

Préparation LPI

Exam 102

101.2. Démarrage du système

Sommaire

- Etapes du démarrage d'un système Linux
- Options du noyau
- Journaux

- Poids : 3
- Options du chargeur de démarrage et du noyau
- Connaître les différentes étapes du démarrage d'un serveur Linux
- Contrôler les évènements et messages dans les journaux

1. Partie matérielle - BIOS-POST



2. Chargeur de démarrage (GRUB - LILO)



3. Chargement du noyau



4. Init – niveaux d'exécution



5. Lancement des processus de service en fonction du niveau d'exécution

- **BIOS** (Basic Input/Output System) : recherche des périphériques présents
- **POST** (Power On Self Test) : test des composants matériels au démarrage. En cas de problème : séquence de bips (parfois un message vocal) indiquant l'origine du problème (car pas de vidéo dispo à cette étape)
- Parcours des périphériques de stockage pour démarrer le système ou boot réseau (PXE)
- Exécution du *chargeur d'amorçage* (bootstrap loader) :
 - recherche d'un secteur d'amorçage (premier secteur de la partition active, MBR – Master Boot Record)
 - chargement et exécution de ce secteur qui permet de charger réellement le bootloader
 - exécution de GRUB, LILO ou SYSLinux

- Etapes de démarrage BIOS-POST
 - ✓ La carte mère reçoit le signal « Power Good » de l'alimentation
 - ✓ Tous les processeurs (x86) récupèrent les 16 derniers octets de la mémoire conventionnelle à partir de FFFF0h. Contient un pointeur vers l'emplacement réel du code du BIOS dans la ROM (compatibilité différentes tailles de ROM BIOS)
 - ✓ Exécution du POST. Si ok : appel interruption INT 19 pour découverte des périphériques connectés à la carte mère
 - ✓ Recherche de la carte vidéo et chargement du bios de la carte situé à l'adresse C000h
 - ✓ Recherche et chargement des bios des autres périphériques (floppy, disque, cartes SCSI,...)

- Etapes de démarrage BIOS-POST ...
 - ✓ Test de la mémoire
 - ✓ Inventaire des éléments matériels du système et vérification de fonctionnement
 - ✓ Affichage d'un résumé des informations récoltées
 - ✓ Recherche d'un périphérique pour démarrer le système d'exploitation
 - ✓ Recherche du Master Boot Record (MBR)
 - ✓ Chargement du code contenu dans le MBR à l'adresse 0000:7C00 et exécution de ce code. A partir de cette étape, le BIOS « passe la main » au système d'exploitation

- Etapes de démarrage BIOS-POST ...
 - ✓ Recherche de la partition active depuis la table de partitions primaires
 - ✓ Chargement du secteur de boot de la partition active à l'@ 0000:7C00
 - ✓ Exécution de ce code : cela correspond à début du chargement du bootloader. La suite dépend du bootloader (LILO ou GRUB dans notre cas)

- **MBR** Master Boot Record
- Se trouve dans les 512 premiers octets du disque (cyl 0, tête 0, secteur 1)
 - Contient la table des partitions primaire
 - Contient le boot code
- Table des partitions primaires
 - 4 entrées de 16 octets = 64 octets
 - 4 partitions primaires max
 - La partition active contient le code pour continuer la séquence de démarrage

- Dans le cas de GRUB
 - Le code de boot tient dans les 512 octets du premier secteur : stage1
 - stage1 : but trouver et charger le stage2
 - stage2 : code réel du bootloader qui ne peut tenir dans 512 octets
- GRUB fait quelques vérifications puisse passe le processeur du mode réel (mode de fonctionnement 8086) au mode protégé.
- GRUB charge le noyau : fichier compressé et dont l'en-tête contient le code de décompression et de chargement en mémoire

- Le noyau est informé de la présence d'un disque d'initialisation (Init Ram Disk au format initramfs). C'est une archive cpio compressée avec gzip.
- Au démarrage, le noyau crée un Ram Disk puis copie le contenu de l'archive
- Ce Ram Disk fournit au noyau un file system temporaire qui va fournir :
 - Le chargement de modules permettant la prise en charge d'autres systèmes de fichiers
 - De fonctionnalités spécifiques : RAID logiciel, LVM
 - Les pseudo-fs : /proc, /sys, /tmp, /dev
- A un moment, le système de fichier racine permute sur le réel

- **GRUB** (GRant Unified Bootloader)
- **LILO** (LILO)
- Premier logiciel (après le BIOS) exécuté au démarrage
- Rôle : charger et transférer le contrôle au noyau d'un système d'exploitation
- Permet le démarrage de plusieurs systèmes d'exploitation différents (en particulier double-boot Windows-Linux)
- Installé sur le MBR (plusieurs OS) ou premier secteur de la partition active

- Possibilité de définir des options lors du chargement du noyau
- Les modifications se font soit en ligne de commande depuis le bootloader soit d'une manière définitive dans le fichier de configuration du bootloader
- Options courantes
 - `init=num` : où num correspond au niveau d'exécution
 - `noapic` : désactivation de la prise en charge de l'APIC (Advanced Programmable Interrupt Controller)
 - `acpi=off` : désactivation de la prise en charge de l'ACPI (économie énergie)
 - `mem=x[K|M|G]` : définit la taille mémoire allouable au noyau en Kilo, Mega ou Gigaoctets
 - `S, s, 1, single` : passage en mode mono-utilisateur

- Journaux ou fichiers de log : fichiers textes
- Localisés dans le répertoire `/var/log`
- Dans certains cas, des sous-répertoires dédiés à des services ou applications
 - `/var/log/cups`
 - `/var/log/mail`
 - `/var/log/httpd`
- Localisation paramétrable dans le fichier de conf du service
- Sur un serveur, intérêt à ce que `/var` soit dans une partition distincte de `/`
- Intérêt de la rotation des fichiers de log pour limiter l'espace utilisé (cf. plus loin)

- Journaux importants :
 - `boot.log` : messages relatifs à la séquence de boot
 - `cron` : messages relatifs à l'utilitaire cron
 - `maillog` : messages relatifs au système de courrier
 - `messages` : la presque totalité des messages qui ne sont pas dirigés dans un fichier spécifique
 - `auth` ou `secure` : messages relatifs à l'authentification des utilisateurs sur le système (locale ou distante avec telnet ou ssh)
 - `dmesg` : messages envoyés par le noyau
 - `yum.log` : messages relatifs au gestionnaire de paquetage yum

- `dmesg` : également une commande permettant d'examiner le kernel ring buffer
 - Taille du buffer limitée à 16384 octets
 - `-n level` : demande à n'afficher sur la console que les messages d'un certain niveau
 - `dmesg -n 1` : affiche uniquement à la console les messages de type PANIC
 - `-s` : vide le buffer
- Différence entre le fichier `/var/log/dmesg` et la commande `dmesg`
 - `/var/log/dmesg` : uniquement les messages générés par le noyau durant la séquence de boot
 - `dmesg` : affiche tous les messages du noyau au cours de la vie du système


```
[root@localhost ~]# dmesg | wc -l
```

```
584
```

```
[root@localhost ~]# cat /var/log/dmesg | wc -l
```

```
500
```

- Action : insertion d'un clef USB

```
[root@localhost ~]# dmesg | wc -l
```

```
604
```

```
[root@localhost ~]# cat /var/log/dmesg | wc -l
```

```
500
```

- Examen des journaux
- Chaque évènement (1 évènement par ligne) enregistré dans un journal contient
 - Date et heure
 - Nom machine à l'origine du message
 - Service ou utilisateur qui génère le message
 - Texte du message

```
Feb 1 22:09:02 localhost rsyslogd: [origin software="rsyslogd" swVersion="2.0.2" x-  
pid="2206" x-info="http://www.rsyslog.com"] [x-configInfo udpReception="No"  
udpPort="514" tcpReception="No" tcpPort="0"] restart
```

```
Feb 1 22:45:50 localhost gconfd (franck-3194): Sortie
```

```
Feb 1 22:45:50 localhost shutdown[9195]: shutting down for system halt
```

```
Feb 1 22:45:50 localhost NetworkManager: <info> Deactivating device eth1.
```

```
Feb 1 22:45:50 localhost NetworkManager: <info> eth1: canceled DHCP transaction,  
dhclient pid 3456
```

```
Feb 1 22:45:51 localhost kernel: ipw2200: Failed to send ASSOCIATE: Already sending  
a command.
```

- Commandes utiles pour examiner les journaux
 - `less`
 - `tail`
 - `head`
 - `more`
 - `tail -f` : affichage dynamique des nouvelles lignes qui apparaissent dans le journal.
Pratique pour suivre l' volution « en ligne » d'un journal
Ctrl + C pour quitter
 - `grep`
 - `-i` : pour inclure minuscules et majuscules

```
[root@localhost ~]# grep -i usb /var/log/messages | wc -l  
504
```

```
[root@localhost ~]# grep usb /var/log/messages | wc -l  
299
```

