

Version : **2024.01**

Dernière mise-à-jour : 2021/06/19 11:36

LCF510 - Gestion du Démarrage et de l'Arrêt du Système

Contenu du Module

- **LCF510 - Gestion du Démarrage et de l'Arrêt du Système**
 - Contenu du Module
 - Présentation
 - BIOS, EFI et OpenFirmware
 - Systèmes à base du BIOS
 - Charger de Démarrage
 - Systèmes à base de l'EFI
 - Autres Systèmes
 - Gestionnaire d'amorçage
 - GRUB 2
 - Le fichier /boot/grub/device.map
 - Le fichier /etc/default/grub
 - Les fichiers du répertoire /etc/grub.d
 - Configurer l'Authentification
 - Modifier la Configuration de GRUB 2 en Ligne de Commande
 - Chargeurs de Démarrages Alternatifs
 - Systemd-boot
 - U-boot
 - Le Projet Syslinux
 - SYSLINUX
 - EXTLINUX
 - ISOLINUX
 - PXELINUX

- Isodhpx
- Initramfs
 - Examiner l'image existante
 - Le script init
 - Consulter le contenu d'un fichier dans initramfs
- Processus de Démarrage du Noyau Linux
- Processus Init
- Systemd
 - LAB #1 - La Commande systemctl
 - LAB #2 - Fichiers de Configuration
 - 2.1 - Fichiers de Configuration par Défaut
 - 2.2 - Surchargement des Fichiers de Configuration par Défaut
 - LAB #3 - La Commande systemd-analyze
 - LAB #4 - Les Cibles Systemd
 - 4.1 - Contrôler les dépendances d'une Cible
 - 4.2 - La Cible par Défaut
 - LAB #5 - Gestion des Services
 - 5.1 - Gestion des Instances Uniques
 - 5.2 - Gestion d'Instances Multiples
 - 5.3 - Interdire la Modification du Statut d'un Service
- Arrêt Système du Système
 - La Commande shutdown
 - La Commande reboot
 - La Commande halt
 - La Commande poweroff

Présentation

Le processus de démarrage de Linux peut être résumé en trois étapes majeurs :

- Le **firmware** ou **micrologiciel** démarre en effectuant un test rapide du matériel, appelé un **Power-On Self Test** ou **POST**, puis recherche le **Charger de Démarrage** (*Bootloader*) à exécuter à partir d'un support bootable,
- Le Charger de Démarrage est exécuté et il détermine quel noyau Linux à charger,

- Le noyau se charge en mémoire et commence à exécuter en arrière plan les programmes nécessaires au fonctionnement du système.



A retenir : Il est possible de consulter le défilement des messages lors du démarrage en appuyant sur la touche **Echap** ou simultanément sur les touches **Ctrl+Alt+F1**. En sachant que la liste des messages se défilent rapidement, il est possible de les consulter **après** le démarrage du système à l'aide de la commande **dmesg** qui lit les derniers messages contenu dans le **Kernel Ring Buffer**. Ces messages sont aussi copiés dans le fichier **/var/log/boot.log**.

Cette description simpliste résume cependant un processus bien plus compliqué que ce cours va détailler.

BIOS, EFI et OpenFirmware

Systèmes à base du BIOS

Au démarrage d'un système à base d'un processeur x86 ou x86-64, le premier programme exécuté a été traditionnellement le BIOS. Le BIOS a pour fonction de :

- Tester les composants et les circuits,
- Faire appel au BIOS de la carte graphique pour initialiser le système d'affichage,
- Détecter les périphériques de stockage,
- Lancer le **Charger de Démarrage** du système d'exploitation en utilisant le **bootstrap loader**.

Chargeur de Démarrage

La première partie du Chargeur de Démarrage est en règle générale placé dans le MBR du disque. Le format du MBR est le suivant :

- 446 octets pour le Chargeur de Démarrage,

- 64 octets pour la table de partitions, soit 16 octets par partition décrite,
- 2 octets ayant une valeur fixe en hexadécimale de **AA55**.

Systèmes à base de l'EFI

Depuis 2011, le BIOS est en train d'être remplacé par l'utilisation de l'**UEFI** (**U**nified **E**xtensible **F**irmware **I**nterface ou *Interface micrologicielle extensible unifiée*) issue du développement de l'EFI conçue par Intel pour les processeurs Itanium..

Sous EFI la première partie du gestionnaire de démarrage est un fichier ayant une extension .efi se trouvant dans un sous-répertoire au nom du système d'exploitation à lancer dans une partition appelée **EFI System Partition** ou **ESP**. Cette partition est normalement montée à **/boot/efi** sous Linux.

Pour que EFI fonctionne, le micrologiciel (**firmware**) d'EFI doit avoir connaissance de chaque système d'exploitation à démarrer.



A retenir : Sous Linux c'est l'application **efibootmgr** qui permet de créer et de supprimer des entrées ainsi que de modifier l'ordre de démarrage.



Important : L'UEFI gère les **SSD** (*Solid State Drives*) qui utilisent le standard **NVMe** (*Non-Volatile Memory Express*). Linux supporte les SSD depuis le noyau 3.3.

Autres Systèmes

Les systèmes utilisant des processeurs autre qu'un x86 ou x86-64 utilisent un logiciel tel **OpenFirmware**.

Gestionnaires de Démarrage

GRUB 2

GRUB 2 est une ré-écriture complète de GRUB Legacy. Il apporte des améliorations, notamment GRUB 2 sait utiliser des partitions RAID et LVM.

Le lancement de GRUB 2 se fait en trois étapes :

- Etape 1 : Le **boot.img**, stocké dans les 512 premiers octets du secteur 0 avec la table des partitions, est lancé. Son seul but est de lancer l'étape 1.5,
- Etape 1.5 : Le **core.img**, d'une taille approximative de 25 Ko et stocké dans les secteurs 1 à 62, est lancé. Son travail est de charger des pilotes qui supportent de multiples systèmes de fichiers puis de lancer l'étape 2 dans un des systèmes de fichiers,
- Etape 2 : Contenu dans le répertoire **/boot/grub2/**, il lance le menu pour que l'utilisateur puisse choisir le système d'exploitation à lancer.

Dans le cas où le Chargeur de Démarrage **GRUB 2** n'est pas installé, il convient de saisir la commande suivante :

```
# grub2-install /dev/périphérique [Entrée]
```

où **périphérique** est le nom du périphérique où l'étape 1 de GRUB2 doit s'installer dans le MBR.

GRUB 2 lit ses entrées de menus à partir du fichier **/boot/grub2/grub.cfg**. Pour visualiser ce fichier, il convient de saisir la commande suivante :

```
[root@centos8 ~]# cat /boot/grub2/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub2-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
set pager=1
```

```
if [ -f ${config_directory}/grubenv ]; then
    load_env -f ${config_directory}/grubenv
elif [ -s $prefix/grubenv ]; then
    load_env
fi
if [ "${next_entry}" ] ; then
    set default="${next_entry}"
    set next_entry=
    save_env next_entry
    set boot_once=true
else
    set default="${saved_entry}"
fi

if [ x"${feature_menuentry_id}" = xy ]; then
    menuentry_id_option="--id"
else
    menuentry_id_option=""
fi

export menuentry_id_option

if [ "${prev_saved_entry}" ]; then
    set saved_entry="${prev_saved_entry}"
    save_env saved_entry
    set prev_saved_entry=
    save_env prev_saved_entry
    set boot_once=true
fi

function savedefault {
    if [ -z "${boot_once}" ]; then
        saved_entry="${chosen}"
        save_env saved_entry
    fi
}
```

```
    fi
}

function load_video {
    if [ x$feature_all_video_module = xy ]; then
        insmod all_video
    else
        insmod efi_gop
        insmod efi_uqa
        insmod ieee1275_fb
        insmod vbe
        insmod vga
        insmod video_bochs
        insmod video_cirrus
    fi
}

terminal_output console
if [ x$feature_timeout_style = xy ] ; then
    set timeout_style=menu
    set timeout=5
# Fallback normal timeout code in case the timeout_style feature is
# unavailable.
else
    set timeout=5
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/00_tuned ###
set tuned_params=""
set tuned_initrd=""
### END /etc/grub.d/00_tuned ###

### BEGIN /etc/grub.d/01_menu_auto_hide ###
```

```
if [ "${boot_success}" = "1" -o "${boot_indeterminate}" = "1" ]; then
    set last_boot_ok=1
else
    set last_boot_ok=0
fi

# Reset boot_indeterminate after a successful boot
if [ "${boot_success}" = "1" ] ; then
    set boot_indeterminate=0
# Avoid boot_indeterminate causing the menu to be hidden more than once
elif [ "${boot_indeterminate}" = "1" ]; then
    set boot_indeterminate=2
fi
set boot_success=0
save_env boot_success boot_indeterminate

if [ x$feature_timeout_style = xy ] ; then
    if [ "${menu_show_once}" ]; then
        unset menu_show_once
        save_env menu_show_once
        set timeout_style=menu
        set timeout=60
    elif [ "${menu_auto_hide}" -a "${last_boot_ok}" = "1" ]; then
        set orig_timeout_style=${timeout_style}
        set orig_timeout=${timeout}
        if [ "${fastboot}" = "1" ]; then
            # timeout_style=menu + timeout=0 avoids the countdown code keypress check
            set timeout_style=menu
            set timeout=0
        else
            set timeout_style=hidden
            set timeout=1
        fi
    fi
fi
```



```
fi
### END /etc/grub.d/01_menu_auto_hide ###

### BEGIN /etc/grub.d/01_users ###
if [ -f ${prefix}/user.cfg ]; then
    source ${prefix}/user.cfg
    if [ -n "${GRUB2_PASSWORD}" ]; then
        set superusers="root"
        export superusers
        password_pbkdf2 root ${GRUB2_PASSWORD}
    fi
fi
### END /etc/grub.d/01_users ###

### BEGIN /etc/grub.d/10_linux ###
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 2ae4c035-9244-458c-82c5-a49ae169cdb6
else
    search --no-floppy --fs-uuid --set=root 2ae4c035-9244-458c-82c5-a49ae169cdb6
fi
insmod part_msdos
insmod ext2
set boot='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=boot --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 2ae4c035-9244-458c-82c5-a49ae169cdb6
else
    search --no-floppy --fs-uuid --set=boot 2ae4c035-9244-458c-82c5-a49ae169cdb6
fi
```

```
# This section was generated by a script. Do not modify the generated file - all changes
# will be lost the next time file is regenerated. Instead edit the BootLoaderSpec files.
#
# The blscfg command parses the BootLoaderSpec files stored in /boot/loader/entries and
# populates the boot menu. Please refer to the Boot Loader Specification documentation
# for the files format: https://www.freedesktop.org/wiki/Specifications/BootLoaderSpec/.

set default_kernels="root=UUID=4c0cc28c-0d59-45be-bd73-d292b80be33c ro crashkernel=auto resume=UUID=c8bb3f47-
d67f-4b21-b781-766899dc83d4 rhgb quiet "

insmod blscfg
blscfg
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_linux_xen ###
### END /etc/grub.d/20_linux_xen ###

### BEGIN /etc/grub.d/20_ppc_terminfo ###
### END /etc/grub.d/20_ppc_terminfo ###

### BEGIN /etc/grub.d/30_os-prober ###
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/30_uefi-firmware ###
### END /etc/grub.d/30_uefi-firmware ###

### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
### END /etc/grub.d/40_custom ###

### BEGIN /etc/grub.d/41_custom ###
if [ -f ${config_directory}/custom.cfg ]; then
```

```
source ${config_directory}/custom.cfg
elif [ -z "${config_directory}" -a -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
fi
### END /etc/grub.d/41_custom ###
```

Notez que ce fichier ne doit pas être modifié manuellement. En effet, il est généré par la commande **grub2-mkconfig**. La commande grub2-mkconfig prend en argument l'emplacement du fichier grub.cfg, par exemple :

- grub2-mkconfig -o /boot/grub2/grub.cfg, ou
- grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg

Lors de l'exécution de la commande **grub2-mkconfig** plusieurs fichiers sont lus :

Le fichier /boot/grub2/device.map

```
[root@centos8 ~]# cat /boot/grub2/device.map
# this device map was generated by anaconda
(hd0)      /dev/sda
(hd1)      /dev/sda
```

Le fichier /etc/default/grub

Ce fichier contient la configuration par défaut des paramètres de GRUB 2 :

```
[root@centos8 ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
```

```
GRUB_CMDLINE_LINUX="crashkernel=auto resume=UUID=c8bb3f47-d67f-4b21-b781-766899dc83d4 rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```



Important : Notez que toute modification de ce fichier nécessite l'exécution de la commande **grub2-mkconfig** pour que les modifications soient prises en compte.

Dans ce fichier les directives sont :

Directive	Description
GRUB_TIMEOUT	Durée de l'affichage du menu avant le démarrage en utilisant la valeur de GRUB_DEFAULT
GRUB_DISTRIBUTOR	Ligne de commande qui génère le texte de l'entrée
GRUB_DEFAULT	Entrée du menu sélectionner par défaut
GRUB_DISABLE_SUBMENU	Active ou désactive les sous-menus de GRUB2
GRUB_TERMINAL_OUTPUT	Spécifie le terminal par défaut
GRUB_CMDLINE_LINUX	Paramètres passés au noyau peu importe le type de démarrage
GRUB_DISABLE_RECOVERY	Active ou désactive la génération des entrées en mode recovery
GRUB_BLSCFG	Active ou désactive l'utilisation de la nouvelle spécification du gestionnaire de démarrage pour configurer celui-ci

Les fichiers du répertoire /etc/grub.d

Les fichiers de ce répertoire sont exécutés dans l'ordre alphanumérique et servent à construire les menus de GRUB 2 :

```
[root@centos8 ~]# ls -l /etc/grub.d
total 92
-rwxr-xr-x. 1 root root 8958 Mar  2 15:51 00_header
-rwxr-xr-x. 1 root root 1043 Jun 15 2020 00_tuned
-rwxr-xr-x. 1 root root 232 Mar  2 15:51 01_users
-rwxr-xr-x. 1 root root 832 Mar  2 15:51 08_fallback_counting
-rwxr-xr-x. 1 root root 14088 Mar  2 15:51 10_linux
```

```
-rwxr-xr-x. 1 root root 830 Mar 2 15:51 10_reset_boot_success
-rwxr-xr-x. 1 root root 889 Mar 2 15:51 12_menu_auto_hide
-rwxr-xr-x. 1 root root 11696 Mar 2 15:51 20_linux_xen
-rwxr-xr-x. 1 root root 2559 Mar 2 15:51 20_ppc_terminfo
-rwxr-xr-x. 1 root root 10670 Mar 2 15:51 30_os-prober
-rwxr-xr-x. 1 root root 1412 Mar 2 15:51 30_uefi-firmware
-rwxr-xr-x. 1 root root 214 Mar 2 15:51 40_custom
-rwxr-xr-x. 1 root root 216 Mar 2 15:51 41_custom
-rw-r--r--. 1 root root 483 Mar 2 15:51 README
```

- **Le fichier /etc/grub.d/10_Linux,**
 - Le fichier **10_Linux** contient des boucles pour rechercher des noyaux Linux,
- **Le fichier /etc/grub.d/30_os-prober,**
 - Ce fichier recherche des éventuels systèmes d'exploitation autre que Linux,
- **Les fichiers /etc/grub.d/40_custom et /etc/grub.d/41_custom,**
 - Ces deux fichiers sont fournis en tant que modèles à personnaliser.

Le Fichier /boot/grub2/grubenv

Le fichier **/boot/grub2/grubenv** contient l'entrée du menu sauvegardée, référencée par la directive **GRUB_DEFAULT** du fichier **/etc/default/grub** :

```
[root@centos8 ~]# cat /boot/grub2/grubenv
# GRUB Environment Block
saved_entry=de79af4f226d480fa7d3fec4cabbf97a-4.18.0-240.22.1.el8_3.x86_64
kernelopts=root=UUID=4c0cc28c-0d59-45be-bd73-d292b80be33c ro crashkernel=auto resume=UUID=c8bb3f47-d67f-4b21-b781-766899dc83d4 rhgb quiet
boot_success=1
boot_indeterminate=0
#####
#####
#####
#####
#####
```

```
#####  
#####  
#####  
#####  
#####  
##### [root@centos8 ~]#
```

Modifier la Configuration de GRUB 2 en Ligne de Commande

Lors du démarrage de GRUB 2, trois actions sont possibles à partir du menu :

- Lancer un système d'exploitation en le sélectionnant avec les flèches puis en appuyant sur la touche ,
- Lancer l'éditeur en appuyant sur la touche **e**,
- Lancer l'interface de la ligne de commande GRUB en appuyant sur la touche **c**.

En mode édition notez l'utilisation des touches suivantes :

- **flèches** : se déplacer dans l'écran. L'édition se fait en utilisant simplement les touches du clavier,
- **Crtl-X** : démarrer avec la configuration modifiée,
- **echap** : abandonner les modifications et retourner à l'interface menu de GRUB 2.

Chargeurs de Démarrages Alternatifs

Systemd-boot

Un Chargeur de Démarrage étroitement lié à Systemd (voir plus bas), celui-ci connaît actuellement un gain de popularité.

U-boot

Un Chargeur de Démarrage qui peut booter n'importe quelle image à partir de n'importe quel support.

Le Projet Syslinux

SYSLINUX

Un Chargeur de Démarrage pour les systèmes qui utilisent le système de fichier FAT. Par exemple le systèmes sur clefs USB.

EXTLINUX

Un Chargeur de Démarrage de petite taille qui sait booter des systèmes de fichier, EXT2, EXT3, EXT4 et BRTFS.

ISOLINUX

Un Chargeur de Démarrage pour booter des LiveCD et LiveDVD. Dans le cas d'ISOLINUX, deux fichiers sont nécessaires :

- **isolinux.bin** qui contient l'image du Chargeur de Démarrage et
- **isolinux.cfg** qui contient les paramètres de configuration.

PXELINUX

Un Chargeur de Démarrage pour booter à partir d'un serveur réseau. Ce système utilise le standard **PXE** (*Pre-boot Execution Environment*) qui utilise :

- **DHCP** pour attribuer une adresse IP à la machine et
- **BOOTP** pour charger l'image du Chargeur de Démarrage à partir du serveur en utilisant le protocole **TFTP** (*Trivial File Transfer Protocol*). L'image à télécharger doit s'appeler **/tftpboot/pxelinux.0** et chaque machine doit avoir un fichier de configuration dans le répertoire **/tftpboot/pxelinux.cfg/**

Isodhpx

Un Chargeur de Démarrage hybride, appelé **isodhpx.bin**, qui peut être chargé sur un disque **ou** une clef USB. Le fichier isodhpx.bin est créé avec le programme **xorriso**.

Initramfs

Le fichier Initramfs *INITial Ram File System* est une archive au format cpio :

```
[root@centos8 ~]# cp /boot/initramfs-4.18.0-240.22.1.el8_3.x86_64.img /tmp/custom
[root@centos8 ~]# cd /tmp
[root@centos8 tmp]# ls
cpio.list
custom
dateref
incremental.tar
mbr.save
systemd-private-9af7a2f7444849578f55b306bfd9f820-chronyd.service-iQiNzF
tblpart.save
test.cpio
test.print
test.tar
tmp.iso
vg0_backup

[root@centos8 tmp]# mv custom custom.gz

[root@centos8 tmp]# gunzip custom.gz

[root@centos8 tmp]# mkdir initramfs

[root@centos8 tmp]# cd initramfs

[root@centos8 initramfs]# cpio -cid -I ../custom
216 blocks
```



```
[root@centos8 initramfs]# ls
bin dev etc init lib lib64 proc root run sbin shutdown sys sysroot tmp usr var
```

Examiner l'image existante

Pour examiner une image initramfs archivée, il convient d'utiliser la commande **lsinitrd** :

```
[root@centos8 tmp]# lsinitrd custom | more
Image: custom: 25M
=====
Early CPIO image
=====
drwxr-xr-x   3 root    root          0 Feb 22 10:57 .
-rw-r--r--   1 root    root          2 Feb 22 10:57 early_cpio
drwxr-xr-x   3 root    root          0 Feb 22 10:57 kernel
drwxr-xr-x   3 root    root          0 Feb 22 10:57 kernel/x86
drwxr-xr-x   2 root    root          0 Feb 22 10:57 kernel/x86/microcode
-rw-r--r--   1 root    root    109568 Feb 22 10:57 kernel/x86/microcode/GenuineIntel.bin
=====
Version: dracut-049-95.git20200804.el8_3.4

Arguments: -f --kver '4.18.0-240.22.1.el8_3.x86_64'

dracut modules:
bash
systemd
systemd-initrd
nss-softokn
rngd
i18n
network-legacy
network
ifcfg
```

```
drm
plymouth
prefixdevname
kernel-modules
kernel-modules-extra
kernel-network-modules
resume
rootfs-block
terminfo
udev-rules
biosdevname
dracut-systemd
usrmount
base
fs-lib
memstrack
microcode_ctl-fw_dir_override
shutdown
```

```
=====
drwxr-xr-x 12 root    root          0 Feb 22 10:57 .
crw-r--r--  1 root    root           5,  1 Feb 22 10:57 dev/console
crw-r--r--  1 root    root           1, 11 Feb 22 10:57 dev/kmsg
crw-r--r--  1 root    root           1,  3 Feb 22 10:57 dev/null
crw-r--r--  1 root    root           1,  8 Feb 22 10:57 dev/random
crw-r--r--  1 root    root           1,  9 Feb 22 10:57 dev/urandom
lrwxrwxrwx  1 root    root            7 Feb 22 10:57 bin -> usr/bin
drwxr-xr-x  2 root    root            0 Feb 22 10:57 dev
drwxr-xr-x 11 root    root            0 Feb 22 10:57 etc
-rw-r--r--  1 root    root          30 Nov 10  2020 etc/centos-release
drwxr-xr-x  2 root    root            0 Feb 22 10:57 etc/cmdline.d
drwxr-xr-x  2 root    root            0 Feb 22 10:57 etc/conf.d
-rw-r--r--  1 root    root         124 Feb 22 10:57 etc/conf.d/systemd.conf
--More--
```

Le script init

RHEL/CentOS 8 utilise le système de démarrage **systemd**. Pour cette raison le script init est un lien symbolique vers **/usr/lib/systemd/systemd** :

```
[root@centos8 tmp]# lsinitrd custom | grep usr/lib/systemd/systemd | grep init
lrwxrwxrwx    1 root      root          23 Feb 22 10:57 init -> usr/lib/systemd/systemd
```

Consulter le contenu d'un fichier dans initramfs

Pour consulter le contenu d'un fichier inclus dans l'initramfs, il convient d'utiliser la commande **lsinitrd** de la manière suivante :

```
[root@centos8 tmp]# lsinitrd /boot/initramfs-$(uname -r).img /etc/ld.so.conf
include ld.so.conf.d/*.conf
```

Processus de Démarrage du Noyau Linux

Le processus de démarrage du Noyau Linux peut être divisé en 6 étapes :

Etape	Description
Chargement, configuration et exécution du chargeur du noyau	Le fichier bootsect.s est chargé en mémoire par le BIOS. Une fois configuré celui-ci charge le reste du noyau en mémoire
Configuration des paramètres et bascule vers le mode 32 bits	Le fichier boot.s met en place un IDT (<i>Interrupt Descriptor Table</i>) temporaire et GDT (<i>Global Descriptor Table</i>) temporaire et gère le basculement vers le mode 32 bits
Décompression du Noyau	Le fichier head.s décompresse le noyau
Initialisation du noyau et de la mémoire	Le fichier head.s crée un GDT et IDT définitif
Configuration du noyau	Le fichier main.c met en place les contraintes de mémoire et configure la mémoire virtuelle
Création du processus Init	Le fichier main.c crée le processus init

La fonction **init_post()** essaie ensuite d'exécuter un des processus suivant dans l'ordre :

- /sbin/init ==> /usr/sbin/init ==> /usr/lib/systemd/systemd
- /etc/init
- /bin/init
- /bin/sh ==> /bin/bash ==> /usr/bin/bash

Dans le cas d'un échec à ce stade le message **Kernel Panic** sera affiché.

Systemd

Les systèmes de démarrage antérieurs à Systemd, **SysVinit** et **Upstart**, étaient des systèmes de démarrage **séquentiels**.

Systemd essaie, par contre, de démarrer autant de services en parallèle que possible. Ceci est rendu possible car la majorité d'architectures matérielles modernes sont multi-cœurs. Si un service dépend d'un autre qui n'est pas encore démarré ce premier est mis en attente dans une mémoire tampon. Qui plus est, les services qui ne sont pas nécessaires au démarrage de la machine, tel cups, ne sont démarrés ultérieurement que si nécessaire. Lors de démarrage, les partitions sont montées en parallèle. Dernièrement, **Systemd** remplace les scripts de démarrage traditionnels avec des binaires compilés, beaucoup plus rapides que leur prédécesseurs.

Au lieu de parler de scripts de démarrage et de niveaux d'exécution, **Systemd** utilise la terminologie **Unités** (*Units*) et **Cibles** (*Targets*). Une Unité peut être :

- **.automount** - active la fonctionnalité d'automount.
- **.device** - expose une périphérique dans systemd.
- **.mount** - contrôle quand et comment les systèmes de fichiers sont montés.
- **.path** - active un service quand il y a un accès à un fichier ou répertoire sous surveillance par le système.
- **.service** - démarre, arrête, redémarre ou recharge un service.
- **.scope** - gère des services.
- **.slice** - regroupe des Unités dans une arborescence afin de limiter des ressources en utilisant des CGroups.
- **.snapshot** - un état sauvegardé du gestionnaire Systemd.
- **.socket** - permet aux Unités d'utiliser des sockets pour la communication inter-processus.
- **.swap** - encapsule une périphérique ou un fichier swap.
- **.timer** - déclenche l'activation d'autres Unités en utilisant des minuteurs de Systemd.
- **.target** - regroupe des Unités multiples afin qu'elles puissent être démarrées en même temps. Par exemple **network.target** regroupe toutes les Unités nécessaires pour démarrer toutes les interfaces réseaux en même temps.

Une Cible est en quelque sorte une **grande étape** dans le démarrage du système :

- **halt.target** - arrête le système.
- **poweroff.target** - arrête le système et coupe le courant.
- **shutdown.target** - arrête le système.
- **rescue.target** - démarre le système en mode single-user (seul root peut s'y connecter). Tous les systèmes de fichiers sont montés mais le réseau n'est pas démarré.
- **emergency.target** - démarre le système en mode single-user (seul root peut s'y connecter). Uniquement le système de fichiers racine est monté en mode lecture seule. Le réseau n'est pas démarré.
- **multi-user.target** - démarre le système en mode multi-utilisateur avec tous les systèmes de fichiers montés et le service network démarré.
- **graphical.target** - démarre le système en multi-user.target puis démarre l'interface graphique.
- **hibernate.target** - sauvegarde l'état courant sur disque et arrête le système. Quand le système est démarré, l'état est restauré.
- **reboot.target** - redémarre le système.

Systemd utilise des Cibles d'une manière similaire à ce que **SysVinit** utilise des niveaux d'exécution. Pour rendre la transition plus facile, il existe des **Cibles** qui "simulent" les niveaux d'exécution de **SysVinit** :

```
[root@centos8 ~]# ls -l /usr/lib/systemd/system/runlevel*
lrwxrwxrwx. 1 root root 15 Apr  7 16:55 /usr/lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 Apr  7 16:55 /usr/lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 Apr  7 16:55 /usr/lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Apr  7 16:55 /usr/lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Apr  7 16:55 /usr/lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 Apr  7 16:55 /usr/lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 Apr  7 16:55 /usr/lib/systemd/system/runlevel6.target -> reboot.target
```

```
/usr/lib/systemd/system/runlevel1.target.wants:
total 0
```

```
/usr/lib/systemd/system/runlevel2.target.wants:
total 0
```

```
/usr/lib/systemd/system/runlevel3.target.wants:
total 0
```

```
/usr/lib/systemd/system/runlevel4.target.wants:
total 0
```

```
/usr/lib/systemd/system/runlevel5.target.wants:
total 0
```

LAB #1 - La Commande systemctl

Pour visualiser la liste des Unités, il convient d'utiliser la commande **systemctl** avec l'option **list-units** :

```
[root@centos8 ~]# systemctl list-units
```

UNIT	LOAD	ACTIVE	SUB
DESCRIPTION			
proc-sys-fs-binfmt_misc.automount	loaded	active	waiting
Arbitrary Executable File Formats File System Automount Point			
sys-devices-pci0000:00-0000:00:01.1-ata2-host1-target1:0:0-1:0:0:0-block-sr0.device	loaded	active	plugged
CD-ROM			
sys-devices-pci0000:00-0000:00:03.0-net-enp0s3.device	loaded	active	plugged
82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)			
sys-devices-pci0000:00-0000:00:05.0-sound-card0.device	loaded	active	plugged
82801AA AC'97 Audio Controller			
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda1.device	loaded	active	plugged
VBOX_HARDDISK 1			
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda2.device	loaded	active	plugged
VBOX_HARDDISK 2			
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda-sda3.device	loaded	active	plugged
VBOX_HARDDISK 3			
sys-devices-pci0000:00-0000:00:0d.0-ata3-host2-target2:0:0-2:0:0:0-block-sda.device	loaded	active	plugged
VBOX_HARDDISK			
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb1.device	loaded	active	plugged
VBOX_HARDDISK 1			
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb10.device	loaded	active	plugged
VBOX_HARDDISK 10			

```
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb11.device loaded active plugged
VBOX_HARDDISK my_ext4
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb12.device loaded active plugged
VBOX_HARDDISK 12
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb2.device loaded active plugged
VBOX_HARDDISK 2
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb3.device loaded active plugged
VBOX_HARDDISK 3
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb4.device loaded active plugged
VBOX_HARDDISK 4
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb5.device loaded active plugged
VBOX_HARDDISK 5
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb6.device loaded active plugged
VBOX_HARDDISK 6
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb7.device loaded active plugged
VBOX_HARDDISK 7
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb8.device loaded active plugged
VBOX_HARDDISK 8
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb-sdb9.device loaded active plugged
VBOX_HARDDISK 9
sys-devices-pci0000:00-0000:00:0d.0-ata4-host3-target3:0:0-3:0:0:0-block-sdb.device loaded active plugged
VBOX_HARDDISK
sys-devices-platform-serial8250-tty-ttyS0.device loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS0
sys-devices-platform-serial8250-tty-ttyS1.device loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS1
sys-devices-platform-serial8250-tty-ttyS2.device loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS2
sys-devices-platform-serial8250-tty-ttyS3.device loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS3
sys-devices-virtual-block-dm\x2d0.device loaded active plugged
/sys/devices/virtual/block/dm-0
sys-devices-virtual-block-dm\x2d1.device loaded active plugged
/sys/devices/virtual/block/dm-1
```

sys-devices-virtual-net-virbr0.device	loaded active plugged
/sys/devices/virtual/net/virbr0	
sys-devices-virtual-net-virbr0\x2dnic.device	loaded active plugged
/sys/devices/virtual/net/virbr0-nic	
sys-module-configfs.device	loaded active plugged
/sys/module/configfs	
sys-subsystem-net-devices-enp0s3.device	loaded active plugged
82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)	
sys-subsystem-net-devices-virbr0.device	loaded active plugged
/sys/subsystem/net/devices/virbr0	
sys-subsystem-net-devices-virbr0\x2dnic.device	loaded active plugged
/sys/subsystem/net/devices/virbr0-nic	
-.mount	loaded active mounted
Root Mount	
boot.mount	loaded active mounted
/boot	
dev-hugepages.mount	loaded active mounted
Huge Pages File System	
dev-mqueue.mount	loaded active mounted
POSIX Message Queue File System	
run-user-1000.mount	loaded active mounted
/run/user/1000	
sys-kernel-config.mount	loaded active mounted
Kernel Configuration File System	
sys-kernel-debug.mount	loaded active mounted
Kernel Debug File System	
sys-kernel-tracing.mount	loaded active mounted
/sys/kernel/tracing	
var-lib-nfs-rpc_pipefs.mount	loaded active mounted
RPC Pipe File System	
cups.path	loaded active running
CUPS Scheduler	
systemd-ask-password-plymouth.path	loaded active waiting
Forward Password Requests to Plymouth Directory Watch	

systemd-ask-password-wall.path	loaded active waiting
Forward Password Requests to Wall Directory Watch	
init.scope	loaded active running
System and Service Manager	
session-96.scope	loaded active running
Session 96 of user trainee	
atd.service	loaded active running
Job spooling tools	
auditd.service	loaded active running
Security Auditing Service	
avahi-daemon.service	loaded active running
Avahi mDNS/DNS-SD Stack	
chronyd.service	loaded active running
NTP client/server	
crond.service	loaded active running
Command Scheduler	
cups.service	loaded active running
CUPS Scheduler	
dbus.service	loaded active running
D-Bus System Message Bus	
dracut-shutdown.service	loaded active exited
Restore /run/initramfs on shutdown	
firewalld.service	loaded active running
firewalld - dynamic firewall daemon	
lines 1-57	

Pour consulter la liste des Unités inactifs, utilisez la commande suivante :

```
[root@centos8 ~]# systemctl list-units --all | grep inactive | more
● boot.automount
not-found inactive dead      boot.automount
  proc-fs-nfsd.mount
loaded    inactive dead      NFSD configuration filesystem
  proc-sys-fs-binfmt_misc.mount
```

```
loaded    inactive dead    Arbitrary Executable File Formats File System
  sys-fs-fuse-connections.mount
loaded    inactive dead    FUSE Control File System
● sysroot.mount
not-found inactive dead    sysroot.mount
  tmp.mount
loaded    inactive dead    Temporary Directory (/tmp)
  var-lib-machines.mount
loaded    inactive dead    Virtual Machine and Container Storage
  systemd-ask-password-console.path
loaded    inactive dead    Dispatch Password Requests to Console Directory Watch
● apparmor.service
not-found inactive dead    apparmor.service
  auth-rpcgss-module.service
loaded    inactive dead    Kernel Module supporting RPCSEC_GSS
  cpupower.service
loaded    inactive dead    Configure CPU power related settings
● display-manager.service
not-found inactive dead    display-manager.service
  dm-event.service
loaded    inactive dead    Device-mapper event daemon
  dnf-makecache.service
loaded    inactive dead    dnf makecache
  dracut-cmdline.service
loaded    inactive dead    dracut cmdline hook
  dracut-initqueue.service
loaded    inactive dead    dracut initqueue hook
  dracut-mount.service
loaded    inactive dead    dracut mount hook
  dracut-pre-mount.service
loaded    inactive dead    dracut pre-mount hook
  dracut-pre-pivot.service
loaded    inactive dead    dracut pre-pivot and cleanup hook
  dracut-pre-trigger.service
```

loaded	inactive dead	dracut pre-trigger hook
dracut-pre-udev.service		
loaded	inactive dead	dracut pre-udev hook
ebtables.service		
loaded	inactive dead	Ethernet Bridge Filtering tables
emergency.service		
loaded	inactive dead	Emergency Shell
initrd-cleanup.service		
loaded	inactive dead	Cleaning Up and Shutting Down Daemons
initrd-parse-etc.service		
loaded	inactive dead	Reload Configuration from the Real Root
initrd-switch-root.service		
loaded	inactive dead	Switch Root
initrd-udevadm-cleanup-db.service		
loaded	inactive dead	Cleanup udevd DB
●	ip6tables.service	
not-found	inactive dead	ip6tables.service
●	ipset.service	
not-found	inactive dead	ipset.service
●	iptables.service	
not-found	inactive dead	iptables.service
iscsi-onboot.service		
loaded	inactive dead	Special handling of early boot iSCSI sessions
iscsi.service		
loaded	inactive dead	Login and scanning of iSCSI devices
iscsid.service		
loaded	inactive dead	Open-iSCSI
iscsiuio.service		
loaded	inactive dead	iSCSI UserSpace I/O driver
ldconfig.service		
loaded	inactive dead	Rebuild Dynamic Linker Cache
libvirt-guests.service		
loaded	inactive dead	Suspend/Resume Running libvirt Guests
libvirtd.service		

loaded	inactive	dead	Virtualization daemon
			loadmodules.service
loaded	inactive	dead	Load legacy module configuration
●			lvm2-activation.service
not-found	inactive	dead	lvm2-activation.service
			lvm2-lvmpolld.service
loaded	inactive	dead	LVM2 poll daemon
			mdmonitor.service
loaded	inactive	dead	Software RAID monitoring and management
			microcode.service
loaded	inactive	dead	Load CPU microcode update
●			network.service
not-found	inactive	dead	network.service
			nfs-blkmap.service
loaded	inactive	dead	pNFS block layout mapping daemon
			nfs-convert.service
loaded	inactive	dead	Preprocess NFS configuration conversion
			nfs-idmapd.service
loaded	inactive	dead	NFSv4 ID-name mapping service
			nfs-mountd.service
loaded	inactive	dead	NFS Mount Daemon
			nfs-server.service
loaded	inactive	dead	NFS server and services
			nfs-utils.service
loaded	inactive	dead	NFS server and client services
			nfsdclld.service
loaded	inactive	dead	NFSv4 Client Tracking Daemon
			nftables.service
loaded	inactive	dead	Netfilter Tables
●			ntpd.service
not-found	inactive	dead	ntpd.service
●			ntpddate.service
not-found	inactive	dead	ntpddate.service
			plymouth-switch-root.service

```
loaded    inactive dead    Plymouth switch root service
rc-local.service
loaded    inactive dead    /etc/rc.d/rc.local Compatibility
rescue.service
loaded    inactive dead    Rescue Shell
rpc-gssd.service
loaded    inactive dead    RPC security service for NFS client and server
--More--
```

Les points noirs au début de certaines lignes dans la sortie ci-dessus sont en réalité des points blancs à l'écran. Ces points impliquent que le service, la cible ou l'unité spécifié n'a pas été trouvé sur le système. Par exemple :

```
[root@centos8 ~]# systemctl status ntpd
Unit ntpd.service could not be found.
```

Pour consulter la liste des Unités ainsi que leurs statuts, utilisez la commande suivante :

```
[root@centos8 ~]# systemctl list-unit-files | more
UNIT FILE                                     STATE
proc-sys-fs-binfmt_misc.automount           static
-.mount                                       generated
boot.mount                                    generated
dev-hugepages.mount                          static
dev-mqueue.mount                             static
proc-fs-nfsd.mount                           static
proc-sys-fs-binfmt_misc.mount                static
sys-fs-fuse-connections.mount                static
sys-kernel-config.mount                      static
sys-kernel-debug.mount                       static
tmp.mount                                     disabled
var-lib-machines.mount                       static
var-lib-nfs-rpc_pipefs.mount                 static
cups.path                                     enabled
systemd-ask-password-console.path            static
```

systemd-ask-password-plymouth.path	static
systemd-ask-password-wall.path	static
session-96.scope	transient
arp-ethers.service	disabled
atd.service	enabled
auditd.service	enabled
auth-rpcgss-module.service	static
autovt@.service	enabled
avahi-daemon.service	enabled
blk-availability.service	disabled
chrony-dnssrv@.service	static
chrony-wait.service	disabled
chronyd.service	enabled
cockpit-motd.service	static
cockpit-wsinstance-http-redirect.service	static
cockpit-wsinstance-http.service	static
cockpit-wsinstance-https-factory@.service	static
cockpit-wsinstance-https@.service	static
cockpit.service	static
console-getty.service	disabled
container-getty@.service	static
cpupower.service	disabled
crond.service	enabled
cups-browsed.service	disabled
cups.service	enabled
dbus-org.fedoraproject.FirewallD1.service	enabled
dbus-org.freedesktop.Avahi.service	enabled
dbus-org.freedesktop.hostname1.service	static
dbus-org.freedesktop.import1.service	static
dbus-org.freedesktop.locale1.service	static
dbus-org.freedesktop.login1.service	static
dbus-org.freedesktop.machine1.service	static
dbus-org.freedesktop.nm-dispatcher.service	enabled
dbus-org.freedesktop.portable1.service	static

dbus-org.freedesktop.timedate1.service	enabled
dbus.service	static
debug-shell.service	disabled
dm-event.service	static
dnf-makecache.service	static
dnsmasq.service	disabled
dracut-cmdline.service	static
--More--	

Pour visualiser les Unités d'un type spécifique, il convient d'utiliser l'option **-t** :

```
[root@centos8 ~]# systemctl list-unit-files -t mount
UNIT FILE                                STATE
-.mount                                 generated
boot.mount                             generated
dev-hugepages.mount                    static
dev-mqueue.mount                       static
proc-fs-nfsd.mount                     static
proc-sys-fs-binfmt_misc.mount          static
sys-fs-fuse-connections.mount          static
sys-kernel-config.mount                static
sys-kernel-debug.mount                 static
tmp.mount                              disabled
var-lib-machines.mount                 static
var-lib-nfs-rpc_pipefs.mount           static

12 unit files listed.
```

Dans la colonne STATE on voit les mots **static** et **generated**.

- STATE = static
 - Ceci implique que l'Unité ne peut ni être démarrée, ni être arrêtée par l'administrateur. Le démarrage et l'arrêt d'une telle Unité est effectué par le système. En règle générale, les Unités dont le STATE est static sont des dépendances d'autres Unité

- STATE = generated
 - Ceci implique que le fichier a été généré automatiquement en utilisant les informations dans le fichier **/etc/fstab** lors du démarrage du système. Dans le cas d'un point de montage, l'exécutable responsable de la génération du fichier est **/lib/systemd/systemd-generators/systemd-fstab-generator** :

```
[root@centos8 ~]# ls -l /lib/systemd/system-generators/systemd-fstab-generator
-rwxr-xr-x. 1 root root 46096 Apr  7 16:56 /lib/systemd/system-generators/systemd-fstab-generator
```

Il existe aussi d'autres exécutables responsables de la génération d'autres fichiers :

```
[root@centos8 ~]# ls -l /lib/systemd/system-generators
total 508
-rwxr-xr-x. 1 root root  504 Jan  4 11:25 kdump-dep-generator.sh
-r-xr-xr-x. 1 root root 134976 Aug 17 2020 lvm2-activation-generator
-rwxr-xr-x. 1 root root 67792 Jul 20 2020 nfs-server-generator
-rwxr-xr-x. 1 root root 38216 Jul 20 2020 rpc-pipefs-generator
-rwxr-xr-x. 1 root root  743 Apr 23 2020 selinux-autorelabel-generator.sh
-rwxr-xr-x. 1 root root 33504 Apr  7 16:56 systemd-cryptsetup-generator
-rwxr-xr-x. 1 root root 16648 Apr  7 16:56 systemd-debug-generator
-rwxr-xr-x. 1 root root 46096 Apr  7 16:56 systemd-fstab-generator
-rwxr-xr-x. 1 root root 17064 Apr  7 16:56 systemd-getty-generator
-rwxr-xr-x. 1 root root 29432 Apr  7 16:56 systemd-gpt-auto-generator
-rwxr-xr-x. 1 root root 12568 Apr  7 16:56 systemd-hibernate-resume-generator
-rwxr-xr-x. 1 root root 12368 Apr  7 16:56 systemd-rc-local-generator
-rwxr-xr-x. 1 root root 12408 Apr  7 16:56 systemd-system-update-generator
-rwxr-xr-x. 1 root root 33544 Apr  7 16:56 systemd-sysv-generator
-rwxr-xr-x. 1 root root 17024 Apr  7 16:56 systemd-veritysetup-generator
```

Les options de la commande **systemctl** sont :

```
[root@centos8 ~]# systemctl --help
systemctl [OPTIONS...] {COMMAND} ...
```

Query or send control commands to the systemd manager.


```
-h --help          Show this help
--version          Show package version
--system           Connect to system manager
--user             Connect to user service manager
-H --host=[USER@]HOST
                   Operate on remote host
-M --machine=CONTAINER
                   Operate on local container
-t --type=TYPE      List units of a particular type
--state=STATE       List units with particular LOAD or SUB or ACTIVE state
-p --property=NAME  Show only properties by this name
-a --all            Show all properties/all units currently in memory,
                   including dead/empty ones. To list all units installed on
                   the system, use the 'list-unit-files' command instead.
--failed            Same as --state=failed
-l --full           Don't ellipsize unit names on output
-r --recursive      Show unit list of host and local containers
--reverse           Show reverse dependencies with 'list-dependencies'
--job-mode=MODE     Specify how to deal with already queued jobs, when
                   queueing a new job
--show-types        When showing sockets, explicitly show their type
--value            When showing properties, only print the value
-i --ignore-inhibitors
                   When shutting down or sleeping, ignore inhibitors
--kill-who=WHO      Who to send signal to
-s --signal=SIGNAL  Which signal to send
--now              Start or stop unit in addition to enabling or disabling it
--dry-run           Only print what would be done
-q --quiet          Suppress output
--wait             For (re)start, wait until service stopped again
--no-block          Do not wait until operation finished
--no-wall           Don't send wall message before halt/power-off/reboot
--no-reload         Don't reload daemon after en-/dis-abling unit files
--no-legend         Do not print a legend (column headers and hints)
```

```
--no-pager      Do not pipe output into a pager
--no-ask-password
                Do not ask for system passwords
--global        Enable/disable/mask unit files globally
--runtime       Enable/disable/mask unit files temporarily until next
                reboot
-f --force      When enabling unit files, override existing symlinks
                When shutting down, execute action immediately
--preset-mode=  Apply only enable, only disable, or all presets
--root=PATH     Enable/disable/mask unit files in the specified root
                directory
-n --lines=INTEGER Number of journal entries to show
-o --output=STRING Change journal output mode (short, short-precise,
                short-iso, short-iso-precise, short-full,
                short-monotonic, short-unix,
                verbose, export, json, json-pretty, json-sse, cat)
--firmware-setup Tell the firmware to show the setup menu on next boot
--plain         Print unit dependencies as a list instead of a tree
```

lines 1-57

LAB #2 - Fichiers de Configuration

2.1 - Fichiers de Configuration par Défaut

Les fichiers de configuration des Cibles et fichiers de configuration des Unités installés par des paquets se trouvent dans le répertoire **/usr/lib/systemd/system** :

```
[root@centos8 ~]# pkg-config systemd --variable=systemdsystemunitdir
/usr/lib/systemd/system
```

```
[root@centos8 ~]# ls -l /usr/lib/systemd/system | more
```

total 1464

```
-rw-r--r--. 1 root root 275 Apr 26 2020 arp-ethers.service
-rw-r--r--. 1 root root 222 May 11 2019 atd.service
-rw-r--r--. 1 root root 1512 Apr 23 2020 auditd.service
-rw-r--r--. 1 root root 628 Jul 20 2020 auth-rpcgss-module.service
lrwxrwxrwx. 1 root root 14 Apr 7 16:55 autovt@.service -> getty@.service
-rw-r--r--. 1 root root 1044 Nov 16 2020 avahi-daemon.service
-rw-r--r--. 1 root root 870 Nov 16 2020 avahi-daemon.socket
-rw-r--r--. 1 root root 956 Apr 7 16:54 basic.target
drwxr-xr-x. 2 root root 6 Apr 7 16:55 basic.target.wants
-r--r--r--. 1 root root 408 Aug 17 2020 blk-availability.service
-rw-r--r--. 1 root root 419 Jun 22 2018 bluetooth.target
-rw-r--r--. 1 root root 455 Apr 7 16:54 boot-complete.target
-rw-r--r--. 1 root root 209 Nov 19 2019 chrony-dnssrv@.service
-rw-r--r--. 1 root root 138 Nov 19 2019 chrony-dnssrv@.timer
-rw-r--r--. 1 root root 491 Nov 19 2019 chronyd.service
-rw-r--r--. 1 root root 472 May 10 2019 chrony-wait.service
-rw-r--r--. 1 root root 222 Aug 24 2020 cockpit-motd.service
-rw-r--r--. 1 root root 835 Aug 24 2020 cockpit.service
-rw-r--r--. 1 root root 373 Aug 24 2020 cockpit.socket
-rw-r--r--. 1 root root 251 Aug 24 2020 cockpit-wsinstance-http-redirect.service
-rw-r--r--. 1 root root 233 Aug 24 2020 cockpit-wsinstance-http-redirect.socket
-rw-r--r--. 1 root root 221 Aug 24 2020 cockpit-wsinstance-http.service
-rw-r--r--. 1 root root 165 Aug 24 2020 cockpit-wsinstance-https-factory@.service
-rw-r--r--. 1 root root 244 Aug 24 2020 cockpit-wsinstance-https-factory.socket
-rw-r--r--. 1 root root 215 Aug 24 2020 cockpit-wsinstance-http.socket
-rw-r--r--. 1 root root 264 Aug 24 2020 cockpit-wsinstance-https@.service
-rw-r--r--. 1 root root 478 Aug 24 2020 cockpit-wsinstance-https@.socket
-rw-r--r--. 1 root root 1082 Apr 7 16:55 console-getty.service
-rw-r--r--. 1 root root 1263 Apr 7 16:55 container-getty@.service
-rw-r--r--. 1 root root 294 Apr 8 15:21 cpupower.service
-rw-r--r--. 1 root root 356 Nov 8 2019 crond.service
-rw-r--r--. 1 root root 465 Jun 22 2018 cryptsetup-pre.target
-rw-r--r--. 1 root root 412 Jun 22 2018 cryptsetup.target
```

```

lrwxrwxrwx. 1 root root    13 Apr  7 16:55 ctrl-alt-del.target -> reboot.target
-rw-r--r--. 1 root root  234 Oct  4  2017 cups-browsed.service
-r--r--r--. 1 root root  142 Jun 15  2020 cups.path
-r--r--r--. 1 root root  248 Jun 15  2020 cups.service
-r--r--r--. 1 root root  136 Jun 15  2020 cups.socket
lrwxrwxrwx. 1 root root    25 Apr  7 16:55 dbus-org.freedesktop.hostname1.service -> systemd-hostnamed.service
lrwxrwxrwx. 1 root root    23 Apr  7 16:55 dbus-org.freedesktop.import1.service -> systemd-importd.service
lrwxrwxrwx. 1 root root    23 Apr  7 16:55 dbus-org.freedesktop.locale1.service -> systemd-localed.service
lrwxrwxrwx. 1 root root    22 Apr  7 16:55 dbus-org.freedesktop.login1.service -> systemd-logind.service
lrwxrwxrwx. 1 root root    24 Apr  7 16:55 dbus-org.freedesktop.machine1.service -> systemd-machined.service
lrwxrwxrwx. 1 root root    25 Apr  7 16:55 dbus-org.freedesktop.portable1.service -> systemd-portabled.service
lrwxrwxrwx. 1 root root    25 Apr  7 16:55 dbus-org.freedesktop.timedate1.service -> systemd-timedated.service
-rw-r--r--. 1 root root  380 Apr  7 12:08 dbus.service
-rw-r--r--. 1 root root  102 Apr  7 12:08 dbus.socket
drwxr-xr-x. 2 root root    6 Apr  7 16:55 dbus.target.wants
-rw-r--r--. 1 root root 1084 Apr  7 16:55 debug-shell.service
lrwxrwxrwx. 1 root root    16 Apr  7 16:55 default.target -> graphical.target
drwxr-xr-x. 2 root root    6 Apr  7 16:55 default.target.wants
-rw-r--r--. 1 root root  750 Jun 22  2018 dev-hugepages.mount
-rw-r--r--. 1 root root  665 Jun 22  2018 dev-mqueue.mount
-r--r--r--. 1 root root  345 Aug 17  2020 dm-event.service
-r--r--r--. 1 root root  248 Aug 17  2020 dm-event.socket
-rw-r--r--. 1 root root  457 Jun  2  2020 dnf-makecache.service
--More--

```

Certains fichiers de configuration sont créés à la volée dans le répertoire **/run/systemd/system** lors du runtime puis ils sont détruits quand le système n'en a plus besoin :

```

[root@centos8 ~]# ls -l /run/systemd/system/
total 0

```

Les fichiers de configuration des Unités créés par les utilisateurs doivent être mis dans le répertoire **/usr/lib/systemd/user** :

```

[root@centos8 ~]# pkg-config systemd --variable=systemduserunitdir

```

/usr/lib/systemd/user



Important : De cette façon les fichiers dans **/usr/lib/systemd/user** surchargent les fichiers dans le répertoire **/run/systemd/system** qui surchargent les fichiers dans le répertoire **/usr/lib/systemd/system**.

Prenons le cas du service **sshd** qui est configuré par le fichier **/usr/lib/systemd/system/sshd.service** :

```
[root@centos8 ~]# cat /usr/lib/systemd/system/sshd.service
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.target
Wants=sshd-keygen.target

[Service]
Type=notify
EnvironmentFile=-/etc/crypto-policies/back-ends/opensshserver.config
EnvironmentFile=-/etc/sysconfig/ssh
ExecStart=/usr/sbin/sshd -D $OPTIONS $CRYPTO_POLICY
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

Dans le fichier on peut noter la présence des lignes suivantes dans la section **[Unit]** :

- **Description=OpenSSH server daemon,**

- Cette directive est utilisée pour donner une courte description des fonctionnalités de l'Unité,
- **Documentation=man:sshd(8) man:sshd_config(5),**
 - Cette directive stipule les chapitres des manuels et les URLs contenant de l'information en relation avec l'Unité,
- **After=network.target sshd-keygen.target,**
 - Cette directive indique les cibles qui devraient être atteintes et les Unités qui devraient être démarrées avant l'Unité sshd. Par contre, cette directive ne spécifie pas une dépendance,
- **Wants=sshd-keygen.target,**
 - Cette directive stipule une dépendance douce. Autrement dit, Systemd essayera de démarrer l'Unité **sshd-keygen.target** mais si ce démarrage est en échec, l'Unité sshd sera démarré.

Dans le fichier on peut aussi noter la présence des lignes suivantes dans la section **[Service]** :

- **Type=notify,**
 - Cette directive indique que le service informera Systemd quand son démarrage a terminé,
- **ExecStart=/usr/sbin/sshd -D \$OPTIONS \$CRYPTO_POLICY,**
 - Cette directive définit l'exécutable à lancer,
- **ExecReload=/bin/kill -HUP \$MAINPID,**
 - Cette directive indique la commande nécessaire pour redémarrer le service,
- **KillMode=process,**
 - Cette directive indique comment les processus du service doivent être arrêtés. La valeur de **process** implique l'utilisation de SIGTERM suivi par SIGHUP,
- **Restart=on-failure,**
 - Cette ligne indique que le service doit être re-démarré en cas d'arrêt de celui-ci.
- **RestartSec=42s,**
 - Cette directive stipule le temps d'attente entre l'arrêt du service et le redémarrage par Systemd,

Dernièrement on note la présence de la ligne suivante dans la section **[Install]** :

- **WantedBy=multi-user.target,**
 - Cette directive indique la Cible dans laquelle le service doit être démarré. La présence de cette directive crée un lien symbolique dans le répertoire **/etc/systemd/system/multi-user.target.wants** qui pointe vers ce fichier.

Pour consulter **l'ensemble** des directives de configuration ainsi que leurs valeurs d'une Unité, il convient d'utiliser la commande **systemctl show** en spécifiant l'Unité concernée :

```
[root@centos8 ~]# systemctl show sshd
Type=notify
Restart=on-failure
NotifyAccess=main
RestartUsec=42s
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
RuntimeMaxUsec=infinity
WatchdogUsec=0
WatchdogTimestamp=Thu 2021-06-03 15:09:54 EDT
WatchdogTimestampMonotonic=12502561
PermissionsStartOnly=no
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
MainPID=902
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
Result=success
UID=[not set]
GID=[not set]
NRestarts=0
ExecMainStartTimestamp=Thu 2021-06-03 15:09:54 EDT
ExecMainStartTimestampMonotonic=12446178
ExecMainExitTimestampMonotonic=0
ExecMainPID=902
ExecMainCode=0
ExecMainStatus=0
ExecStart={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $OPTIONS $CRYPTO_POLICY ; ignore_errors=no ;
start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }
ExecReload={ path=/bin/kill ; argv[]=/bin/kill -HUP $MAINPID ; ignore_errors=no ; start_time=[n/a] ;
stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }
```

```
Slice=system.slice
ControlGroup=/system.slice/sshd.service
MemoryCurrent=6270976
CPUUsageNSec=[not set]
EffectiveCPUs=
EffectiveMemoryNodes=
TasksCurrent=1
IPIngressBytes=18446744073709551615
IPIngressPackets=18446744073709551615
IPEgressBytes=18446744073709551615
IPEgressPackets=18446744073709551615
Delegate=no
CPUAccounting=no
CPUWeight=[not set]
StartupCPUWeight=[not set]
CPUShares=[not set]
StartupCPUShares=[not set]
CPUQuotaPerSecUSec=infinity
CPUQuotaPeriodUSec=infinity
AllowedCPUs=
AllowedMemoryNodes=
IOAccounting=no
IOWeight=[not set]
StartupIOWeight=[not set]
BlockIOAccounting=no
BlockIOWeight=[not set]
lines 1-57
```

Pour consulter la liste des dépendances d'une Unité, il convient d'utiliser **systemctl list-dependencies** en spécifiant l'Unité concernée :

```
[root@centos8 ~]# systemctl list-dependencies sshd.service
sshd.service
● └─system.slice
● └─sshd-keygen.target
```


- |—sshd-keygen@ecdsa.service
- |—sshd-keygen@ed25519.service
- |—sshd-keygen@rsa.service
- |—sysinit.target
- |—dev-hugepages.mount
- |—dev-mqueue.mount
- |—dracut-shutdown.service
- |—import-state.service
- |—iscsi-onboot.service
- |—kmod-static-nodes.service
- |—ldconfig.service
- |—loadmodules.service
- |—lvm2-lvmpolld.socket
- |—lvm2-monitor.service
- |—nis-domainname.service
- |—plymouth-read-write.service
- |—plymouth-start.service
- |—proc-sys-fs-binfmt_misc.automount
- |—rngd.service
- |—selinux-autorelabel-mark.service
- |—sys-fs-fuse-connections.mount
- |—sys-kernel-config.mount
- |—sys-kernel-debug.mount
- |—systemd-ask-password-console.path
- |—systemd-binfmt.service
- |—systemd-firstboot.service
- |—systemd-hwdb-update.service
- |—systemd-journal-catalog-update.service
- |—systemd-journal-flush.service
- |—systemd-journald.service
- |—systemd-machine-id-commit.service
- |—systemd-modules-load.service
- |—systemd-random-seed.service
- |—systemd-sysctl.service

- |—systemd-sysusers.service
- |—systemd-tmpfiles-setup-dev.service
- |—systemd-tmpfiles-setup.service
- |—systemd-udev-trigger.service
- |—systemd-udevd.service
- |—systemd-update-done.service
- |—systemd-update-utmp.service
- |—cryptsetup.target
- |—local-fs.target
 - |—|.mount
 - |—boot.mount
 - |—systemd-remount-fs.service
- |—swap.target
 - |—dev-disk-by\x2duuid-c8bb3f47\x2dd67f\x2d4b21\x2ddb781\x2d766899dc83d4.swap

2.2 - Surchargement des Fichiers de Configuration par Défaut

Les fichiers de configuration par défaut peuvent aussi être surchargés par des fichiers dans d'autres répertoires :

```
[root@centos8 ~]# pkg-config systemd --variable=systemdsystemunitpath
/etc/systemd/system:/etc/systemd/system:/run/systemd/system:/usr/local/lib/systemd/system:/usr/lib/systemd/system
:/usr/lib/systemd/system:/lib/systemd/system
```

```
[root@centos8 ~]# ls -l /etc/systemd/system
total 4
drwxr-xr-x. 2 root root 31 May 8 2020 basic.target.wants
lrwxrwxrwx. 1 root root 41 May 8 2020 dbus-org.fedoraproject.FirewallD1.service ->
/usr/lib/systemd/system/firewalld.service
lrwxrwxrwx. 1 root root 44 Jun 3 14:02 dbus-org.freedesktop.Avahi.service -> /usr/lib/systemd/system/avahi-
daemon.service
lrwxrwxrwx. 1 root root 57 May 8 2020 dbus-org.freedesktop.nm-dispatcher.service ->
/usr/lib/systemd/system/NetworkManager-dispatcher.service
lrwxrwxrwx. 1 root root 41 May 8 2020 dbus-org.freedesktop.timedate1.service ->
```

```
/usr/lib/systemd/system/timedatex.service
lrwxrwxrwx. 1 root root 37 May 8 2020 default.target -> /lib/systemd/system/multi-user.target
drwxr-xr-x. 2 root root 32 May 8 2020 getty.target.wants
drwxr-xr-x. 2 root root 4096 Jun 3 14:02 multi-user.target.wants
drwxr-xr-x. 2 root root 48 May 8 2020 network-online.target.wants
drwxr-xr-x. 2 root root 33 Apr 19 12:07 nfs-blkmap.service.requires
drwxr-xr-x. 2 root root 33 Apr 19 12:07 nfs-idmapd.service.requires
drwxr-xr-x. 2 root root 33 Apr 19 12:07 nfs-mountd.service.requires
drwxr-xr-x. 2 root root 33 Apr 19 12:07 nfs-server.service.requires
drwxr-xr-x. 2 root root 26 Jun 3 14:02 printer.target.wants
drwxr-xr-x. 2 root root 52 Apr 19 12:07 remote-fs.target.wants
drwxr-xr-x. 2 root root 33 Apr 19 12:07 rpc-gssd.service.requires
drwxr-xr-x. 2 root root 33 Apr 19 12:07 rpc-statd-notify.service.requires
drwxr-xr-x. 2 root root 33 Apr 19 12:07 rpc-statd.service.requires
drwxr-xr-x. 2 root root 260 Jun 3 14:02 sockets.target.wants
drwxr-xr-x. 2 root root 235 Apr 19 12:07 sysinit.target.wants
lrwxrwxrwx. 1 root root 39 May 8 2020 syslog.service -> /usr/lib/systemd/system/rsyslog.service
lrwxrwxrwx. 1 root root 9 May 11 2019 systemd-timedated.service -> /dev/null
drwxr-xr-x. 2 root root 34 May 8 2020 timers.target.wants
```

LAB #3 - La Commande systemd-analyze

Pour avoir une évaluation du temps de démarrage, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# systemd-analyze
Startup finished in 1.665s (kernel) + 6.977s (initrd) + 8.458s (userspace) = 17.101s
multi-user.target reached after 5.528s in userspace
```

L'option **blame** de la commande systemd-analyze permet de voir le temps de démarrage de chaque Unité afin de pouvoir se concentrer sur les plus lentes :

```
[root@centos8 ~]# systemd-analyze blame
4.080s dracut-initqueue.service
```

```
3.257s kdump.service
1.727s tuned.service
1.415s initrd-switch-root.service
1.393s NetworkManager-wait-online.service
1.116s systemd-machined.service
1.111s dracut-cmdline.service
850ms sssd.service
791ms vdo.service
713ms firewalld.service
622ms lvm2-monitor.service
605ms polkit.service
508ms chronyd.service
434ms avahi-daemon.service
426ms systemd-logind.service
417ms netcf-transaction.service
410ms dracut-pre-udev.service
295ms libvirtd.service
275ms dnf-makecache.service
243ms systemd-udevd.service
221ms systemd-journald.service
196ms systemd-tmpfiles-setup.service
151ms dracut-pre-pivot.service
139ms sysroot.mount
139ms systemd-update-utmp-runlevel.service
122ms systemd-vconsole-setup.service
110ms lvm2-pvscan@8:25.service
98ms systemd-udev-trigger.service
82ms gssproxy.service
81ms cups.service
79ms initrd-parse-etc.service
77ms NetworkManager.service
72ms lvm2-pvscan@8:23.service
69ms systemd-user-sessions.service
68ms lvm2-pvscan@8:22.service
```

```
67ms unbound-anchor.service
66ms rsyslog.service
62ms boot.mount
56ms sshd.service
54ms smartd.service
54ms systemd-fsck@dev-disk-by\x2duuid-2ae4c035\x2d9244\x2d458c\x2d82c5\x2da49ae169cdb6.service
53ms user@1000.service
52ms auditd.service
51ms plymouth-quit.service
49ms rngd-wake-threshold.service
46ms import-state.service
46ms systemd-tmpfiles-setup-dev.service
43ms ksmtuned.service
42ms plymouth-quit-wait.service
42ms var-lib-nfs-rpc_pipefs.mount
42ms rpc-statd-notify.service
38ms plymouth-switch-root.service
37ms systemd-remount-fs.service
37ms plymouth-start.service
33ms dev-disk-by\x2duuid-c8bb3f47\x2dd67f\x2d4b21\x2db781\x2d766899dc83d4.swap
33ms systemd-tmpfiles-clean.service
31ms dev-hugepages.mount
```

lines 1-57

L'option **critical-chain** de la commande **systemd-analyze** permet de voir l'enchaînement des événements qui amènent au chargement de l'Unité qui est passée en argument :

```
[root@centos8 ~]# systemd-analyze critical-chain sshd.service
The time after the unit is active or started is printed after the "@" character.
The time the unit takes to start is printed after the "+" character.

sshd.service +56ms
└─network.target @3.799s
   └─NetworkManager.service @3.719s +77ms
```

```
└─network-pre.target @3.718s
  └─firewalld.service @3.004s +713ms
    └─polkit.service @2.397s +605ms
      └─basic.target @2.392s
        └─sockets.target @2.392s
          └─sssd-kcm.socket @2.391s
            └─sysinit.target @2.379s
              └─systemd-update-utmp.service @2.370s +8ms
                └─auditd.service @2.317s +52ms
                  └─systemd-tmpfiles-setup.service @2.118s +196ms
                    └─import-state.service @2.070s +46ms
                      └─local-fs.target @2.069s
                        └─boot.mount @2.006s +62ms
                          └─systemd-fsck@dev-disk-
by\x2duuid-2ae4c035\x2d9244\x2d458c\x2d82c5\x2da49ae169cdb6.service @1.943s +54ms
                  └─local-fs-pre.target @1.942s
                    └─lvm2-monitor.service @1.319s +622ms
                      └─dm-event.socket @1.317s
                        └─.mount
                          └─system.slice
                            └─.slice
```

Les options de la commande **systemd-analyze** sont :

```
[root@centos7 ~]# systemd-analyze --help
systemd-analyze [OPTIONS...] {COMMAND} ...
```

Process systemd profiling information

-h --help	Show this help
--version	Show package version
--system	Connect to system manager
--user	Connect to user service manager
--order	When generating a dependency graph, show only order

```
--require      When generating a dependency graph, show only requirement
--from-pattern=GLOB, --to-pattern=GLOB
                When generating a dependency graph, filter only origins
                or destinations, respectively
--fuzz=TIMESPAN When printing the tree of the critical chain, print also
                services, which finished TIMESPAN earlier, than the
                latest in the branch. The unit of TIMESPAN is seconds
                unless specified with a different unit, i.e. 50ms
--no-pager      Do not pipe output into a pager
```

Commands:

```
time           Print time spent in the kernel before reaching userspace
blame          Print list of running units ordered by time to init
critical-chain Print a tree of the time critical chain of units
plot           Output SVG graphic showing service initialization
dot            Output dependency graph in dot(1) format
set-log-level LEVEL Set logging threshold for systemd
dump           Output state serialization of service manager
```

LAB #4 - Les Cibles Systemd

Chaque Cible est décrite par un fichier de configuration :

```
[root@centos8 ~]# cat /usr/lib/systemd/system/graphical.target
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
```

```
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes
```

Dans ce fichier on peut noter la présence des lignes suivantes :

- **Requires=multi-user.target,**
 - Cette ligne indique que le **graphical.target** ne peut pas être atteint si le **multi-user.target** n'a pas été atteint au préalable,
- **After=multi-user.target rescue.service rescue.target display-manager.service,**
 - Cette ligne indique le **multi-user.target** et **rescue.target** doivent d'abord être atteints et que les services **rescue.service** et **display-manager.service** doivent d'abord être démarrés,
- **Conflicts=rescue.service rescue.target,**
 - Cette ligne indique la Cible et le service en conflits avec le **graphical.target**,
- **Wants=display-manager.service,**
 - Cette ligne indique quel service doit être démarré.

4.1 - Contrôler les dépendances d'une Cible

Les dépendances d'une Cible peuvent être consultées en utilisant la commande **systemctl list-dependencies** :

```
[root@centos8 ~]# systemctl list-dependencies multi-user.target
multi-user.target
● └─atd.service
● └─auditd.service
● └─avahi-daemon.service
● └─chronyd.service
● └─crond.service
● └─cups.path
```



```
● └─ cups.service
● └─ dbus.service
● └─ dnf-makecache.timer
● └─ firewalld.service
● └─ irqbalance.service
● └─ kdump.service
● └─ ksm.service
● └─ ksmtuned.service
● └─ libstoragemgmt.service
● └─ libvirtd.service
● └─ mcelog.service
● └─ mdmonitor.service
● └─ netcf-transaction.service
● └─ NetworkManager.service
● └─ plymouth-quit-wait.service
● └─ plymouth-quit.service
● └─ rpcbind.service
● └─ rsyslog.service
● └─ smartd.service
● └─ sshd.service
● └─ sssd.service
● └─ systemd-ask-password-wall.path
● └─ systemd-logind.service
● └─ systemd-update-utmp-runlevel.service
● └─ systemd-user-sessions.service
● └─ tuned.service
● └─ vdo.service
● └─ basic.target
●   └─ .mount
●   └─ microcode.service
●   └─ paths.target
●   └─ slices.target
●     └─ .slice
●     └─ system.slice
```

```
● | sockets.target
● |   └─avahi-daemon.socket
● |   └─cups.socket
● |   └─dbus.socket
● |   └─dm-event.socket
● |   └─iscsid.socket
● |   └─iscsiuio.socket
● |   └─libvirt-ro.socket
● |   └─libvirt.socket
● |   └─rpcbind.socket
● |   └─sssd-kcm.socket
● |   └─systemd-coredump.socket
● |   └─systemd-initctl.socket
● |   └─systemd-journald-dev-log.socket
● |   └─systemd-journald.socket
● |   └─systemd-udevd-control.socket
lines 1-57
```

Les points noirs au début de chaque ligne dans la sortie ci-dessus peuvent être de trois couleurs différentes :

- **Vert** implique que le service, la cible ou l'unité est activé et démarré.
- **Blanc** implique que le service, la cible ou l'unité est inactif.
- **Rouge** implique que le service, la cible ou l'unité n'a pas démarré à cause d'une erreur fatale.

Pour visualiser les Unités en état d'erreur fatale, utilisez la commande **systemctl -failed** :

```
[root@centos8 ~]# systemctl --failed
0 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

Les dépendances sont créés sous la forme de liens symboliques dans les répertoires **/etc/systemd/system/multi-user.target.wants** et **/usr/lib/systemd/system/multi-user.target.wants** :

```
[root@centos8 ~]# ls -l /etc/systemd/system/multi-user.target.wants
```

```
total 0
lrwxrwxrwx. 1 root root 35 May  8  2020 atd.service -> /usr/lib/systemd/system/atd.service
lrwxrwxrwx. 1 root root 38 May  8  2020 auditd.service -> /usr/lib/systemd/system/auditd.service
lrwxrwxrwx. 1 root root 44 Jun  3 14:02 avahi-daemon.service -> /usr/lib/systemd/system/avahi-daemon.service
lrwxrwxrwx. 1 root root 39 May  8  2020 chronyd.service -> /usr/lib/systemd/system/chronyd.service
lrwxrwxrwx. 1 root root 37 May  8  2020 crond.service -> /usr/lib/systemd/system/crond.service
lrwxrwxrwx. 1 root root 33 Jun  3 14:02 cups.path -> /usr/lib/systemd/system/cups.path
lrwxrwxrwx. 1 root root 36 Jun  3 14:02 cups.service -> /usr/lib/systemd/system/cups.service
lrwxrwxrwx. 1 root root 43 May  8  2020 dnf-makecache.timer -> /usr/lib/systemd/system/dnf-makecache.timer
lrwxrwxrwx. 1 root root 41 May  8  2020 firewalld.service -> /usr/lib/systemd/system/firewalld.service
lrwxrwxrwx. 1 root root 42 May  8  2020 irqbalance.service -> /usr/lib/systemd/system/irqbalance.service
lrwxrwxrwx. 1 root root 37 May  8  2020 kdump.service -> /usr/lib/systemd/system/kdump.service
lrwxrwxrwx. 1 root root 35 Apr 19 12:07 ksm.service -> /usr/lib/systemd/system/ksm.service
lrwxrwxrwx. 1 root root 40 Apr 19 12:07 ksmtuned.service -> /usr/lib/systemd/system/ksmtuned.service
lrwxrwxrwx. 1 root root 46 May  8  2020 libstoragemgmt.service -> /usr/lib/systemd/system/libstoragemgmt.service
lrwxrwxrwx. 1 root root 40 Apr 19 12:07 libvirtd.service -> /usr/lib/systemd/system/libvirtd.service
lrwxrwxrwx. 1 root root 38 May  8  2020 mcelog.service -> /usr/lib/systemd/system/mcelog.service
lrwxrwxrwx. 1 root root 41 May  8  2020 mdmonitor.service -> /usr/lib/systemd/system/mdmonitor.service
lrwxrwxrwx. 1 root root 49 Apr 19 12:07 netcf-transaction.service -> /usr/lib/systemd/system/netcf-transaction.service
lrwxrwxrwx. 1 root root 46 May  8  2020 NetworkManager.service -> /usr/lib/systemd/system/NetworkManager.service
lrwxrwxrwx. 1 root root 41 Apr 19 12:07 nfs-client.target -> /usr/lib/systemd/system/nfs-client.target
lrwxrwxrwx. 1 root root 40 May  8  2020 remote-fs.target -> /usr/lib/systemd/system/remote-fs.target
lrwxrwxrwx. 1 root root 39 Apr 19 12:07 rpcbind.service -> /usr/lib/systemd/system/rpcbind.service
lrwxrwxrwx. 1 root root 39 May  8  2020 rsyslog.service -> /usr/lib/systemd/system/rsyslog.service
lrwxrwxrwx. 1 root root 38 May  8  2020 smartd.service -> /usr/lib/systemd/system/smartd.service
lrwxrwxrwx. 1 root root 36 May  8  2020 sshd.service -> /usr/lib/systemd/system/sshd.service
lrwxrwxrwx. 1 root root 36 May  8  2020 sssd.service -> /usr/lib/systemd/system/sss.service
lrwxrwxrwx. 1 root root 37 May  8  2020 tuned.service -> /usr/lib/systemd/system/tuned.service
lrwxrwxrwx. 1 root root 35 May  8  2020 vdo.service -> /usr/lib/systemd/system/vdo.service

[root@centos8 ~]# ls -l /usr/lib/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx. 1 root root 15 Apr  7 12:08 dbus.service -> ../dbus.service
```

```
lrwxrwxrwx. 1 root root 15 Apr 7 16:55 getty.target -> ../getty.target
lrwxrwxrwx. 1 root root 24 Oct 6 2020 plymouth-quit.service -> ../plymouth-quit.service
lrwxrwxrwx. 1 root root 29 Oct 6 2020 plymouth-quit-wait.service -> ../plymouth-quit-wait.service
lrwxrwxrwx. 1 root root 33 Apr 7 16:55 systemd-ask-password-wall.path -> ../systemd-ask-password-wall.path
lrwxrwxrwx. 1 root root 25 Apr 7 16:55 systemd-logind.service -> ../systemd-logind.service
lrwxrwxrwx. 1 root root 39 Apr 7 16:55 systemd-update-utmp-runlevel.service -> ../systemd-update-utmp-runlevel.service
lrwxrwxrwx. 1 root root 32 Apr 7 16:55 systemd-user-sessions.service -> ../systemd-user-sessions.service
```

4.2 - La Cible par Défaut

Consulter la Cible par Défaut

Pour consulter la cible par défaut, il convient d'utiliser la commande **systemctl get-default** :

```
[root@centos8 ~]# systemctl get-default
multi-user.target
```

La Cible par défaut est représentée par le lien symbolique **/etc/systemd/system/default.target** :

```
[root@centos8 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37 May 8 2020 /etc/systemd/system/default.target -> /lib/systemd/system/multi-user.target
```

Modifier la Cible par Défaut

Pour modifier la Cible par défaut avec une prise en compte lors du **prochain** démarrage, il convient d'utiliser la commande **systemctl set-default** :

```
[root@centos8 ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
```

```
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
[root@centos8 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40 Jun  6 08:11 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/graphical.target
[root@centos8 ~]# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/multi-user.target.
[root@centos8 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 41 Jun  6 08:11 /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-
user.target
```

Modifier la Cible en Cours

Il est possible de modifier la cible actuellement en cours en utilisant la commande **systemctl isolate** :

```
[root@centos8 ~]# systemctl isolate rescue
[root@centos8 ~]# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
rescue.target          loaded active active Rescue Mode
[root@centos8 ~]# runlevel
3 1
[root@centos8 ~]# who -r
run-level 1  2021-06-15 04:22          last=3
```

```
[root@centos8 ~]# systemctl isolate multi-user
[root@centos8 ~]# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
multi-user.target      loaded active active Multi-User System
[root@centos8 ~]# runlevel
1 3
[root@centos8 ~]# who -r
run-level 3  2021-06-15 04:24          last=1
```

LAB #5 - Gestion des Services

5.1 - Gestion des Instances Uniques

Commencez par installer le paquet **httpd** :

```
[root@centos8 ~]# dnf install httpd
```

Pour obtenir le détail sur un service donné, il convient d'utiliser la commande **systemctl status** :

```
[root@centos8 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
```

Dans le cas du service httpd ci-dessus, on peut constater que le statut est **disabled**. Le statut peut être une de 2 valeurs :

- **disabled** - le service ne démarrera pas lors du prochain démarrage du système.
- **enabled** - le service démarrera lors du prochain démarrage du système.

Il est possible de vérifier le statut en utilisant le commande **systemctl is-enabled** :

```
[root@centos8 ~]# systemctl is-enabled httpd.service
disabled
```

Pour rendre le statut **enabled**, il convient d'utiliser la commande **systemctl enable** :

```
[root@centos8 ~]# systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

```
[root@centos8 ~]# systemctl is-enabled httpd.service
enabled
```

```
[root@centos8 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
```

Dans le cas du service httpd ci-dessus, on peut maintenant constater que l'état est **inactive (dead)**. L'état peut être une de 7 valeurs :

- **inactive (dead)** - le service est arrêté.
- **active(running)** - le service est démarré avec un ou plusieurs processus.
- **active(exited)** - le service a terminé une configuration unique.
- **active(waiting)** - le service est démarré mais en attente d'un évènement.
- **activating** - le service est en cours d'activation.
- **deactivating** - le service est en cours de désactivation.
- **failed** - le service a rencontré une erreur fatale.

Il est possible de vérifier l'état en utilisant la commande **systemctl is-active** :

```
[root@centos8 ~]# systemctl is-active httpd.service
inactive
```

Pour rendre l'état **active(running)**, utilisez la commande suivante :

```
[root@centos8 ~]# systemctl start httpd.service
```

Vérifiez ensuite l'état du service :

```
[root@centos8 ~]# systemctl is-active httpd.service
active
```

```
[root@centos8 ~]# systemctl status httpd.service
```

- `httpd.service` - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)

Active: active (running) since Sun 2021-06-06 15:33:06 EDT; 14s ago

Docs: man:httpd.service(8)

Main PID: 34382 (httpd)

Status: "Running, listening on: port 80"

Tasks: 213 (limit: 23719)

Memory: 36.3M

CGroup: /system.slice/httpd.service

└─34382 /usr/sbin/httpd -DFOREGROUND

└─34383 /usr/sbin/httpd -DFOREGROUND

└─34384 /usr/sbin/httpd -DFOREGROUND

└─34385 /usr/sbin/httpd -DFOREGROUND

└─34386 /usr/sbin/httpd -DFOREGROUND

Jun 06 15:33:05 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...

Jun 06 15:33:06 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.

Jun 06 15:33:06 centos8.ittraining.loc httpd[34382]: Server configured, listening on: port 80

Pour arrêter une Unité de service, utilisez la commande suivante :

```
[root@centos8 ~]# systemctl stop httpd.service
```

```
[root@centos8 ~]# systemctl status httpd.service
```

- `httpd.service` - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)

Active: inactive (dead) since Sun 2021-06-06 23:58:04 EDT; 8s ago

Docs: man:httpd.service(8)

Process: 34382 ExecStart=/usr/sbin/httpd \$OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)

Main PID: 34382 (code=exited, status=0/SUCCESS)

Status: "Running, listening on: port 80"

Jun 06 15:33:05 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...

Jun 06 15:33:06 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.

Jun 06 15:33:06 centos8.ittraining.loc httpd[34382]: Server configured, listening on: port 80


```
Jun 06 23:58:02 centos8.ittraining.loc systemd[1]: Stopping The Apache HTTP Server...
Jun 06 23:58:04 centos8.ittraining.loc systemd[1]: httpd.service: Succeeded.
Jun 06 23:58:04 centos8.ittraining.loc systemd[1]: Stopped The Apache HTTP Server.
```

Pour désactiver un service au prochain démarrage du système, utilisez l'option **disable** :

```
[root@centos8 ~]# systemctl disable httpd.service
Removed /etc/systemd/system/multi-user.target.wants/httpd.service.
[root@centos8 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)

Jun 06 15:33:05 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
Jun 06 15:33:06 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.
Jun 06 15:33:06 centos8.ittraining.loc httpd[34382]: Server configured, listening on: port 80
Jun 06 23:58:02 centos8.ittraining.loc systemd[1]: Stopping The Apache HTTP Server...
Jun 06 23:58:04 centos8.ittraining.loc systemd[1]: httpd.service: Succeeded.
Jun 06 23:58:04 centos8.ittraining.loc systemd[1]: Stopped The Apache HTTP Server.
```

5.2 - Gestion d'Instances Multiples

Systemd permet l'utilisation des gabarits (templates) des fichiers de configuration des Unités. De cette façon il est possible de faire coexister deux ou plusieurs instances du même service. Un gabarit est reconnaissable par le caractère @ qui est placé juste avant le point dans le nom du fichier :

```
[root@centos8 ~]# cat /usr/lib/systemd/system/httpd@.service
# This is a template for httpd instances.
# See httpd@.service(8) for more information.

[Unit]
Description=The Apache HTTP Server
```

```
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd@.service(8)

[Service]
Type=notify
Environment=LANG=C
Environment=HTTPD_INSTANCE=%i
ExecStartPre=/bin/mkdir -m 710 -p /run/httpd/instance-%i
ExecStartPre=/bin/chown root.apache /run/httpd/instance-%i
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND -f conf/%i.conf
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful -f conf/%i.conf
# Send SIGWINCH for graceful stop
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

Une instance créée à partir de ce gabarit devrait avoir un nom sous la forme suivante :

```
httpd@<nom_instance>.service
```

Dans ce fichier on peut constater l'utilisation d'un **identifiant** sous la forme de **%i**. Les identifiants sont de deux types - un dit **échappé** où les caractères non-ASCII alphanumérique sont remplacés par **escapes** de type langage C et l'autre non-échappé :

- **%n** : est remplacé par le nom complet échappé de l'Unité.
- **%N** : est remplacé par le nom complet non-échappé de l'Unité.
- **%p** : est remplacé par le préfixe échappé de l'Unité, c'est-à-dire la partie **avant** le caractère @.
- **%P** : est remplacé par le préfixe non-échappé de l'Unité, c'est-à-dire la partie **avant** le caractère @.
- **%i** : est remplacé par le nom de l'instance échappé de l'Unité, c'est-à-dire la partie **après** le caractère @ et **avant** le point.
- **%I** : est remplacé par le nom de l'instance non-échappé de l'Unité, c'est-à-dire la partie **après** le caractère @ et **avant** le point.
- **%f** : est remplacé par le préfixe non-échappé ou le nom de l'instance non-échappé préfixé par le caractère /.
- **%c** : est remplacé par le CGroup de l'Unité sans le chemin /sys/fs/cgroup/systemd/.

- %u : est remplacé par le nom de l'utilisateur responsable de l'exécution de l'Unité.
- %U : est remplacé par l'UID de l'utilisateur responsable de l'exécution de l'Unité.
- %H : est remplacé par le nom d'hôte sur lequel est exécuté l'Unité.
- %% : est remplacé par le caractère %.

Créez maintenant deux copies du fichier **/usr/lib/systemd/system/httpd@.service** :

```
[root@centos8 ~]# cp /usr/lib/systemd/system/httpd@.service /usr/lib/systemd/system/httpd@instance01.service
[root@centos8 ~]# cp /usr/lib/systemd/system/httpd@.service /usr/lib/systemd/system/httpd@instance02.service
```

Créez deux copies du fichier **/etc/httpd/conf/httpd.conf** :

```
[root@centos8 ~]# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/instance01.conf
[root@centos8 ~]# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/instance02.conf
```

Editez la directive **Listen** du fichier **/etc/httpd/conf/instance01.conf** et ajoutez la directive **PidFile** :

```
[root@centos8 ~]# vi /etc/httpd/conf/instance01.conf
[root@centos8 ~]# more /etc/httpd/conf/instance01.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
```

```
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.
```

```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
```

```
#
ServerRoot "/etc/httpd"
```

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
```

```
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
```

```
#
#Listen 12.34.56.78:80
Listen 8008
PidFile /var/run/httpd/instance01.pid
```

```
#
# Dynamic Shared Object (DSO) Support
```

```
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
--More-- (19%)
```

Editez la directive **Listen** du fichier **/etc/httpd/conf/instance02.conf** et ajoutez la directive **PidFile** :

```
[root@centos8 ~]# vi /etc/httpd/conf/instance02.conf
[root@centos8 ~]# more /etc/httpd/conf/instance02.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
```

```
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8009
PidFile /var/run/httpd/instance02.pid
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
```

```
# to be loaded here.  
#  
# Example:  
# LoadModule foo_module modules/mod_foo.so  
--More-- (19%)
```

Démarrez les deux services :

```
[root@centos8 ~]# systemctl start httpd@instance01.service  
[root@centos8 ~]# systemctl status httpd@instance01.service  
● httpd@instance01.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd@instance01.service; disabled; vendor preset: disabled)  
   Active: active (running) since Mon 2021-06-07 01:40:43 EDT; 7s ago  
     Docs: man:httpd@.service(8)  
  Process: 43854 ExecStartPre=/bin/chown root.apache /run/httpd/instance-instance01 (code=exited,  
status=0/SUCCESS)  
  Process: 43852 ExecStartPre=/bin/mkdir -m 710 -p /run/httpd/instance-instance01 (code=exited, status=0/SUCCESS)  
 Main PID: 43856 (httpd)  
   Status: "Started, listening on: port 8008"  
    Tasks: 213 (limit: 23719)  
  Memory: 43.6M  
   CGroup: /system.slice/system-httpd.slice/httpd@instance01.service  
           └─43856 /usr/sbin/httpd -DFOREGROUND -f conf/instance01.conf  
             └─43857 /usr/sbin/httpd -DFOREGROUND -f conf/instance01.conf  
               └─43858 /usr/sbin/httpd -DFOREGROUND -f conf/instance01.conf  
                 └─43859 /usr/sbin/httpd -DFOREGROUND -f conf/instance01.conf  
                   └─43860 /usr/sbin/httpd -DFOREGROUND -f conf/instance01.conf  
  
Jun 07 01:40:43 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...  
Jun 07 01:40:43 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.  
Jun 07 01:40:43 centos8.ittraining.loc httpd[43856]: Server configured, listening on: port 8008  
[root@centos8 ~]# systemctl start httpd@instance02.service  
[root@centos8 ~]# systemctl status httpd@instance02.service  
● httpd@instance02.service - The Apache HTTP Server
```

```
Loaded: loaded (/usr/lib/systemd/system/httpd@instance02.service; disabled; vendor preset: disabled)
Active: active (running) since Mon 2021-06-07 01:36:45 EDT; 4min 24s ago
Docs: man:httpd@.service(8)
Process: 43568 ExecStartPre=/bin/chown root.apache /run/httpd/instance-instance02 (code=exited,
status=0/SUCCESS)
Process: 43566 ExecStartPre=/bin/mkdir -m 710 -p /run/httpd/instance-instance02 (code=exited, status=0/SUCCESS)
Main PID: 43569 (httpd)
Status: "Running, listening on: port 8009"
Tasks: 213 (limit: 23719)
Memory: 43.6M
CGroup: /system.slice/system-httpd.slice/httpd@instance02.service
├─43569 /usr/sbin/httpd -DFOREGROUND -f conf/instance02.conf
├─43571 /usr/sbin/httpd -DFOREGROUND -f conf/instance02.conf
├─43572 /usr/sbin/httpd -DFOREGROUND -f conf/instance02.conf
├─43573 /usr/sbin/httpd -DFOREGROUND -f conf/instance02.conf
└─43574 /usr/sbin/httpd -DFOREGROUND -f conf/instance02.conf

Jun 07 01:36:45 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
Jun 07 01:36:45 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.
Jun 07 01:36:45 centos8.ittraining.loc httpd[43569]: Server configured, listening on: port 8009
```

5.3 - Interdire la Modification du Statut d'un Service

Il est possible d'interdire la modification en utilisant la commande **systemctl mask**:

```
[root@centos8 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
  Docs: man:httpd.service(8)

Jun 07 18:27:25 centos8.ittraining.loc httpd[58535]: Server configured, listening on: port 80
Jun 07 18:27:29 centos8.ittraining.loc systemd[1]: Stopping The Apache HTTP Server...
```



```
Jun 07 18:27:30 centos8.ittraining.loc systemd[1]: httpd.service: Succeeded.  
Jun 07 18:27:30 centos8.ittraining.loc systemd[1]: Stopped The Apache HTTP Server.  
Jun 07 18:27:32 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...  
Jun 07 18:27:32 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.  
Jun 07 18:27:32 centos8.ittraining.loc httpd[58760]: Server configured, listening on: port 80  
Jun 07 18:27:34 centos8.ittraining.loc systemd[1]: Stopping The Apache HTTP Server...  
Jun 07 18:27:36 centos8.ittraining.loc systemd[1]: httpd.service: Succeeded.  
Jun 07 18:27:36 centos8.ittraining.loc systemd[1]: Stopped The Apache HTTP Server.
```

```
[root@centos8 ~]# systemctl mask httpd.service  
Created symlink /etc/systemd/system/httpd.service → /dev/null.
```

```
[root@centos8 ~]# systemctl enable httpd.service  
Failed to enable unit: Unit file /etc/systemd/system/httpd.service is masked.
```

```
[root@centos8 ~]# systemctl start httpd.service  
Failed to start httpd.service: Unit httpd.service is masked.
```

Pour autoriser de nouveau les modifications, il convient d'utiliser la commande **systemctl unmask** :

```
[root@centos8 ~]# systemctl unmask httpd.service  
Removed /etc/systemd/system/httpd.service.
```

```
[root@centos8 ~]# systemctl enable httpd.service  
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →  
/usr/lib/systemd/system/httpd.service.
```

```
[root@centos8 ~]# systemctl start httpd.service
```

```
[root@centos8 ~]# systemctl status httpd.service
```

```
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)  
   Active: active (running) since Mon 2021-06-07 18:30:59 EDT; 5s ago  
     Docs: man:httpd.service(8)
```

```
Main PID: 59101 (httpd)
  Status: "Started, listening on: port 80"
  Tasks: 213 (limit: 23719)
  Memory: 39.4M
  CGroup: /system.slice/httpd.service
          └─59101 /usr/sbin/httpd -DFOREGROUND
          └─59102 /usr/sbin/httpd -DFOREGROUND
          └─59103 /usr/sbin/httpd -DFOREGROUND
          └─59104 /usr/sbin/httpd -DFOREGROUND
          └─59105 /usr/sbin/httpd -DFOREGROUND
```

```
Jun 07 18:30:59 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
```

```
Jun 07 18:30:59 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.
```

```
Jun 07 18:31:00 centos8.ittraining.loc httpd[59101]: Server configured, listening on: port 80
```

Arrêt du Système

Sous RHEL / CentOS 8 les commandes **halt**, **poweroff**, **reboot** et **shutdown** sont des liens symboliques vers **/bin/systemctl** :

```
[root@centos8 ~]# ls -l /usr/sbin/shutdown /usr/sbin/halt /usr/sbin/poweroff /usr/sbin/reboot
lrwxrwxrwx. 1 root root 16 Apr  7 16:55 /usr/sbin/halt -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Apr  7 16:55 /usr/sbin/poweroff -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Apr  7 16:55 /usr/sbin/reboot -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Apr  7 16:55 /usr/sbin/shutdown -> ../bin/systemctl
```

La manière recommandée d'utiliser ces commandes est donc :

- systemctl halt
- systemctl poweroff
- systemctl reboot
- systemctl shutdown

Il est cependant toujours possibles d'utiliser les commandes **halt**, **poweroff**, **reboot** et **shutdown**.

La Commande shutdown

Lors de l'arrêt de la machine, Linux procède, entre autre, aux tâches suivantes :

- Il prévient les utilisateurs,
- Il arrête tous les services,
- Il inscrit toutes les données sur disque,
- Il démonte les systèmes de fichiers.

La commande utilisée pour arrêter le système est la commande **shutdown** :

```
shutdown [-t sec] [-HPrhkc] heure [message]
```

Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# shutdown --help
shutdown [OPTIONS...] [TIME] [WALL...]
```

Shut down the system.

--help	Show this help
-H --halt	Halt the machine
-P --poweroff	Power-off the machine
-r --reboot	Reboot the machine
-h	Equivalent to --poweroff, overridden by --halt
-k	Don't halt/power-off/reboot, just send warnings
--no-wall	Don't send wall message before halt/power-off/reboot

-c Cancel a pending shutdown

L'option **heure** peut prendre plusieurs valeurs :

Valeur	Description
hh:mm	L'heure à laquelle l'opération aura lieu
+m	Nombre de minutes avant que l'opération aura lieu. Si aucune valeur n'est spécifiée, la valeur par défaut est +1
now	L'opération est immédiate. now est un alias de +0



Important : Si l'opération est programmée pour dans moins de 5 minutes, les connexions supplémentaires sont interdites, y comprises les tentatives de connexion de root.

L'option **-t** est utilisée pour accorder un temps de grâce :

```
[root@centos8 ~]# date && shutdown -t 60 -k
Mon Jun  7 18:58:26 EDT 2021
Shutdown scheduled for Mon 2021-06-07 18:59:26 EDT, use 'shutdown -c' to cancel.
```

La Commande reboot

Cette commande redémarre le système.

Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# reboot --help
reboot [OPTIONS...] [ARG]
```

Reboot the system.

```
--help      Show this help
--halt      Halt the machine
-p --poweroff Switch off the machine
--reboot     Reboot the machine
-f --force   Force immediate halt/power-off/reboot
-w --wtmp-only Don't halt/power-off/reboot, just write wtmp record
-d --no-wtmp Don't write wtmp record
--no-wall    Don't send wall message before halt/power-off/reboot
```

La Commande halt

Cette commande arrête le système.

Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# halt --help
halt [OPTIONS...]
```

Halt the system.

```
--help      Show this help
--halt      Halt the machine
-p --poweroff Switch off the machine
--reboot     Reboot the machine
-f --force   Force immediate halt/power-off/reboot
-w --wtmp-only Don't halt/power-off/reboot, just write wtmp record
-d --no-wtmp Don't write wtmp record
```

```
--no-wall    Don't send wall message before halt/power-off/reboot
```

La Commande **poweroff**

Cette commande arrête le système et coupe l'alimentation électrique.

Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# poweroff --help
poweroff [OPTIONS...]
```

Power off the system.

```
--help      Show this help
--halt      Halt the machine
-p --poweroff Switch off the machine
--reboot     Reboot the machine
-f --force   Force immediate halt/power-off/reboot
-w --wtmp-only Don't halt/power-off/reboot, just write wtmp record
-d --no-wtmp Don't write wtmp record
--no-wall    Don't send wall message before halt/power-off/reboot
```