Table des partition en début de la partition primaire
- Copie en fin de disque



alan.simon@free.fr

Le boot loader

Problématique :

- Un ordinateur exécute un programme placé dans sa mémoire.
- Un programme ne peut être placé en mémoire sans l'intervention d'un autre programme.

Le boot loader effectue cette action

- Architecture X86 : MBR
- Autre architecture : microcode de constructeur

Grub

Stage1

- Sur le MBR
- Charge a partir des 1024 premiers cylindres du disque stage 1.5 ou stage 2

Stage 1.5

- Code suplémentaire permettant de charger stage2

Stage 2

- Menu de choix d'OS

Notion de filesystems

Les systèmes GNU/linux et Unix ne dispose que d'une seule arborescance

la racine (root) est /

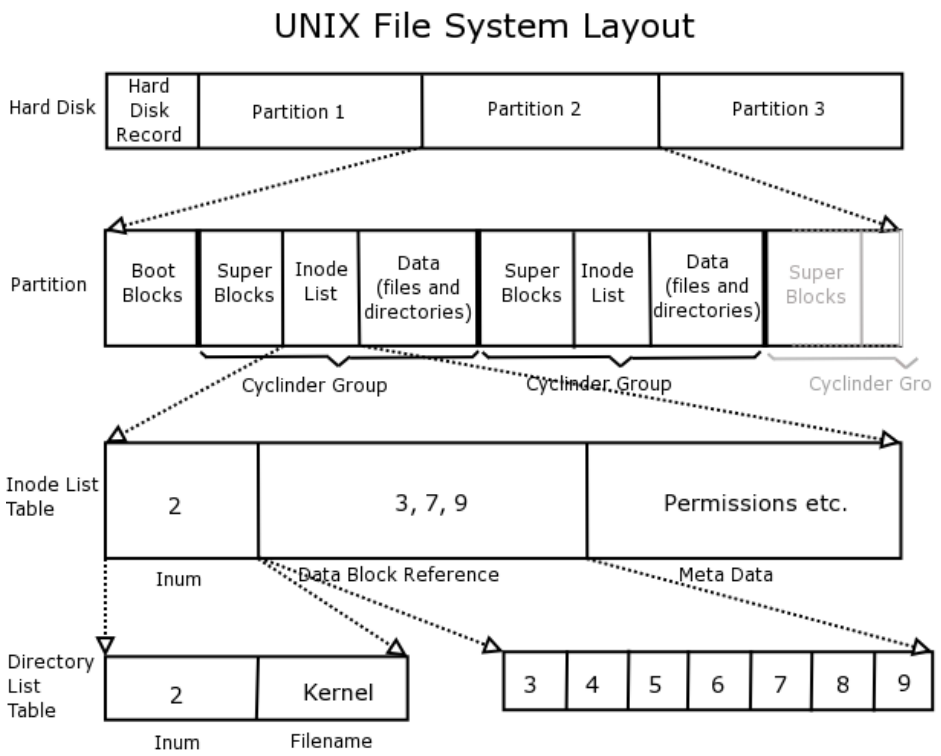
Les autres volumes ou système de fichier sont montés dans l'arborescence

- Point de montage : un répertoire
- Le volume / FS est alors une sous arbo

Exemple :

- Au montage d'une clé usb
- La clé est accèssible sous /media/disk-1 ou /mnt/usb-disk
- Les fichiers et répertoires $file se situent sous /media/disk-1/$file

Présentation
filesystem / FS / système de fichiers



- Organisation des données sur disque
- Gestion des droits et des accès
- Arborescence

alan.simon@free.fr

Rappel Arborescence Unix

/ Répertoire racine du système de fichiers.

/boot/ Configuration noyau utilisés durant le boot

/dev/ Fichiers spéciaux de périphérique

/etc/ Procédures et fichiers de configuration du système.

/root/ Répertoire personnel du compte root

/bin/ Programmes utilisateur minimes

/sbin/ Programmes systèmes

/lib/ Librairie minime

/usr/ La majorité des utilitaires et des applications utilisateur. (/usr/bin)

/usr/bin **/usr/local**

/var/ Fichiers de traces, fichiers temporaires, et fichiers tampons

/var/log/ /var/spool/ /var/tmp/ /var/lib/

/tmp/ Fichiers temporaires

/home/ Repertoire parent des homedir utilisateur

/mnt/ **Point de montage temporaire**

/proc/ Le système de fichiers pour les processus (procfs)

Partitionning sous linux

/boot : contiens les binaire nécessaire au boot

- Kernel
- Boot loader

/ : racine de l'arborescance : monté au boot

Swap : espace de stockage pour libèré la RAM

- Mémoire virtuelle

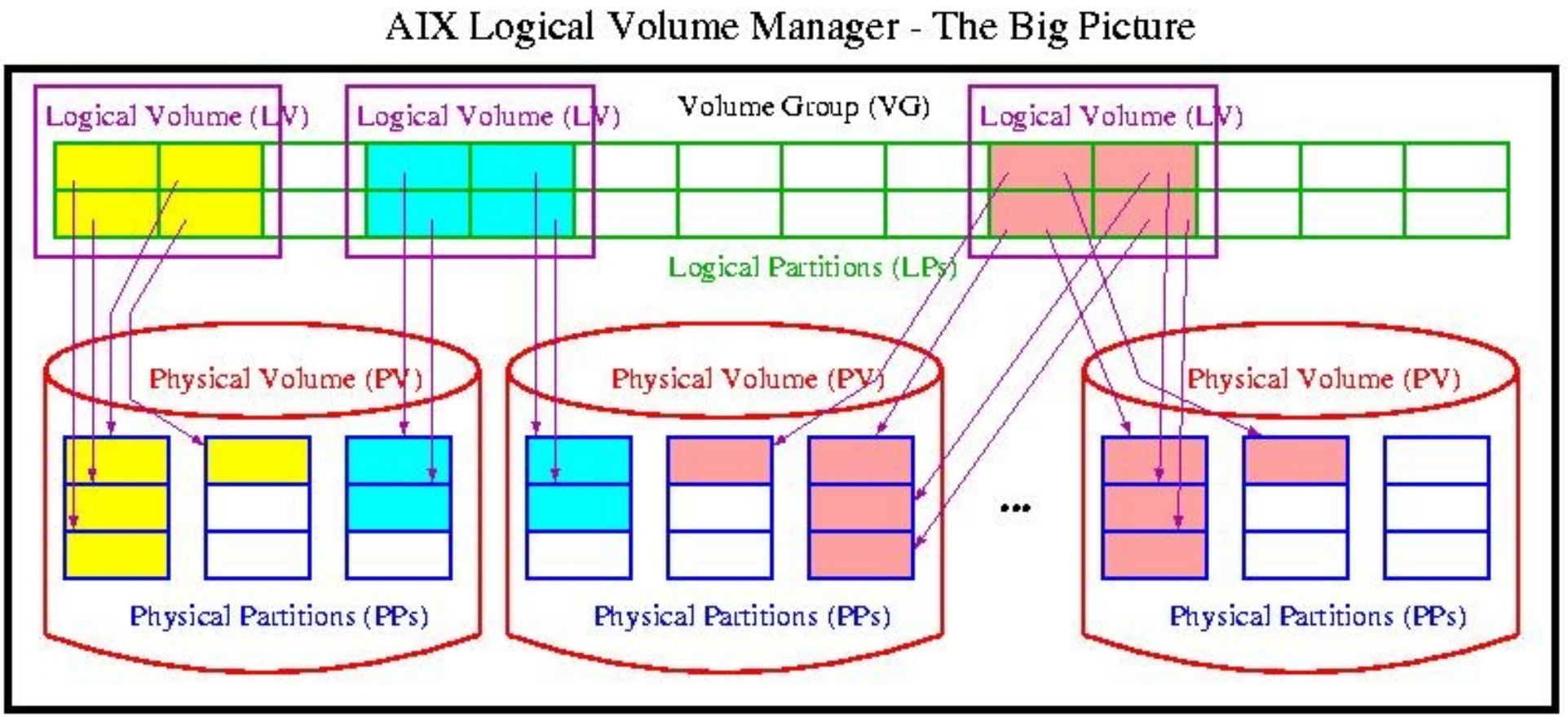
Partitionning idéale : Ça n'existe pas !

- 2 partitions : /boot + LVM2
- / sous lvm : taille maximum de la distribution ~ 10GB
- Swap : sous lvm : 2xRamsize ~ 2GB max
- Ajoute des volumes suivant les besoins

Présentation LVM

- Fonctions :
  - Couche d'abstraction entre materiel et volume
  - Fonction avancée :
    - Snapshot
    - Raid 1
- Avantages
  - Gestion du materiel décorèlé du logiciel
  - Support à la migration de données

LVM



alan.simon@free.fr

Vocabulaire

PV : Physical Volume : Unité de stockage source pour LVM

VG : Volume Group : Une configuration LVM

- Stocké sur un/les PV
- Defini la taille des extent (non modifiable)
- Défini les limites en nombre de pv et lv (illimité sous LVM2)

PE : Physical Extent : Unité minimum de stockage physique

- Les PV sont découpé en PE

LV : Logical Volume : Unité de stockage fournie par LVM

LE : Logical Extent : Unité minimum de stockage sur un LV

Partitionning sous linux 1/2

Partitionner fortement un systeme permet de séparer / Optimiser les usages

- Limiter les impacts : un FS full n'impacte pas les autres FS
- Utiliser des options de montages spécifique :
  - Quotas
  - ReadOnly
- Sauvegarde par dump complet du FS

Risques : FS-full

Ne pas partitionner un systeme permet de simplifier l'administration de celui-ci

- Moins de FS : moins de FS-full
- Réduction de surveillance

Risques : / full

Partitionning sous linux 1/2

Optimiser le partitionning : partitionner en fonction de l'usage du systeme.

- Limiter les FS full, Séparer les usages
  - /opt ou /opt/appli1, /opt/appli2
  - /data ou /data/database1, /data/database2
  - /var ou /var/www/siteweb1, /var/www/siteweb2

- En cas de besoin supplémentaire il reste possible de migrer une sous arbo dans un FS dédié pour cloisonner l'usage.

- On utilise LVM!