

**A d m i n i s t r a t i o n**

**S y s t è m e**

**G N U / L i n u x**

# Chapitre 3. Démarrage

## 1. Démarrage du système

1.1. Démarrage de l'ordinateur

1.2. Le chargeur GRUB

1.3. Démarrage du noyau

1.4. Le processus init

1.5. lancement des scripts de démarrage

1.6. Arrêt du système

## 2. Démarrage du système sous CentOS

2.1. Le processus de démarrage

2.2. Systemd

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
  - mise sous tension de l'ordinateur ;
  - lecture **BIOS/UEFI** stocké en mémoire morte ;
  - BIOS effectue un autotest : **POST** ;
  - lecture des paramètres (périphériques de boot, ...) ;
  - lancement du chargeur de démarrage trouvé.
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus **init** ;
- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
  - Le chargeur de démarrage **localise** le **noyau du système d'exploitation** sur le disque, le **charge** et **l'exécute**.
    - La majorité des chargeurs sont interactifs, ils permettent :
      - de **spécifier un noyau** ;
      - de positionner des **paramètres optionnels**.
        - **Types** :
          - **Grub** : GRand Unified Bootloader (le plus répandu) ;
          - **Lilo** : Linux LOader (délaissé par les développeurs) ;
          - **Elilo** : pour UEFI.
  - démarrage du noyau ;
  - lancement du processus **init** ;
  - lancement des scripts de démarrage.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
  - GRUB est un programme de **multiboot** permettant de choisir entre plusieurs O.S lors du démarrage.
  - GRUB fournit une interface avec menu
  - **/boot/grub/ grub.conf.**, ou bien via **/etc/grub.conf** = lien symbolique

*Pour la suite du cours nous utiliserons GRUB.*

---> [GRUB](#)

- démarrage du noyau ;
- lancement du processus **init** ;
- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;

Au démarrage du système, GRUB apparaît:

**[ENTREE]** active la configuration par défaut.

**Une autre touche** fait apparaître le menu du GRUB.

Niveaux de démarrage: *Les 6 niveaux de démarrage :*

**s ou single :**

Le processus **init** démarre le système en mode mono-utilisateur.

Par défaut l'utilisateur est connecté en tant que **root** sans fournir de mot de passe.

**1 - 5 :** Le processus **init** démarre le système avec le niveau demande.

- lancement du processus **init** ;
- lancement des scripts de démarrage.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;

### Etapas du démarrage:

- chargement du noyau (processus 0) ;
- installation des périphériques via leur pilote ;
- démarrage du gestionnaire de swap ;
- montage du système de fichiers racine ;
- création par le noyau du premier processus qui porte le numéro

1 ;

passant                      • ce processus exécute le programme **/sbin/init** en lui  
noyau.                      les paramètres qui ne sont pas déjà gérés par le

- lancement du processus **init** ;
- lancement des scripts de démarrage.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- **lancement du processus init :**

## Les différents niveaux d'exécution :

0	Arrête le système.
1	Mode mono-utilisateur (console).
2	Mode multi-utilisateurs. Les systèmes de fichiers sont montés. Le service réseau est démarré.
3	Sur-ensemble du niveau 2. Il est associé au démarrage des services de partage à distance.
4	Mode multi-utilisateurs spécifique au site informatique.
5	Sur-ensemble du niveau 3. Interface X-Window (graphique).
6	Redémarre le système.
s, S, single	Mode mono-utilisateur (single). Les systèmes de fichiers sont montés. Seuls les processus fondamentaux pour le bon fonctionnement du système sont activés. Un shell en mode <b>root</b> est activé sur une console. Le répertoire <b>/etc</b> n'est pas indispensable.

Il n'y a qu'un niveau d'exécution actif à la fois.

- lancement des scripts de démarrage.



# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- **lancement du processus init :**

### **La commande init**

La commande init permet de changer le niveau d'exécution courant .

*Syntaxe de la commande init :*

init [-options] [0123456Ss]

Exemple :

[root]# init 5

### **La commande runlevel**

La commande runlevel permet de connaître le niveau d'exécution courant.

- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init :

**La commande init**

**La commande runlevel**

Exemple :

```
[root]# runlevel
```

```
N 3
```

→ le système se trouve au niveau d'exécution 3 - Multiuser.

→ Le N : le niveau d'exécution précédent = démarrage du système.      Après un init 5: → [root]# runlevel

```
3 5
```

- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init :

La commande **init**

La commande **runlevel**

Le fichier **/etc/inittab** :

Lors du démarrage, à la création du processus **init**, le niveau est celui défini dans **GRUB** sinon celui dans **/etc/inittab**.

Le niveau défini dans **GRUB**.

- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;

## • lancement du processus init : Le fichier /etc/inittab :

# For information on how to write upstart event handlers, or how  
# upstart works, see init(5), init(8), and initctl(8).

# Default runlevel. The runlevels used are:

# 0 - halt (Do NOT set initdefault to this)

# 1 - Single user mode

# 2 - Multiuser, without NFS (The same as 3, if you do not have  
networking)

# 3 - Full multiuser mode

# 4 – unused

# 5 - X11

# 6 - reboot (Do NOT set initdefault to this)

id:5:initdefault:

• lancement des scripts de démarrage

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- **lancement du processus init :**

## Démarrage des démons :

**init** lance le script **/etc/rc.d/rc.sysinit** quelque soit le niveau.

**Init** exécute ensuite le script **/etc/rc.d/rc** en lui passant en paramètre le niveau d'exécution demandé

- lancement des scripts de démarrage.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init ;
- **lancement des scripts de démarrage:**

## Script de démarrage des services

Pour chaque service, il y a un script de démarrage stocké dans **/etc/rc.d/init.d**.

Chaque script accepte au minimum en argument :

- stop : pour arrêter le service ;
  - start : pour démarrer le service ;
  - restart : pour redémarrer le service ;
  - status : pour connaître l'état du service.
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init ;
- **lancement des scripts de démarrage:**

## Répertoires d'ordonnancement :

Pour chaque niveau d'exécution, il existe un répertoire correspondant : **/etc/rc.d/rc[0-6].d/**

Ces répertoires contiennent les liens symboliques vers les scripts placés dans **/etc/rc.d/init.d**

- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init ;
- **lancement des scripts de démarrage:**

Nom des liens :

Mise en route (Start) : SXXnom

Arrêt (Kill) : KYYnom

XX et YY : nombre de 00 a 99 qui guide l'ordre d'exécution (Start ou Kill).

**nom** : nom exact du service à démarrer ou à arrêter tel qu'écrit dans /etc/rc.d/init.d/.

La somme des nombres est un complément a 100 :  $XX + YY = 100$ .

Cette méthode permet d'ordonnancer le démarrage et l'arrêt des services. Un service qui est démarre en premier doit être le dernier a s'arrêter. La liste étant lue dans l'ordre alphabétique.



# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init ;
- **lancement des scripts de démarrage:**

Nom des liens :

Exemple :

Dans /etc/rc.d/rc3.d/, nous avons :

- K15httpd
- S10network
- S26acpid

Donc, pour le niveau d'exécution 3 (rc3.d) :

- le service httpd doit être arrêté (lettre K),
- les services network et acpid doivent être lancés (lettre S) dans cet ordre (numéro du service network 10 plus petit que celui de acpid 26)

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init ;
- **lancement des scripts de démarrage:**

### Le programme `/etc/rc.d/rc` :

Ce programme est lancé par **init** avec le niveau d'exécution en paramètre.

**Il comporte deux boucles.**

**Init** lance le script **rc** avec le niveau **X** en paramètre.

Première boucle : lecture des scripts d'arrêt **K...** présents dans **rcX.d**.

Deuxième boucle : lecture des scripts de démarrage **S...** présents dans **rcX.d**.

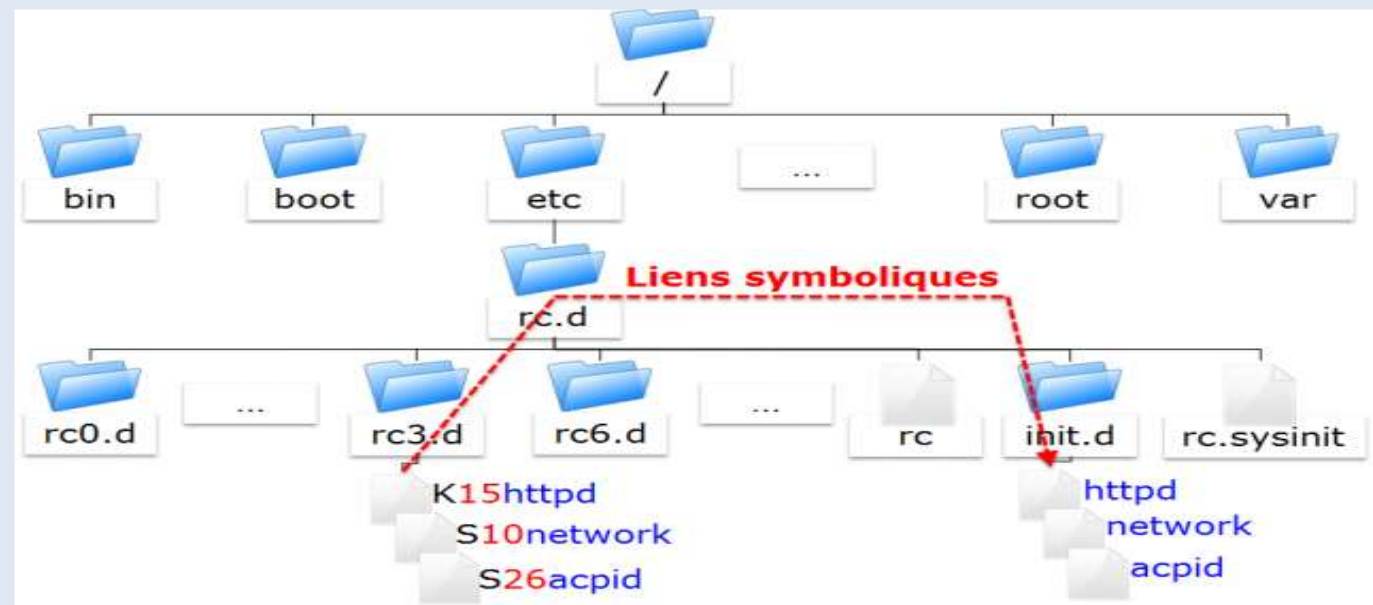
- Arrêt du système.

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init :
- **lancement des scripts de démarrage :**

## Architecture de démarrage :



Suite →

# GRUB

## *Les touches de selection du menu de Grub*

Touche	Action
[ENTREE]	Démarre le système sélectionné.
[e]	Édite la configuration du système.
[a]	Permet de modifier les arguments.
[c]	Permet d'utiliser l'interface Shell de GRUB.

**[ENTREE]** : lancer l'initialisation du serveur.  
affichage des messages places dans **/etc/issue** et login & password

**[e]** : éditer la configuration avant de démarrer.  
modifier les lignes → **[e]**.  
Pour initialiser le système avec modifications → **[b]**

('boot').

**[a]** / **[q]** : permet de modifier les arguments du noyau.

**[c]** : permet d'obtenir l'interface Shell de Grub.

# GRUB

**/boot/grub/grub.conf :**



```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
Hiddenmenu
title CentOS (2.6.32-220.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-220.el6.i686 ro
        root=LABEL=/ rhgb quiet
    initrd  /initramfs-2.6.32-220.el6.i686.img
```

# GRUB

## /boot/grub/grub.conf :

### Variables du fichier /boot/grub/grub.conf :

Variable	Observation
default=0	Correspond au système d'exploitation lancé par défaut. La première rubrique <b>title</b> porte le numéro 0.
timeout=5	GRUB amorcera automatiquement le système par défaut au bout de 5 secondes, à moins d'être interrompu.
splashimage	<p>Déclaration de l'image qui s'affiche avec le chargeur Grub. Il faut indiquer l'emplacement de cette image.</p> <p>Les systèmes de fichiers n'étant pas encore montés, indiquer le disque et la partition de /boot (<b>hd0,0</b>), le chemin jusqu'à l'image <b>grub/</b> et enfin le nom de votre image <b>splash.xpm.gz</b>.</p>
hiddenmenu	Sert à masquer le menu Grub et après le délai du timeout, le système se lancera automatiquement en fonction de l'option <b>default</b> .
title	<p>Il s'agit en fait du nom qui apparaîtra dans le menu Grub (exemple "ma distrib Linux préférée"). En règle générale, le nom du système est choisi : exemple Fedora, Suse, Ubuntu, Vista, Xp, Frugal, etc,... et éventuellement la version du noyau.</p> <p>Un seul nom par rubrique <b>title</b>, il faut donc déclarer autant de lignes 'title' qu'il y a d'options de démarrage ou de systèmes installés.</p>
root	Indique le disque puis la partition ( <b>hdx,y</b> ) où se trouvent les fichiers permettant l'initialisation du système (exemple : ( <b>hd0,0</b> ), correspondant à la partition /boot) pour ce "title".



```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
Hiddenmenu
title CentOS (2.6.32-220.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-220.el6.i686 ro
    root=LABEL=/ rhgb quiet
    initrd /initramfs-2.6.32-220.el6.i686.img
```

# GRUB

## /boot/grub/grub.conf :

Variable	Observation
kernel	Indique le nom du noyau à charger, son emplacement et les options utiles à son démarrage pour ce "title".
initrd	initrd (INITial RamDisk) charge un ramdisk initial pour une image de démarrage au format Linux et définit les paramètres adéquats dans la zone de configuration de Linux en mémoire pour ce "title".



```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
Hiddenmenu
title CentOS (2.6.32-220.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-220.el6.i686 ro
    root=LABEL=/ rhgb quiet
    initrd /initramfs-2.6.32-220.el6.i686.img
```

---> Retour  
Démarrage

# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init :

## Arrêt du système:

Les opérations de maintenance, diagnostics, modifications de logiciels, ajouts et retraits de matériel, tâches administratives, coupures électriques, ....

Cet arrêt peut être planifié, périodique ou impromptu et demander une réactivité immédiate.

Ceci garantit **l'intégrité** du disque et la **terminaison propre** des différents services du système.

## Arrêt programmé du système :

- utilisateurs prévenus de l'arrêt ;
- applications arrêtées proprement ;
- intégrité des systèmes de fichiers assurée ;
- sessions utilisateurs stoppées.



# Chapitre 3. Démarrage

## Séquence de démarrage :

- démarrage de l'ordinateur ou amorçage ;
- exécution du chargeur de démarrage ;
- démarrage du noyau ;
- lancement du processus init :

## Arrêt du système:

En fonction des options, le système :

- passe en mode mono-utilisateur ;
- est arrêté ;
- est redémarré.

Commandes de mise hors service :

- init ;
- shutdown ;
- halt ;
- reboot.

# Chapitre 3. Démarrage

## Arrêt du système:

### Commande shutdown

La commande shutdown éteint le système.

Syntaxe de la commande shutdown :

**shutdown [-t sec] [options] heure [message-avertissement]**

### Exemples :

```
[root]# shutdown -r +2          "arrêt puis reboot dans 2 minutes"
[root]# shutdown -r 10:30       "arrêt puis reboot à 10h30"
[root]# shutdown -h now         "arrêt électrique"
```

### Options:

-t sec  
processus

-r

-h

-P

-f

-F

-c

### Commentaires:

Attendre **sec** entre le message d'avertissement et le signal de fin aux

Redémarrer la machine après l'arrêt du système

Arrêter la machine après l'arrêt du système

Eteindre alimentation

Ne pas effectuer de fsck en cas de redémarrage

Forcer l'utilisation de fsck en cas de redémarrage

Annuler un redémarrage en cours

# Chapitre 3. Démarrage

## Arrêt du système:

### Commande shutdown:

La commande shutdown éteint le système.

*Syntaxe de la commande shutdown :*

**shutdown [-t sec] [options] heure [message-avertissement]**

#### **Options:**

**heure**

**message-avertissement**

#### **Commentaires:**

Quand effectuer le shutdown (soit une heure fixe **hh:mm**, soit un délai d'attente en minute **+mm**).

Message à envoyer à tous les utilisateurs.

### **Commande halt :**

La commande halt provoque un arrêt immédiat du système.

**[root]# halt**

Cette commande appelle le processus **init**

**halt ⇒ init 0**

### **Commande reboot :**

La commande reboot provoque un redemarrage immédiat du système.

**[root]# reboot**

Cette commande appelle le processus **init**

**reboot ⇒ init 6**

# Chapitre 3. Démarrage

## Démarrage de CentOS:

### Le démarrage du BIOS

→ **POST** (power on self test) pour détecter, tester et initialiser les composants matériels du système.

→ Il charge ensuite le **MBR** (Master Boot Record).

### Le Master boot record (MBR)

= 512 premiers bytes du disque de démarrage. Le MBR

→ charge **GRUB2** en mémoire et lui transfère le contrôle.

Les 64 bytes suivants contiennent la table de partition du disque.

### Le chargeur de démarrage GRUB2 (Bootloader)

Le chargeur de démarrage par défaut de la distribution CentOS 7 est **GRUB2** (GRand Unified Bootloader). GRUB2 remplace l'ancien chargeur de démarrage Grub (appelé également GRUB legacy).

Le fichier de configuration de GRUB 2 se situe sous **/boot/grub2/grub.cfg** ne doit pas être édité.

Les paramètres de config. de GRUB2 se trouvent sous **/etc/default/grub** → gén. de grub.cfg.

```
[root] # grub2-mkconfig -o /boot/grub2/grub.cfg
```

- GRUB2 cherche l'image du noyau compressé (le fichier `vmlinuz`) dans le répertoire `/boot`.
- GRUB2 charge l'image du noyau en mémoire et extrait le contenu du fichier image `initramfs` dans un dossier temporaire en mémoire en utilisant le système de fichiers `tmpfs`.

# Chapitre 3. Démarrage

## Démarrage de CentOS:

### Le noyau

Le noyau démarre le processus **systemd** avec le PID 1.

### systemd

Systemd est le père de tous les processus du système. Il lit la cible du lien **/etc/systemd/system/default.target** (par exemple **/usr/lib/systemd/system/multi-user.target**) pour déterminer la cible par défaut du système. Le fichier définit les services à démarrer.

**Systemd** positionne ensuite le système dans l'état défini par la cible en effectuant les tâches d'initialisations suivantes :

1. Paramétrer le nom de machine
2. Initialiser le réseau
3. Initialiser SELinux
4. Afficher la bannière de bienvenue
5. Initialiser le matériel en se basant sur les arguments fournis au kernel lors du démarrage
6. Monter les systèmes de fichiers, en incluant les systèmes de fichiers virtuels comme **/proc**
7. Nettoyer les répertoires dans **/var**
8. Démarrer la mémoire virtuelle (swap)

# Chapitre 3. Démarrage /Systemd :

**systemd** est le système d'initialisation adopté par toutes les dernières distributions Linux. Tout comme Upstart, il est basé sur des **événements**, ce qui rend son fonctionnement souple et dynamique. Il permet un démarrage en parallèle des processus avec une gestion très fine des dépendances. Il peut même redémarrer un processus qui se serait arrêté par erreur, gérer la planification des tâches, les logs système, les périphériques et plus encore.

## Gérez le démarrage et l'arrêt de vos processus :

**systemd** est fourni avec la commande **systemctl** qui permet de gérer les “unités” (units, en anglais). Pour systemd, unité est un terme générique qui désigne tout objet sur lequel il va pouvoir agir. Les unités peuvent être de différents types : **service**, **montage**, **périphérique**, etc.

Les daemons sont de type “service”.

Par exemple

```
sudo systemctl start smbd
```

```
sudo systemctl stop smbd
```

# Chapitre 3. Démarrage /Systemd :

D'autres actions possibles sont :

- **restart** pour stop puis start en une seule commande ;
- **reload** pour mettre à jour la configuration sans couper le service (pour les services qui en sont capables)
- **status** pour connaître l'état de votre service ;
- **enable** pour lancer le service au démarrage ;
- **disable** pour ne pas lancer le service au démarrage.

Si vous lancez la commande **systemctl** sans argument, ça équivaut à lancer la commande suivante qui liste toutes les unités "actives" :

```
systemctl list-units
```

filtrer cette liste pour n'afficher que les unités de type "service" :

```
systemctl list-units --type=service
```

# Chapitre 3. Démarrage /Systemd :

Vous voyez que toutes les unités se terminent par “**.service**”. C’est parce que la configuration de chaque unité est définie dans un fichier dont le nom est du type **NOM\_DE\_L\_UNITE.TYPE\_DE\_L\_UNITE**.

Presque tous les fichiers des unités sont dans **/lib/systemd/system**

pas besoin de connaître l’emplacement réel des fichiers car la commande `systemctl` permet de lire et d’éditer les fichiers d’unité.

vous pouvez afficher le contenu du fichier unité **smbd.service** par la commande :

```
systemctl cat smbd.service
```

Cette commande modifier le fichier original, mais créer un fichier de “surcharge” qui aura la priorité sur le fichier original.

L’avantage :revenir à la configuration par défaut en cas de besoin.

éditer directement le fichier de configuration, utilisez la commande :

```
sudo systemctl edit --full smbd.service
```

→ faire prendre en compte vos changements au système par la commande :

```
sudo systemctl daemon-reload
```



# Chapitre 3. Démarrage /Systemd :

## Utilisez les targets, évolution des runlevels de systemV :

Vous pouvez afficher la liste des unités de type “target” par la commande :

```
sudo systemctl list-units --type target --all
```

Le terme anglais “target” signifie “cible”. Il permet d’amener le système à un certain état cohérent. À la différence des **runlevels**, vous voyez qu’il peut y avoir plusieurs **targets** actives en même temps.

Certaines de ces targets, comme **swap.target** , correspondent plutôt à des fonctions du système qu’on peut activer ou non, et d’autres, comme **graphical.target** , correspondent à des états cohérents du système qui ont pour dépendances les différentes autres targets.

afficher la liste des dépendances d’une unité par la commande :

```
sudo systemctl list-dependencies graphical.target
```

Vous voyez ainsi toutes les targets et autres unités dont dépend **graphical.target** .

# Chapitre 3. Démarrage /Systemd :

Vous pouvez voir la target par défaut par la commande :

```
sudo systemctl get-default
```

Vous pouvez changer de target par la sous-commande `isolate` . Par exemple, la commande suivante permet de passer en mode de secours (sans réseau, seulement root connecté) :

```
sudo systemctl isolate rescue.target
```

Vous pouvez fixer `graphical.target` comme target par défaut, par la commande :

```
sudo systemctl set-default graphical.target
```

# Chapitre 3. Démarrage /Systemd :

## Gérez vos logs avec journalctl :

Par défaut, la commande **journalctl** affiche tous les logs gérés par **systemd** par ordre chronologique :

```
sudo journalctl
```

Vous pouvez choisir de n'afficher les logs que depuis le dernier démarrage du système, par la commande :

```
sudo journalctl -b
```

ou n'afficher que les logs du kernel depuis le dernier démarrage (équivalent de la commande `dmesg`) par :

```
sudo journalctl -b -k
```

Vous pouvez aussi choisir d'afficher uniquement les logs d'un service (ici **smbd.service**) ou d'une unité en particulier, par la commande :

```
sudo journalctl -u smbd.service
```

**systemd** est un élément central du fonctionnement de toutes les dernières distributions Linux, et il est probable qu'il prenne encore plus d'importance dans le futur. Il est donc important d'être familier avec ce système.

# Chapitre 3. Démarrage /Systemd :

## En résumé :

**systemd** gère des unités qui peuvent être de différents types.

La gestion des unités se fait par la commande **systemctl** .

Vous pouvez démarrer vos services par la commande **systemctl start SERVICE\_NAME** .

Les **targets** de **systemd** ont succédé aux **runlevels** de **systemV**.

Vous pouvez gérer vos logs système par la commande **journalctl** .