

Introduction aux systèmes GNU/Linux

S23E05 inetdoc.net



Philippe Latu / Université Toulouse 3

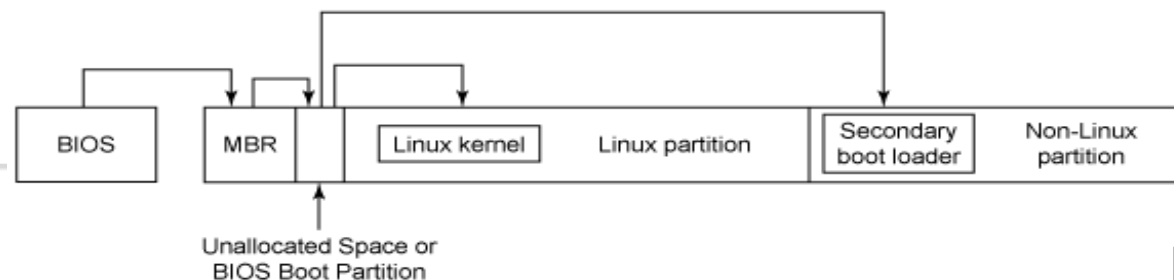
Document sous licence GNU FDL v1.3
<http://www.gnu.org/licenses/fdl.html>

Plan séance

- Analyser l'initialisation du système et des services
 - Présenter les étapes de l'initialisation d'un système
 - Identifier le rôle du gestionnaire d'amorce
 - Différencier les espaces mémoire noyau & utilisateur
 - Analyser la gestion des modules pilotes de périphériques
 - Reconnaître les processus et les services lancés au démarrage
- Manipuler sur machines virtuelles & conteneurs
 - Étudier les services lancés au démarrage
 - Identifier les ressources allouées : mémoire, CPU, réseau & stockage

Initialisation du système

- POST & BIOS
 - POST → Power On Self Test
 - BIOS → Basic Input Output System
 - Premiers programmes appelés par le processeur
 - Analyse de la configuration matérielle
- Recherche d'un système d'exploitation
 - Ordre de scrutation défini dans les paramètres du BIOS
 - Pour chaque périphérique désigné → lecture du Master Boot Record (MBR)
 - Si le code Boot Loader est présent dans un MBR → initialisation du système



Initialisation du système

- Master Boot Record
 - Contient le code Boot Loader
 - Désigne la partition d'amorçage
 - Accède au gestionnaire d'amorce
- Gestionnaire d'amorce → **GRUB2**
 - Grand Unified Boot Loader
 - Code modulaire à deux «étages» → Boot Loader + Shell
 - Support GPT → Grand Partition Table
 - Support systèmes de fichiers + RAID + LVM
 - Fichier de configuration lu à chaque initialisation
 - Support Linux Unified Key Setup (LUKS)

Initialisation du système

- Partition ou répertoire /boot
 - Gestionnaire d'amorce & noyau(x)

```
$ lsblk
NAME          MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINTS
sr0            11:0    1 1024M  0  rom
nvme0n1       259:0    0   60G  0  disk
├─nvme0n1p1   259:1    0   512M  0  part /boot/efi
├─nvme0n1p2   259:2    0  58,5G  0  part /
└─nvme0n1p3   259:3    0   976M  0  part [SWAP]
```

```
$ ls -lX /boot/efi
drwxr-xr-x 2 root root 4096 Aug 11 10:00 grub
-rw-r--r-- 1 root root  128 Aug 11 10:00 config-6.5.0-4-amd64
-rw-r--r-- 1 root root  128 Aug 11 10:00 initrd.img-6.5.0-4-amd64
-rw-r--r-- 1 root root  128 Aug 11 10:00 System.map-6.5.0-4-amd64
-rw-r--r-- 1 root root  128 Aug 11 10:00 vmlinuz-6.5.0-4-amd64
```

Gestionnaire d'amorce

Partie monolithique du noyau

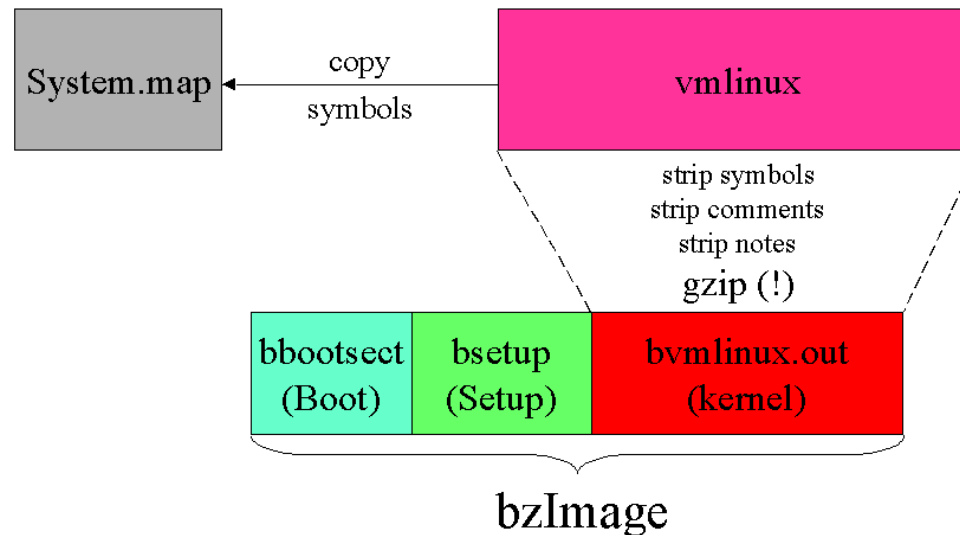
```
$ ls -lX /boot/grub/x86_64-efi/part*
-rw-r--r-- 1 root root  128 Aug 11 10:00 partmap.lst
-rw-r--r-- 1 root root  128 Aug 11 10:00 parttool.lst
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_acorn.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_amiga.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_apple.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_bsd.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_dfly.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_dvh.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_gpt.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_msdos.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_plan.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_sun.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 part_sunpc.mod
-rw-r--r-- 1 root root  128 Aug 11 10:00 parttool.mod
```

Modules relatifs aux types de partitions

Initialisation du système

- Partie monolithique du noyau
 - Fichier bzimage → image compressée du noyau
 - Initialisation → éclatement en plusieurs zones mémoire discontinues

Anatomy of bzImage



Initialisation du système

- Contenu du disque RAM initial → initrd
 - Shell minimal + boîte à outils busybox

```
$ lsinitramfs /boot/initrd.img-5.9.0-5-amd64 |less
```

- Objectif : accéder au système de fichiers racine
 - Firmwares de pilotage de composants
 - Modules nécessaires
 - Contrôleurs stockage
 - Systèmes de fichiers
 - Cryptographie
 - Interfaces réseau

Compiler un paquet noyau

▪ Prérequis

- L'utilisateur `etu` doit appartenir au groupe système `src`
- L'arborescence `/usr/src` appartient au groupe `src`
- Le masque des permissions du répertoire `/usr/src` est `rwxrwsr-x`
- Les paquets à installer sont `fakeroot` & `libncurses-dev`

```
$ cd /usr/src
$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.86.tar.xz
$ tar xf linux-5.4.86.tar.xz
$ ln -s linux-5.4.86 linux && cd linux

$ make menuconfig
```

Reprise de la configuration du noyau de la distribution par défaut

```
$ make -j8 bindeb-pkg
$ cd ..
$ sudo dpkg -i linux-image-5.4.86_5.4.86-1_amd64.deb linux-libc-dev_5.4.86-1_amd64.deb
```


Compilation d'un nouveau noyau

- Interface de configuration
 - Arborescence assez complexe et difficile à appréhender
 - Noyau de la distribution → configuration déjà fonctionnelle
 - Pour débiter → procéder par modifications successives

```
.config - Linux/x86 5.4.13 Kernel Configuration
> Networking support > Networking options

Networking options
-----
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----)
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
[*] built-in [ ] excluded <M> module < > module capable

* (-)
[*] TCP/IP networking
[*] IP: multicasting
[*] IP: advanced router
[*] FIB TRIE statistics
[*] IP: policy routing
[*] IP: equal cost multipath
[*] IP: verbose route monitoring
[ ] IP: kernel level autoconfiguration
<M> IP: tunneling
<M> IP: GRE demultiplexer
<M> IP: GRE tunnels over IP
[*] IP: broadcast GRE over IP
[*] IP: multicast routing
v (+)

<Select> < Exit > < Help > < Save >
```

Gestionnaire d'amorce

- Configuration révisée
 - À chaque nouvelle installation de noyau
 - À chaque nouvelle version des outils
- Script `update-grub` & personnalisation

```
$ sudo update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.86
Found initrd image: /boot/initrd.img-5.4.86
Found linux image: /boot/vmlinuz-5.4.0-2-amd64
Found initrd image: /boot/initrd.img-5.4.0-2-amd64
done
```

nouveau noyau

noyau de la
distribution

```
$ ls -1X /etc/grub.d/
00_header
05_debian_theme
10_linux
20_linux_xen
30_os-prober
30_uefi-firmware
40_custom
41_custom
README
```

personnalisation
du menu

Gestionnaire d'amorce

▪ Applications

- Quelle la version du noyau en cours d'exécution ?
 - Chercher l'option utile de la commande `uname`
- Quel est la version du paquet de noyau installé ?
 - Rechercher dans la liste des paquets installé l'empreinte `linux-image`
- Quel est le fichier de journalisation dédié aux messages du noyau ?
 - Rechercher dans les options de la commande `journalctl`
- Comment obtenir la liste des modules du noyau chargés en mémoire ?
 - Rechercher dans la liste des commandes du paquet `kmod`
- Quel est le rôle de la commande `dmesg` ?

```
$ sudo dmesg -HT
```

Initialisation du noyau

- Séquence d'initialisation du noyau
 - Séquence d'initialisation
 - Architecture
 - Mémoire virtuelle
 - Ordonnanceur → horloges et interruptions
 - Paramètres de la ligne de commande
 - Systèmes embarqués → Raspberry Pi
 - Ouverture du disque RAM initial
 - Lancement de la boîte à outils busybox
 - Chargement des modules propres au système
- Distinction entre noyau monolithique ou modulaire
 - Noyau monolithique → architecture figée → smartphone par exemple
 - Noyau modulaire → architecture évolutive → périphériques d'un PC

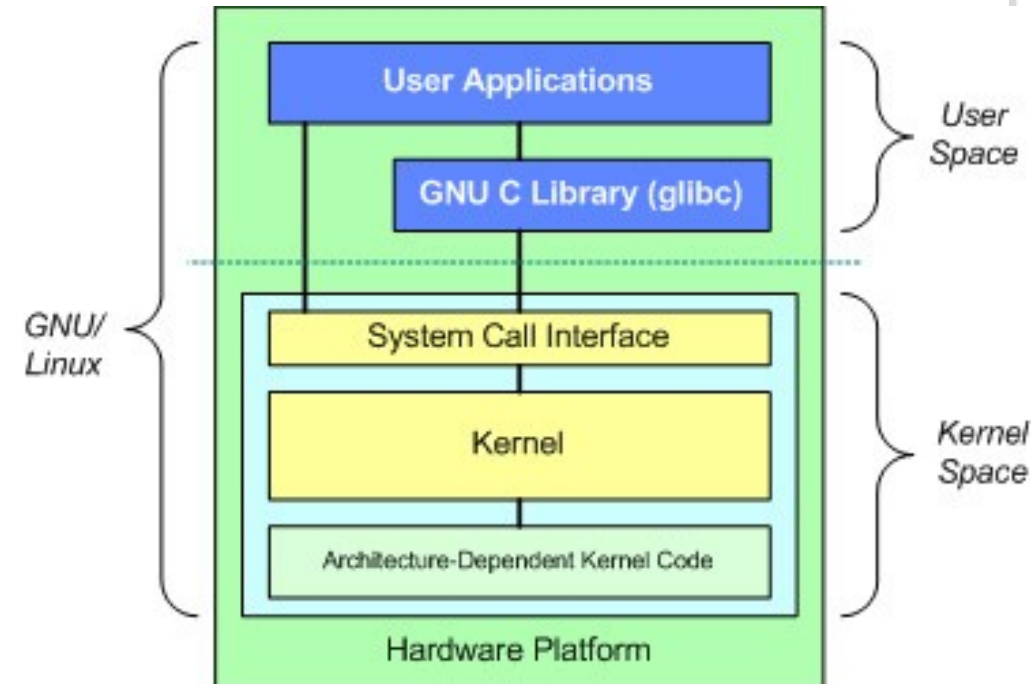
Représentation des périphériques

- Contexte historique Unix

- Par principe, tout est fichier
 - tout périphérique matériel est représenté dans le système de fichiers
- L'arborescence `/dev` contient la liste des périphériques
- Entrées générées avec la commande `mknod`

- Contexte contemporain

- Découpage en deux espaces mémoire
 - Kernel space → noyau
 - Userspace → utilisateur
- Évènements matériels automatiquement « traduits » dans l'espace utilisateur



Représentation des périphériques

- **KernelSpace** → espace noyau
 - 2 Systèmes de fichiers virtuels → répertoires `/sys` et `/proc`
 - `sysfs`
 - Exportation des informations du noyau vers l'espace utilisateur
 - `procfs`
 - Informations sur les processus
 - Réglages des sous-systèmes du noyau → sous-système réseau !
- **Userspace** → espace utilisateur
 - Démon `udev`
 - Actions déclenchées par les informations `sysfs`
 - Règles de configuration dans `/etc/udev`

Représentation des périphériques

- Démon udev et unité de disque
 - Informations sur le matériel

```
$ lspci | grep -i nvm  
00:04.0 Non-Volatile memory controller: Red Hat, Inc. QEMU NVM Express Controller (rev 02)
```

- Informations collectées par udev pour l'unité de disque `/dev/nvme0`

```
$ udevadm info --query=all --name=/dev/nvme0  
P: /devices/pci0000:00/0000:00:04.0/nvme/nvme0  
M: nvme0  
R: 0  
U: nvme  
D: c 246:0  
N: nvme0  
L: 0  
E: DEVPATH=/devices/pci0000:00/0000:00:04.0/nvme/nvme0  
E: DEVNAME=/dev/nvme0  
E: NVME_TRTYPE=pcie  
...
```

Représentation des périphériques

- Démon udev et interface réseau
 - Informations sur le matériel

```
$ lspci | grep -i ethernet  
00:01.0 Ethernet controller: Red Hat, Inc. Virtio 1.0 network device (rev 01)
```

- Informations collectées par udev pour l'interface `enp0s1`

```
$ udevadm info --query=all --path /sys/class/net/enp0s1  
P: /devices/pci0000:00/0000:00:01.0/virtio0/net/enp0s1  
M: enp0s1  
R: 1  
U: net  
I: 2  
E: DEVPATH=/devices/pci0000:00/0000:00:01.0/virtio0/net/enp0s1  
E: SUBSYSTEM=net  
E: INTERFACE=enp0s1  
E: IFINDEX=2  
E: USEC_INITIALIZED=2020069  
E: ID_NET_NAMING_SCHEME=v253  
E: ID_NET_NAME_MAC=enxb8adcafe0000  
E: ID_NET_NAME_PATH=enp0s1
```

Accès aux autres attributs avec la commande
`udevadm info --attribute-walk --path=/sys/class/net/eth0`

Manipulations sur les modules

- Correspondance entre matériel et nom de module

```
# lspci -k
...
00:19.0 Ethernet controller: Intel Corporation Ethernet Connection (3) I218-V (rev 03)
      Subsystem: Intel Corporation Ethernet Connection (3) I218-V
      Kernel driver in use: e1000e
      Kernel modules: e1000e
```

- Outils du paquet kmod
 - lsmod → liste des modules chargés en mémoire
 - modprobe → (dé)chargement d'un module et de ses dépendances en mémoire
- Fichier /etc/modules
 - Liste des modules à charger obligatoirement

Gestion des services – systemd

- Initialisation des processus ou services
 - Solution historique → runlevels
 - Ensemble de scripts shell lancés séquentiellement
 - Solution actuelle pour les distributions → **systemd**
 - Processus **init**
 - Contrôle d'unités (**units**) → processus + conditions d'exécution
 - Gestion des dépendances entre les unités
 - Les services sont la propriété d'un groupe de contrôle (**control group**) → **cgroup**
 - Suivi des processus à partir des informations de service
 - Configuration des ressources CPU, réseau, mémoire et I/O → **SLAs**
 - Gestion du démarrage/arrêt des services



Gestion des services - systemd

- Applications → contrôle de l'état de l'ensemble des services

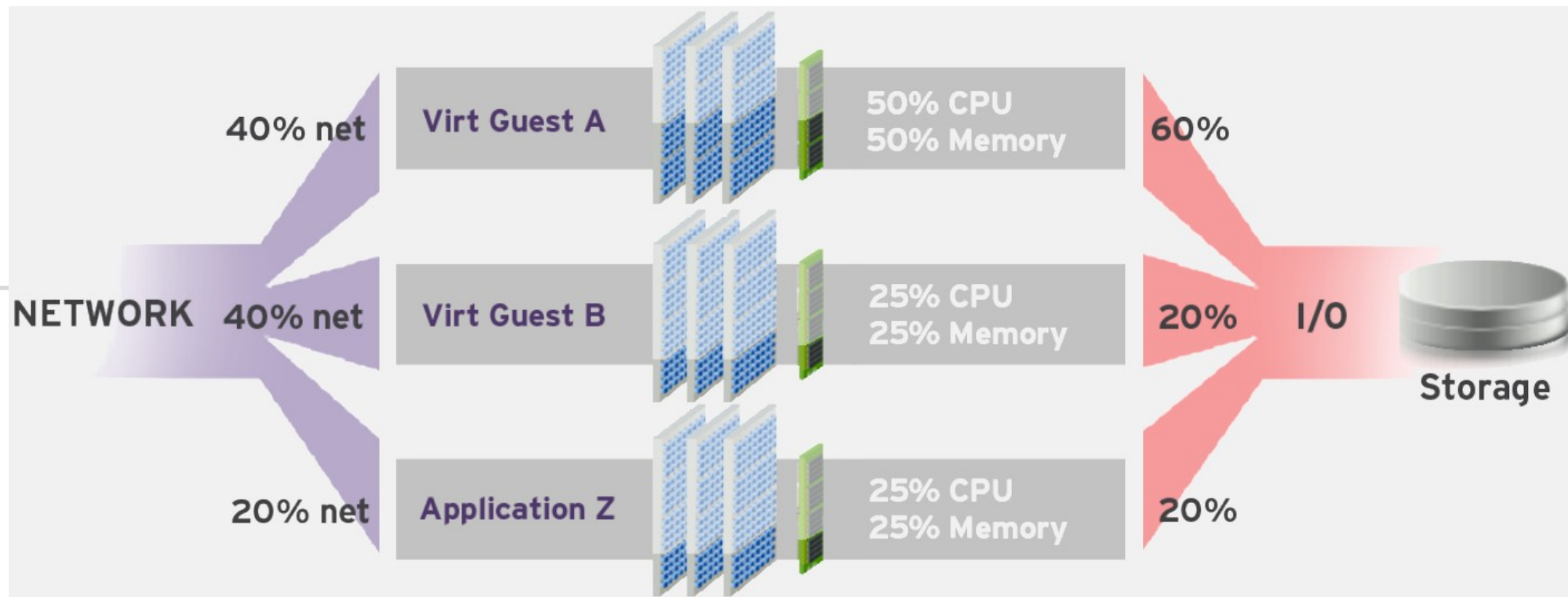
```
$ systemctl  
$ systemctl status  
$ systemctl --state failed
```

- Quel est l'état global du système ? Vert / Rouge
- Combien de services sont en défaut ?

```
$ systemctl --type=service --state=active
```

Gestion des services – systemd

- Control groups
 - Réduire la congestion des accès aux ressources
 - Rendre le comportement système prédictif



Gestion des services – systemd

- Définitions

- **Slice**

- Type d'unité responsable de la création d'une hiérarchie pour la gestion des ressources

- **Scope**

- Unité organisationnelle qui regroupe les processus d'un service.
 - Utilisé pour les sessions utilisateur, les machines virtuelles, les conteneurs, etc.

- **Service**

- Processus ou groupe de processus contrôlés par systemd

- Illustrations

```
$ systemctl status  
$ systemd-cgtop
```

Gestion des services - systemd

- Applications → contrôle de l'état du service **apache**
 - Installer le paquet `task-web-server`

```
$ systemctl status apache2  
$ systemctl stop
```

- Activer le contrôle des ressources pour `apache2`
 - Relever les paramètres de comptabilisation des ressources du service `apache2`
 - Changer ces paramètres de façon à activer la comptabilisation

```
$ systemctl show apache2 | grep -i accounting  
$ sudo systemctl set-property apache2.service IOAccounting=yes  
$ sudo systemctl daemon-reload  
$ sudo systemctl restart apache2
```

- Relancer le service et vérifier son état avec **systemd-cgtop**

```
$ sudo systemd-cgtop
```

Gestion des services - systemd

- Retour sur la journalisation → **journalctl**
 - Pratique pour le dépannage des services
 - Rotation des journaux directement intégrée
 - Collecte des métadonnées en plus du message

```
$ journalctl -xuf apache2  
$ systemctl --type=service --state=active | grep apache2
```

- Journalisation réseau
 - Destination et source → paquet **systemd-journal-remote**

Gestion des services - systemd

- Applications
 - Quel est l'état de chacun des services suivants ?
 - systemd-networkd
 - systemd-resolved
 - systemd-timesyncd
 - Tester le fonctionnement de ces services
- Utiliser le support : [Configuration réseau 100% systemd](#)

Bilan séance

- Initialisation du système
 - Gestionnaire d'amorce
 - GRUB2
 - Noyau Linux
 - Contrôle de l'empreinte mémoire
- Modules du noyau
 - 2 espaces mémoire
 - kernel space → sysfs
 - user space → udev
- Démons & Services
 - Gestion autonome de chaque unité avec systemd

BIOS

Basic Input Output System
→ recherche MBR

MBR

Master Boot Record
→ recherche GRUB

GRUB

Grand Unified Bootloader
→ recherche noyau

kernel

Noyau Linux
→ identification des ressources

systemd

→ exécution /sbin/init
→ lancement des services