Introduction aux systèmes GNU/Linux

S20E06 inetdoc.net

UPSSITECH
ÉCOLE D'INGÉNIEURS

Philippe Latu / Université Toulouse 3

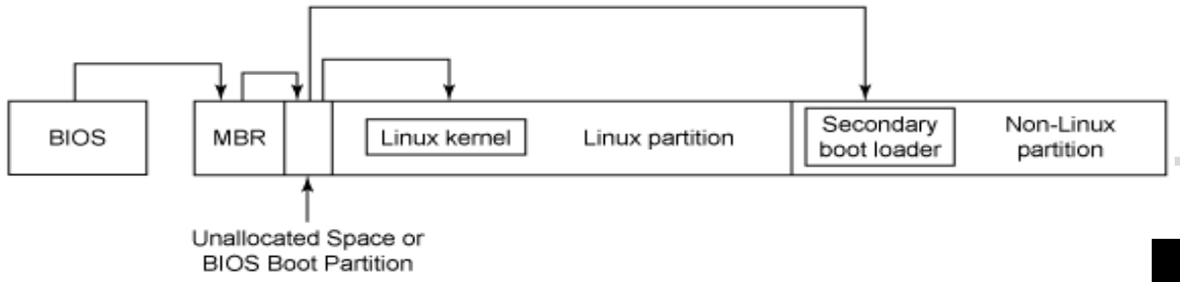
Plan séance 6

- Séance 6 – 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 /dev/sdb
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sdb      8:16   0 111,8G  0 disk
├─sdb1   8:17   0   512M  0 part /boot/efi
├─sdb2   8:18   0  95,4G  0 part /
└─sdb3   8:19   0  15,9G  0 part [SWAP]
```

```
$ ls -1X /boot
efi
grub
config-5.4.0-2-amd64
initrd.img-5.4.0-2-amd64
System.map-5.4.0-2-amd64
vmlinuz-5.4.0-2-amd64
```

Gestionnaire d'amorce

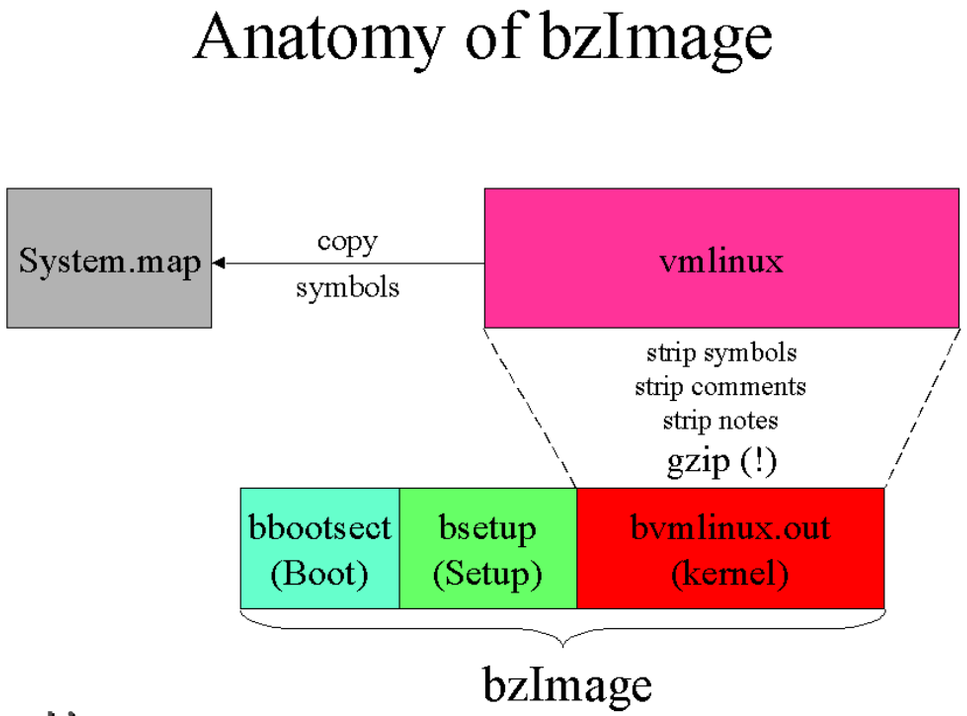
Partie monolithique du noyau

```
$ ls -1X /boot/grub/x86_64-efi/part*
/boot/grub/x86_64-efi/partmap.lst
/boot/grub/x86_64-efi/parttool.lst
/boot/grub/x86_64-efi/part_acorn.mod
/boot/grub/x86_64-efi/part_amiga.mod
/boot/grub/x86_64-efi/part_apple.mod
/boot/grub/x86_64-efi/part_bsd.mod
/boot/grub/x86_64-efi/part_dfly.mod
/boot/grub/x86_64-efi/part_dvh.mod
/boot/grub/x86_64-efi/part_gpt.mod
/boot/grub/x86_64-efi/part_msdos.mod
/boot/grub/x86_64-efi/part_plan.mod
/boot/grub/x86_64-efi/part_sun.mod
/boot/grub/x86_64-efi/part_sunpc.mod
/boot/grub/x86_64-efi/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 discontigues



Source wikipedia

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

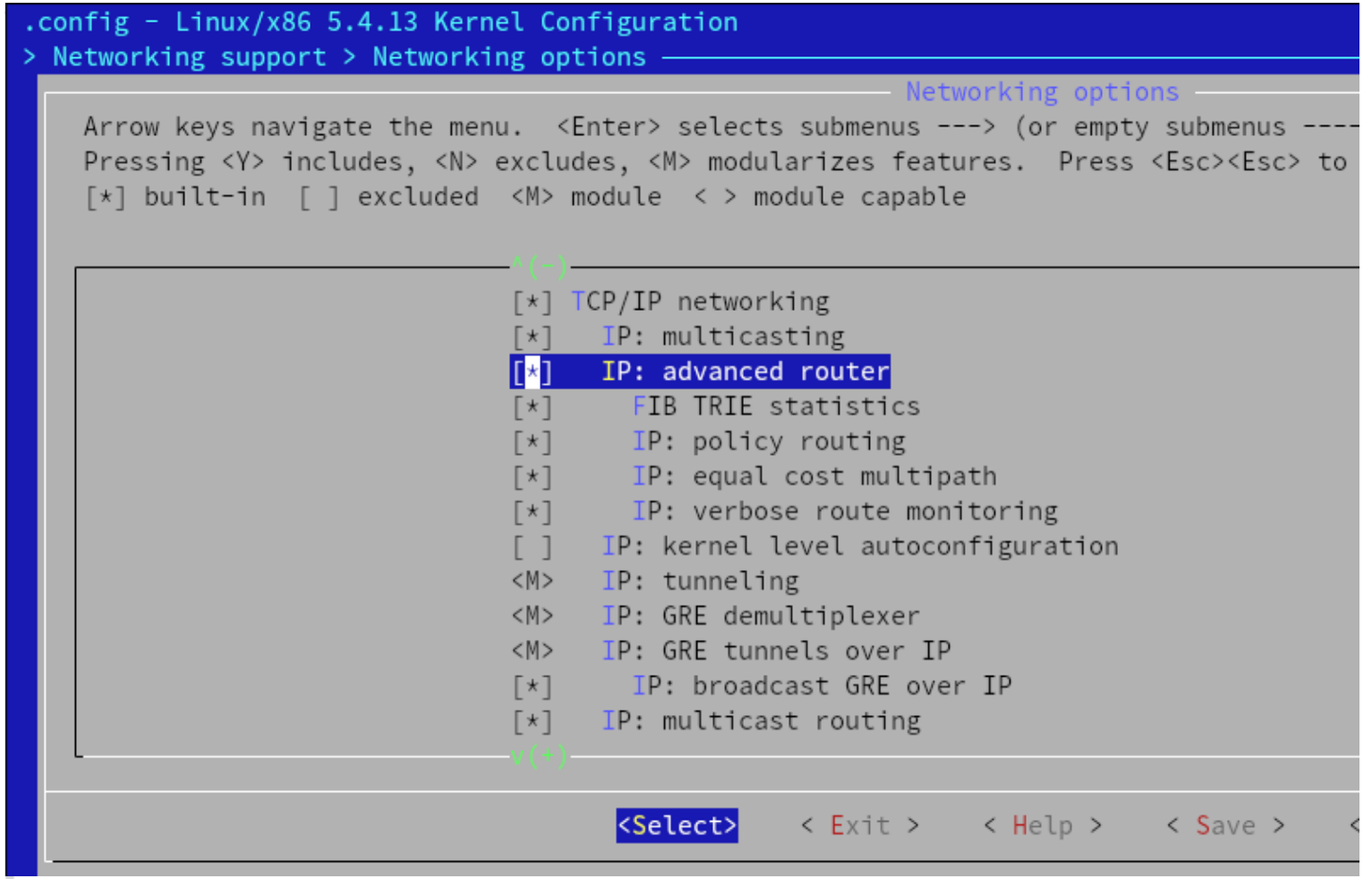
- Prérequis
  - L'utilisateur etu doit appartenir au groupe système src
  - L'arborescence /usr/src appartient au groupe src
  - Le masque des permissions du répertoire /usr/src est rwxrwsr-x
  - Les paquets à installer sont fakeroot & libncurses-dev

```
$ 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

$ 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
```

Reprise de la configuration du noyau de la distribution par défaut

Compilation d'un nouveau noyau

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



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 la configuration du service rsyslogd
  - 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 ?

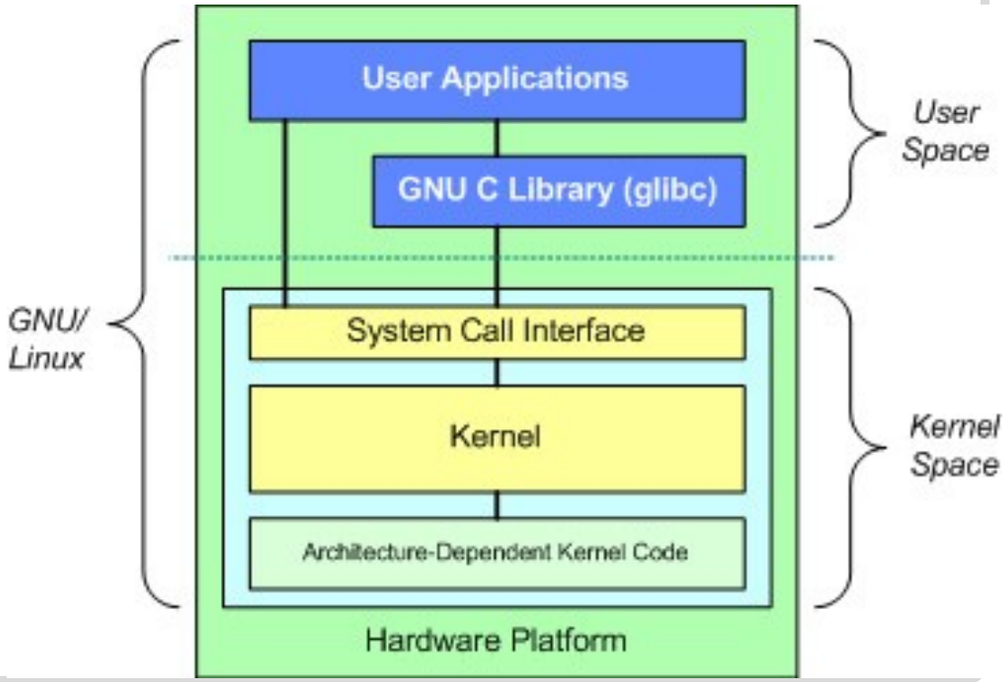
```
$ sudo dmesg -T
```

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
    - Kernelspace → 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 storage
00:06.0 SCSI storage controller: Red Hat, Inc Virtio block device
```

  - Informations collectées par udev

```
# udevadm info --query=all --name=/dev/vda
P: /devices/pci0000:00/0000:00:06.0/virtio2/block/vda
N: vda
S: disk/by-path/pci-0000:00:06.0-virtio-pci-virtio2
E: DEVLINKS=/dev/disk/by-path/pci-0000:00:06.0-virtio-pci-virtio2
E: DEVNAME=/dev/vda
E: DEVPATH=/devices/pci0000:00/0000:00:06.0/virtio2/block/vda
E: DEVTYPE=disk
E: ID_PART_TABLE_TYPE=dos
E: ID_PATH=pci-0000:00:06.0-virtio-pci-virtio2
E: ID_PATH_TAG=pci-0000_00_06_0-virtio-pci-virtio2
...
```

Représentation des périphériques

- Démon udev et interface réseau
  - Informations sur le matériel

```
$ lspci | grep -i ethernet
00:03.0 Ethernet controller: Red Hat, Inc Virtio network device
```

  - Informations collectées par udev

```
$ udevadm info --query=all --path /sys/class/net/eth0
P: /devices/pci0000:00/0000:00:06.0/virtio3/net/eth0
E: DEVPATH=/devices/pci0000:00/0000:00:06.0/virtio3/net/eth0
E: ID_BUS=pci
E: ID_MODEL_FROM_DATABASE=Virtio network device
E: ID_MODEL_ID=0x1000
E: ID_NET_DRIVER=virtio_net
E: ID_NET_LINK_FILE=/etc/systemd/network/50-virtio-kernel-names.link
E: ID_NET_NAME_MAC=enxbaadcafe0000
E: ID_NET_NAME_PATH=enp0s6
E: ID_PATH=pci-0000:00:06.0
...
```

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

Démarrage des services – systemd

- Initialisation des processus
  - 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

Démarrage 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 ?

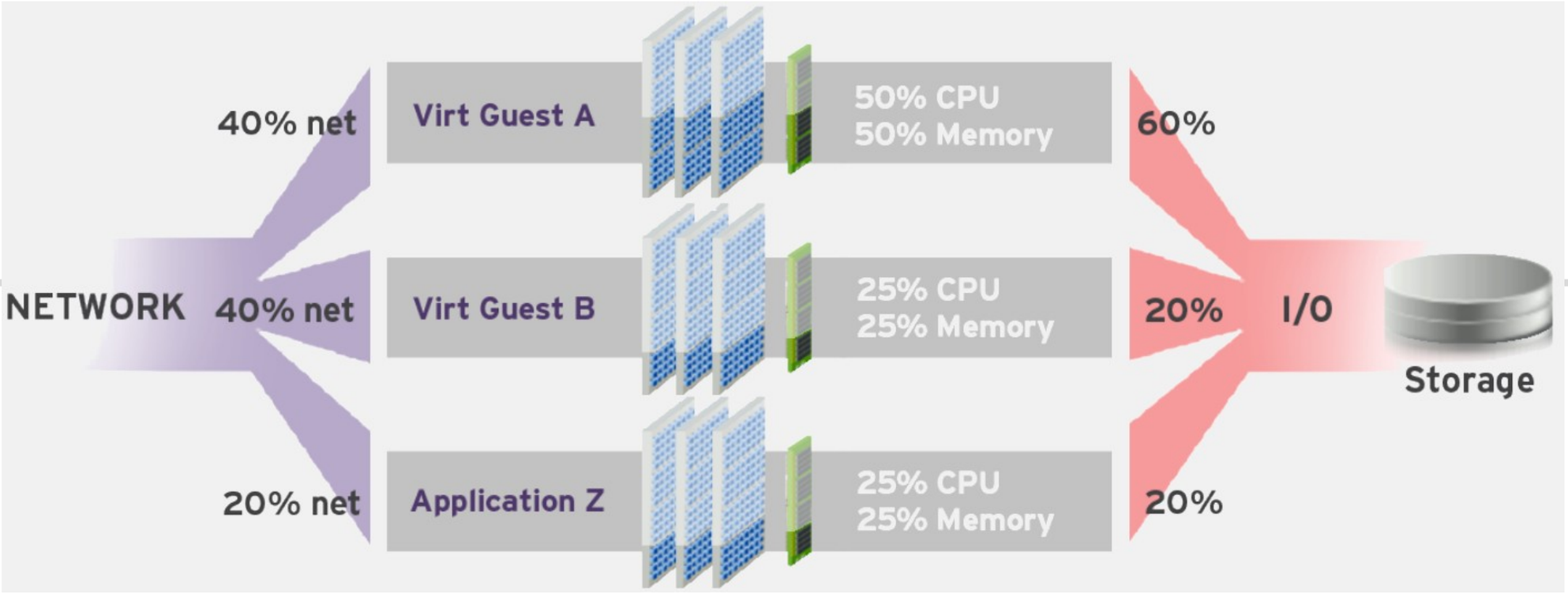
```
$ systemctl --type=service --state=active
```

- Quel est l'état de l'unité systemd-resolved ?
- Comment activer le service ?
- Quelle est la modification à faire sur /etc/nsswitch.conf ?

https://www.freedesktop.org/software/systemd/man/nss-resolve.html

Démarrage des services – systemd

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



Démarrage 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
```

Démarrage 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
$ 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
```

Démarrage des services - systemd

- Retour sur la journalisation → **journalctl**
  - Limité à l'échelle du seul système → pas de fonction réseau
  - Intégration possible dans rsyslog
  - Pratique pour le dépannage des services
  - Rotation des journaux directement intégrée
  - Collecte des métadonnées en plus du message

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

Démarrage des services - systemd

- Applications
  - Quelle est l’unité responsable de la synchronisation horaire ?
  - Comment consulter le journal de cette unité ?
  - Quelle source de temps a été utilisée ?
  - À quel groupe système un utilisateur normal doit-il appartenir pour accéder aux journaux ?
  - Comment consulter le journal de l’activité de l’utilisateur ?

Bilan séance 6

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

BIOS	Basic Input Ouput 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