Système d'Exploitation

Processus de démarrage
Structure des répertoires Unix

Juan Angel Lorenzo del Castillo
Florent Devin

ING1 GI-GM
2020-2021

Plan

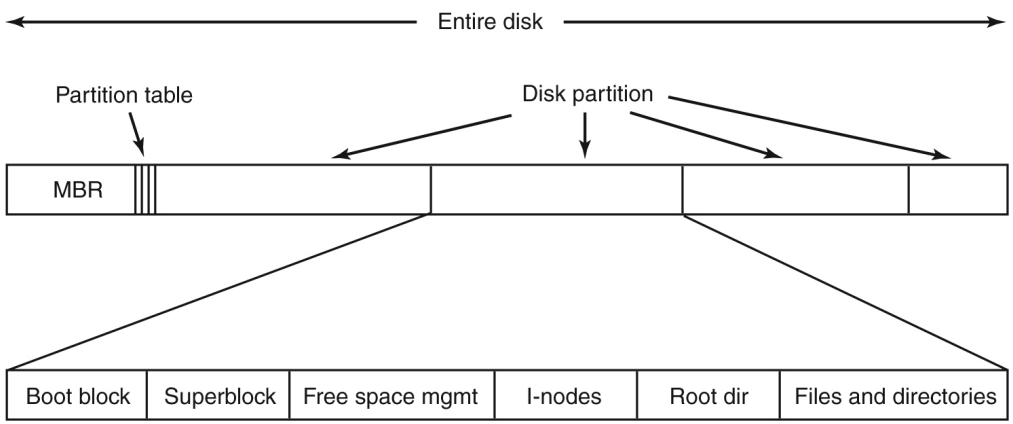
Processus de démarrage

UNIX System V

- Une des premières versions commerciales d'UNIX.
- Développée par AT&T (1983).
- La plupart des systèmes propriétaires UNIX descendent directement de System V.
- Rivalité avec BSD (*Berkeley Software Distribution*)
- Apparition de la norme POSIX (*Portable Operating System Interface*) : effort de standardisation

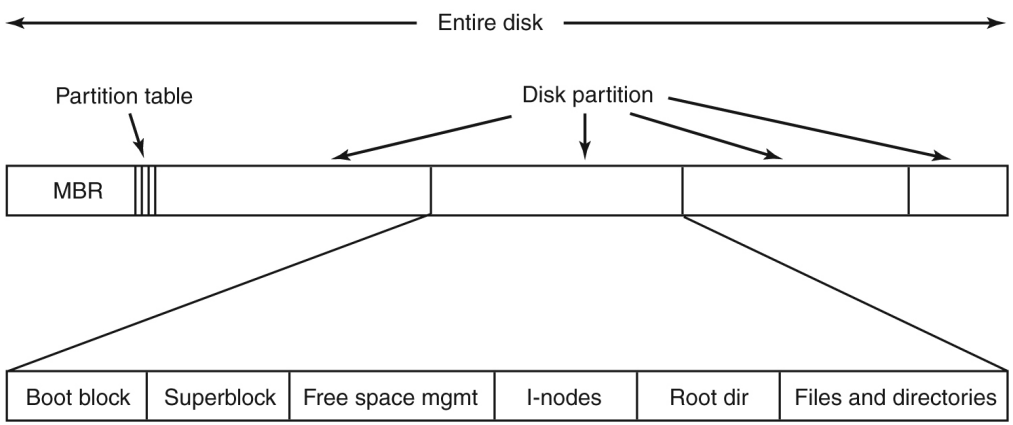
Démarrage de l'ordinateur

- BIOS contenu dans une ROM
- Disque dur divisé en partitions, chacune avec un système de fichiers.
- Le secteur 0 ou MBR (*Master Boot Record*) est lu au moment du démarrage
  - La fin du MBR contient la table de partitions
  - L'une d'entre elles est marqué comme active

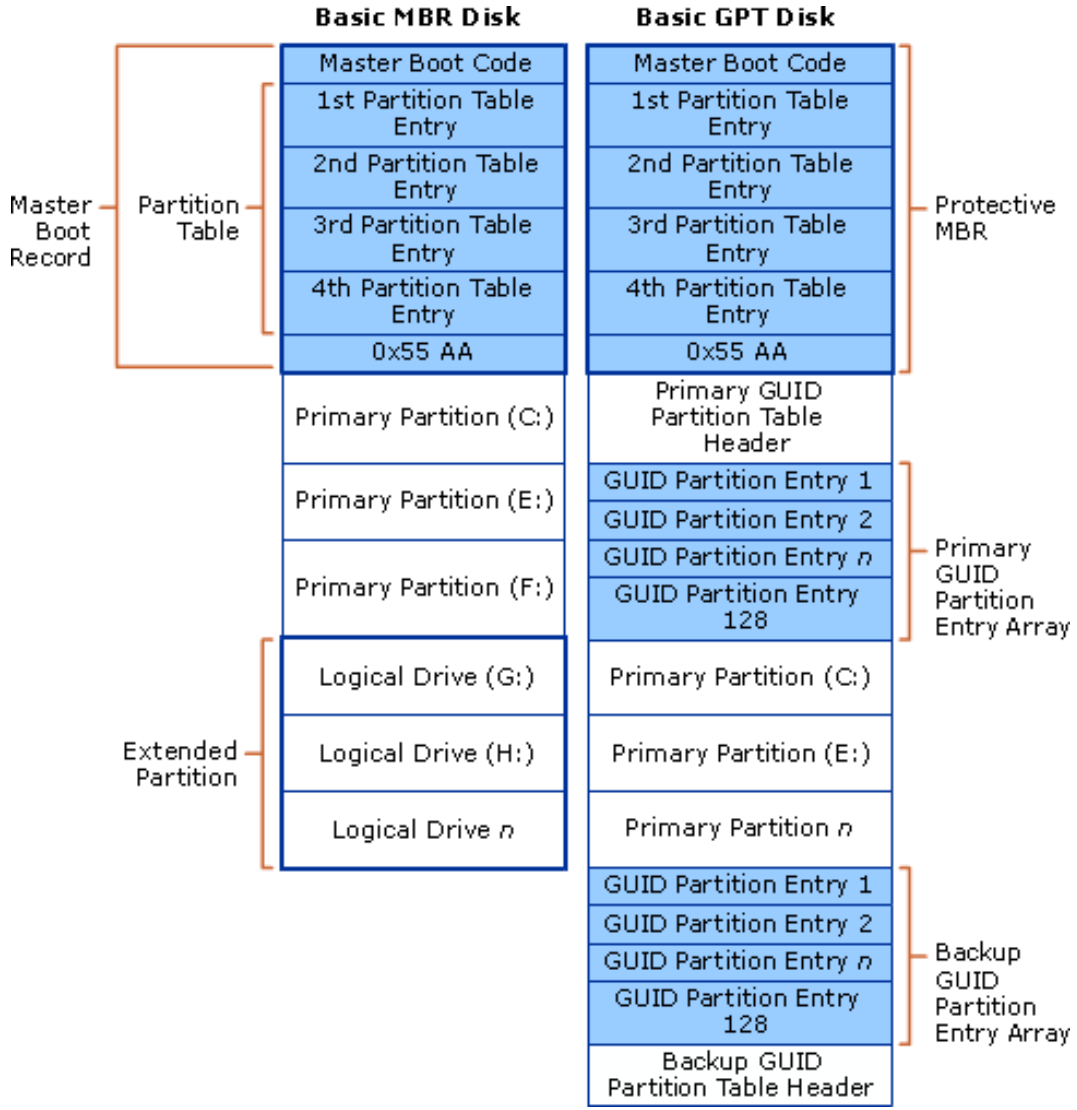


Démarrage de l'ordinateur

- Démarche :
  1. La BIOS cherche, lit et exécute le *first-stage boot loader* contenu dans le MBR du disque dur.
  2. Le MBR cherche la partition active, lit son premier bloc (*bloc de démarrage*, *Volume Boot Record* ou *Boot Block*) et l'exécute
  3. Le *second-stage boot loader* dans le bloque de démarrage (GRUB2, par exemple) charge le SE contenu dans ce partition
- Dans des systèmes UEFI, le noyau du SE peut être exécuté directement (donc il n'y a pas besoin d'un *boot loader*)



Démarrage de l'ordinateur



Source : technet.microsoft.com

Processus de démarrage *SysVinit*

Initialisation :

1. Décompression du noyau linux
2. Initialisation des périphériques (disque, réseau, son, . . .)
3. Configuration “automatique” des pilotes
4. Montage du système de fichier racine
   - Lecture seule : permet de vérifier s'il y a des erreurs
5. Lacement du processus `init` (PID 1)

Processus de démarrage *SysVinit*

Initialisation (cont.) :

6. Lecture du fichier `/etc/inittab`
7. Lancement des processus à partir des **scripts shell** contenus dans `/etc/rc<n>.d/`
8. Commutation vers le mode multi utilisateurs
9. Exécution de `getty` pour chaque console
10. Remontage du système de fichier en lecture-écriture
11. Exécution des processus de services (`daemon`)
12. Construction d'une arborescence de processus

Arrêt du système

Procédé d'arrêt :

- Nécessité d'arrêter le système "proprement"
- Risque de corruption du système de fichiers (utilisation de cache en écriture)
- Commande : `shutdown` provoque l'arrêt
- Commande : `shutdown -r` ou `reboot` provoque le redémarrage
- Démontage des systèmes de fichiers, arrêt des processus utilisateurs et service, enfin le système est stoppé

Processus `init`

- Lancement du noyau : démarrage du processus `init`
- Père de tous les processus : PID 1
- Lecture du fichier `/etc/inittab`
  - Informations de la forme :

    `id:runlevel:action:processus`

  - `id` : identifieur
  - `runlevel` : niveau de démarrage
  - `action` : manière de lancer le processus
  - `processus` : ce qui est lancé

Processus `init`

- **`action :`**
  - `respawn` : relance le processus une fois celui-ci terminé
  - `wait` : attendre le fin du processus avant de continuer
  - `boot` : doit être lancer au démarrage uniquement
  - `ctrlaltdel` : processus lance quand séquence tapée

Processus `init`

- **`runlevel`** (Niveau de démarrage) **:**
  - État dans lequel se trouve le système
    - 0 Halt (arrêt de la machine)
    - 1 Single-user
    - 2 Ne pas utilisé
    - 3 Full multi-user
    - 4 Ne pas utilisé
    - 5 Multi-user complète avec du mode graphique
    - 6 Redémarrage
  - `init` recherche la ligne qui contient `initdefault` et lance le runlevel associé. ou `init`

Processus de démarrage *systemd*

- Simplicité d'utilisation
- Maîtrise du système moins éparpillée
- L'utilisation de scripts de lancement est découragée
- Services pilotés par des fichiers texte de configuration à la place des scripts de SysVinit

Apports de *systemd*

- Allocation fine des ressources (processeur, mémoire, E/S, etc) aux services (cgroups)
- Surveillance améliorée
- Log plus complet (au format binaire) et normalisation du format
- Possibilité de *démoniser* tout processus en le relançant automatiquement s'il s'arrête
- Séparation claire entre les services fournis par la distribution et les services créés par l'administrateur
- Simplification du processus d'empaquetage des services
- Centralisation du développement des briques de base du système

Source : `linuxfr.org`

systemd

`systemd` est adopté par :

- Fedora et RHEL (*Red Hat Entreprise Linux*)
- openSUSE, Mageia, Arch Linux
- SLES (SUSE Linux Entreprise Server)
- Debian 8.0 et Ubuntu 15.04

Plan

Structure des répertoires Unix

Structure des répertoires Unix

/ ← racine
|
|___ /bin ← fichiers binaires (exécutables).
|___ /etc ← fichiers de configuration.
|___ /media ← dispositifs CDROM, clé USB, etc.
|___ /root ← home de l'administrateur du système.
|___ /dev ← fichiers des périphériques.
|___ /home ← répertoires home des utilisateurs.
|___ /sbin ← fichiers binaires n'accessibles qu'au root.
|___ /tmp ← fichiers temporaires.
|___ /var ← fichiers qui peuvent varier (p.ex logs).
|___ /usr ← logiciel pour l'utilisateur.
|___ /lib ← librairies dynamiques.
|___ /boot ← fichiers de démarrage du système.
|___ /mnt ← dispositifs éjectables.

Structure des répertoires Unix

- `usr` : logiciel de seule lecture, disponible pour tout le système. Structure similaire à la racine

/usr
|
|___ /bin ← fichiers binaires d'utilisateur.
|___ /local ← programmes compilés d'utilisateur.
|___ /src ← sources des utilisateurs.
|___ /share ← fichiers indépendants de l'architecture matérielle.
|___ /include ← fichiers *header* des applications.
|___ /lib ← liens dynamiques à des librairies.

Plan

Mise à jour du système

Mise à jour en Debian/Ubuntu

apt-get update

La commande update permet de resynchroniser un fichier répertoriant les paquets disponibles et sa source. Ces fichiers sont récupérés aux endroits spécifiés dans /etc/apt/sources.list (...). On doit toujours exécuter une commande `update` avant les commandes `upgrade` ou `dist-upgrade`.

apt-get upgrade

La commande `upgrade` permet d'installer les versions les plus récentes de tous les paquets présents sur le système en utilisant les sources énumérées dans `/etc/apt/sources.list`. Les paquets installés dont il existe de nouvelles versions sont récupérés et mis à niveau. **En aucun cas des paquets déjà installés ne sont supprimés** ; de même, des paquets qui ne sont pas déjà installés ne sont ni récupérés ni installés.

apt-get dist-upgrade (**à ne pas faire sur vos machines CY Tech !**)

La commande `dist-upgrade` effectue la fonction `upgrade` en y ajoutant une gestion intelligente des changements de dépendances dans les nouvelles versions des paquets ; apt-get possède un système « intelligent » de résolution des conflits et il essaye, quand c'est nécessaire, de mettre à niveau les paquets les plus importants aux dépens des paquets les moins importants. Le fichier /etc/apt/sources.list contient une liste de sources où récupérer les paquets désirés.