

Version : **2024.01**

Dernière mise-à-jour : 2024/10/24 09:58

RH13406 - Gestion du Démarrage et de l'Arrêt du Système

Contenu du Module

- **RH13406 - 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
- 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 Charger 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@redhat9 ~]# 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
}
```

```
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_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/08_fallback_counting ###
insmod increment
# Check if boot_counter exists and boot_success=0 to activate this behaviour.
if [ -n "${boot_counter}" -a "${boot_success}" = "0" ]; then
    # if countdown has ended, choose to boot rollback deployment,
    # i.e. default=1 on OSTree-based systems.
    if [ "${boot_counter}" = "0" -o "${boot_counter}" = "-1" ]; then
        set default=1
        set boot_counter=-1
    # otherwise decrement boot_counter
    else
        decrement boot_counter
    fi
    save_env boot_counter
fi
### END /etc/grub.d/08_fallback_counting ###

### BEGIN /etc/grub.d/10_linux ###
insmod part_msdos
insmod xfs
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' 6f6c5bb9-30be-4734-bc23-03fed8541616
else
    search --no-floppy --fs-uuid --set=root 6f6c5bb9-30be-4734-bc23-03fed8541616
fi
```

```
insmod part_msdos
insmod xfs
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' 6f6c5bb9-30be-4734-bc23-03fed8541616
else
    search --no-floppy --fs-uuid --set=boot 6f6c5bb9-30be-4734-bc23-03fed8541616
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://systemd.io/BOOT\_LOADER\_SPECIFICATION/.

# The kernelopts variable should be defined in the grubenv file. But to ensure that menu
# entries populated from BootLoaderSpec files that use this variable work correctly even
# without a grubenv file, define a fallback kernelopts variable if this has not been set.
#
# The kernelopts variable in the grubenv file can be modified using the grubby tool or by
# executing the grub2-mkconfig tool. For the latter, the values of the GRUB_CMDLINE_LINUX
# and GRUB_CMDLINE_LINUX_DEFAULT options from /etc/default/grub file are used to set both
# the kernelopts variable in the grubenv file and the fallback kernelopts variable.
if [ -z "${kernelopts}" ]; then
    set kernelopts="root=/dev/mapper/rhel-root ro crashkernel=1G-4G:192M,4G-64G:256M,64G-:512M
resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet "
fi

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

```
### BEGIN /etc/grub.d/10_reset_boot_success ###
# Hiding the menu is ok if last boot was ok or if this is a first boot attempt to boot the entry
if [ "${boot_success}" = "1" -o "${boot_indeterminate}" = "1" ]; then
    set menu_hide_ok=1
else
    set menu_hide_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
# Reset boot_success for current boot
set boot_success=0
save_env boot_success boot_indeterminate
### END /etc/grub.d/10_reset_boot_success ###

### BEGIN /etc/grub.d/12_menu_auto_hide ###
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 [ "${menu_auto_hide}" -a "${menu_hide_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
### END /etc/grub.d/12_menu_auto_hide ###

### BEGIN /etc/grub.d/14_menu_show_once ###
if [ x$feature_timeout_style = xy ]; then
    if [ "${menu_show_once_timeout}" ]; then
        set timeout_style=menu
        set timeout="${menu_show_once_timeout}"
        unset menu_show_once_timeout
        save_env menu_show_once_timeout
    fi
fi
### END /etc/grub.d/14_menu_show_once ###

### 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 ###
if [ "$grub_platform" = "efi" ]; then
    menuentry 'UEFI Firmware Settings' $menuentry_id_option 'uefi-firmware' {
        fwsetup
    }
fi
### END /etc/grub.d/30_uefi-firmware ###
```

```
### BEGIN /etc/grub.d/35_fwupd ###
### END /etc/grub.d/35_fwupd ###

### 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@redhat9 ~]# cat /boot/grub2/device.map
# this device map was generated by anaconda
(hd0)      /dev/sda
```

Le fichier `/etc/default/grub`

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

```
[root@redhat9 ~]# 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=1G-4G:192M,4G-64G:256M,64G-:512M resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root
rd.lvm.lv=rhel/swap 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@redhat9 ~]# ls -l /etc/grub.d
total 108
-rwxr-xr-x. 1 root root 9346 Jun 27 12:12 00_header
-rwxr-xr-x. 1 root root 1046 Feb 22 2024 00_tuned
-rwxr-xr-x. 1 root root 236 Jun 27 12:12 01_users
-rwxr-xr-x. 1 root root 835 Jun 27 12:12 08_fallback_counting
-rwxr-xr-x. 1 root root 19665 Jun 27 12:12 10_linux
-rwxr-xr-x. 1 root root 833 Jun 27 12:12 10_reset_boot_success
-rwxr-xr-x. 1 root root 892 Jun 27 12:12 12_menu_auto_hide
-rwxr-xr-x. 1 root root 410 Jun 27 12:12 14_menu_show_once
-rwxr-xr-x. 1 root root 13613 Jun 27 12:12 20_linux_xen
-rwxr-xr-x. 1 root root 2562 Jun 27 12:12 20_ppc_terminfo
-rwxr-xr-x. 1 root root 10869 Jun 27 12:12 30_os-prober
-rwxr-xr-x. 1 root root 1122 Jun 27 12:12 30_uefi-firmware
-rwxr-xr-x. 1 root root 725 Feb 19 2024 35_fwupd
-rwxr-xr-x. 1 root root 218 Jun 27 12:12 40_custom
-rwxr-xr-x. 1 root root 219 Jun 27 12:12 41_custom
-rw-r--r--. 1 root root 483 Jun 27 12:12 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`

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 BTRFS.

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@redhat9 ~]# cp /boot/initramfs-`uname -r`.img /tmp/custom  
  
[root@redhat9 ~]# cd /tmp  
  
[root@redhat9 tmp]# ls custom  
custom
```

Examiner l'image existante

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

```
[root@redhat9 tmp]# lsinitrd custom | more
```

Image: custom: 54M

=====

Early CPIO image

=====

drwxr-xr-x	3	root	root	0	Jan	4	2024	.
-rw-r--r--	1	root	root	2	Jan	4	2024	early_cpio
drwxr-xr-x	3	root	root	0	Jan	4	2024	kernel
drwxr-xr-x	3	root	root	0	Jan	4	2024	kernel/x86
drwxr-xr-x	2	root	root	0	Jan	4	2024	kernel/x86/microcode
-rw-r--r--	1	root	root	28672	Jan	4	2024	kernel/x86/microcode/GenuineIntel.bin

=====

Version: dracut-057-53.git20240104.el9

Arguments: -f -v

dracut modules:

bash
systemd
systemd-initrd
systemd-sysusers
nss-softokn
dbus-broker
dbus
i18n
network-manager
network
ifcfg
drm
plymouth
prefixdevname
dm
kernel-modules
kernel-modules-extra
kernel-network-modules

```
lvm
qemu
qemu-net
resume
rootfs-block
terminfo
udev-rules
virtiofs
dracut-systemd
usrmount
base
fs-lib
microcode_ctl-fw_dir_override
shutdown
```

```
=====
drwxr-xr-x 12 root    root          0 Jan  4  2024 .
crw-r--r--  1 root    root          5,  1 Jan  4  2024 dev/console
crw-r--r--  1 root    root          1, 11 Jan  4  2024 dev/kmsg
crw-r--r--  1 root    root          1,  3 Jan  4  2024 dev/null
crw-r--r--  1 root    root          1,  8 Jan  4  2024 dev/random
crw-r--r--  1 root    root          1,  9 Jan  4  2024 dev/urandom
--More--
```

Le script init

Red Hat 9 utilise le système de démarrage **systemd**. Pour cette raison le script init est un lien symbolique vers **/usr/lib/systemd/systemd** :

```
[root@redhat9 tmp]# lsinitrd custom | grep usr/lib/systemd/systemd | grep init
lrwxrwxrwx  1 root    root          23 Jan  4  2024 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@redhat9 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@redhat9 tmp]# ls -l /usr/lib/systemd/system/runlevel*
lrwxrwxrwx. 1 root root 15 Jul 18 13:00 /usr/lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx. 1 root root 13 Jul 18 13:00 /usr/lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx. 1 root root 17 Jul 18 13:00 /usr/lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Jul 18 13:00 /usr/lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx. 1 root root 17 Jul 18 13:00 /usr/lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx. 1 root root 16 Jul 18 13:00 /usr/lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 Jul 18 13:00 /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@redhat9 tmp]# 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-host2-target2:0:0-2:0:0:0-block-sr0.device	loaded
active plugged QEMU_DVD-ROM	
sys-devices-pci0000:00-0000:00:05.0-0000:01:01.0-virtio2-host0-target0:0:0-0:0:0:0-block-sda-sda1.device	loaded
active plugged QEMU_HARDDISK 1	
sys-devices-pci0000:00-0000:00:05.0-0000:01:01.0-virtio2-host0-target0:0:0-0:0:0:0-block-sda-sda2.device	loaded
active plugged QEMU_HARDDISK 2	
sys-devices-pci0000:00-0000:00:05.0-0000:01:01.0-virtio2-host0-target0:0:0-0:0:0:0-block-sda.device	loaded
active plugged QEMU_HARDDISK	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb1.device	loaded
active plugged QEMU_HARDDISK 1	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb10.device	loaded
active plugged QEMU_HARDDISK redhat.ittraining.loc	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb11.device	loaded
active plugged QEMU_HARDDISK my_ext4	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb12.device	loaded
active plugged QEMU_HARDDISK 12	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb2.device	loaded
active plugged QEMU_HARDDISK 2	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb3.device	loaded
active plugged QEMU_HARDDISK 3	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb4.device	loaded
active plugged QEMU_HARDDISK 4	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb5.device	loaded
active plugged QEMU_HARDDISK redhat.ittraining.loc	

sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb6.device	loaded
active plugged QEMU_HARDDISK 6	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb7.device	loaded
active plugged QEMU_HARDDISK 7	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb8.device	loaded
active plugged QEMU_HARDDISK redhat.ittraining.loc	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb9.device	loaded
active plugged QEMU_HARDDISK 9	
sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb.device	loaded
active plugged QEMU_HARDDISK	
sys-devices-pci0000:00-0000:00:07.0-ata4-host4-target4:0:0-4:0:0:0-block-sdc.device	loaded
active plugged QEMU_HARDDISK	
sys-devices-pci0000:00-0000:00:07.0-ata5-host5-target5:0:0-5:0:0:0-block-sdd.device	loaded
active plugged QEMU_HARDDISK	
sys-devices-pci0000:00-0000:00:12.0-virtio1-net-ens18.device	loaded
active plugged Virtio network device	
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-block-dm\x2d2.device	loaded
active plugged /sys/devices/virtual/block/dm-2	
sys-devices-virtual-block-dm\x2d3.device	loaded
active plugged /sys/devices/virtual/block/dm-3	
sys-devices-virtual-block-md1.device	loaded
active plugged /sys/devices/virtual/block/md1	

sys-devices-virtual-misc-rfkill.device	loaded
active plugged /sys/devices/virtual/misc/rfkill	
sys-module-configfs.device	loaded
active plugged /sys/module/configfs	
sys-module-fuse.device	loaded
active plugged /sys/module/fuse	
sys-subsystem-net-devices-ensl8.device	loaded
active plugged Virtio network device	
-.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-credentials-systemd\x2dsysctl.service.mount	loaded
active mounted /run/credentials/systemd-sysctl.service	
run-credentials-systemd\x2dtmpfiles\x2dsetup.service.mount	loaded
active mounted /run/credentials/systemd-tmpfiles-setup.service	
run-credentials-systemd\x2dtmpfiles\x2dsetup\x2ddev.service.mount	loaded
active mounted /run/credentials/systemd-tmpfiles-setup-dev.service	
run-user-1000.mount	loaded
active mounted /run/user/1000	
run-user-42.mount	loaded
active mounted /run/user/42	
sys-fs-fuse-connections.mount	loaded
active mounted FUSE Control File System	
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 Kernel Trace File System	

```
 cups.path                                loaded
active running    CUPS Scheduler
  systemd-ask-password-wall.path          loaded
active running    Forward Password Requests to Wall Directory Watch
  init.scope                                loaded
active running    System and Service Manager
  session-4.scope                          loaded
active running    Session 4 of User trainee
  session-c1.scope                        loaded
active running    Session c1 of User gdm
  accounts-daemon.service                 loaded
active running    Accounts Service
  atd.service                             loaded
active running    Deferred execution scheduler
lines 1-55
```

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

```
[root@redhat9 tmp]# systemctl list-units --all | grep inactive | more
● boot.automount
not-found inactive dead    boot.automount
  dev-mapper-sdb12.device
loaded    inactive dead    /dev/mapper/sdb12
● home.mount
not-found inactive dead    home.mount
  mnt-sdb12.mount
loaded    inactive dead    /mnt/sdb12
  proc-sys-fs-binfmt_misc.mount
loaded    inactive dead    Arbitrary Executable File Formats File System
● sysroot.mount
not-found inactive dead    sysroot.mount
  tmp.mount
loaded    inactive dead    Temporary Directory /tmp
● var.mount
```

```
not-found inactive dead    var.mount
systemd-ask-password-console.path
loaded    inactive dead    Dispatch Password Requests to Console Directory Watch
systemd-ask-password-plymouth.path
loaded    inactive dead    Forward Password Requests to Plymouth Directory Watch
alsa-restore.service
loaded    inactive dead    Save/Restore Sound Card State
alsa-state.service
loaded    inactive dead    Manage Sound Card State (restore and store)
● auto-cpufreq.service
not-found inactive dead    auto-cpufreq.service
● autofs.service
not-found inactive dead    autofs.service
blk-availability.service
loaded    inactive dead    Availability of block devices
cpupower.service
loaded    inactive dead    Configure CPU power related settings
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
dracut-shutdown-onfailure.service
loaded    inactive dead    Service executing upon dracut-shutdown failure to perform cleanup
● ebtables.service
not-found inactive dead    ebtables.service
emergency.service
loaded    inactive dead    Emergency Shell
● fcoe.service
not-found inactive dead    fcoe.service
getty@tty1.service
loaded    inactive dead    Getty on tty1
● httpd-init.service
not-found inactive dead    httpd-init.service
initrd-cleanup.service
loaded    inactive dead    Cleaning Up and Shutting Down Daemons
initrd-parse-etc.service
loaded    inactive dead    Mountpoints Configured in the Real Root
initrd-switch-root.service
loaded    inactive dead    Switch Root
initrd-udevadm-cleanup-db.service
loaded    inactive dead    Cleanup udev Database
insights-client-boot.service
loaded    inactive dead    Run Insights Client at boot
● 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-init.service
loaded    inactive dead    One time configuration for iscsi.service
iscsi-onboot.service
loaded    inactive dead    Special handling of early boot iSCSI sessions
iscsi-shutdown.service
```

```
loaded inactive dead Logout off all iSCSI sessions on shutdown
iscsi-starter.service
loaded inactive dead iscsi-starter.service
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
logrotate.service
loaded inactive dead Rotate log files
low-memory-monitor.service
loaded inactive dead Low Memory Monitor
● lvm2-activation-early.service
not-found inactive dead lvm2-activation-early.service
lvm2-lvmpolld.service
loaded inactive dead LVM2 poll daemon
mdadm-last-resort@md1.service
loaded inactive dead Activate md array md1 even though degraded
microcode.service
loaded inactive dead Load CPU microcode update
mlocate-updatedb.service
loaded inactive dead Update a database for mlocate
modprobe@configfs.service
loaded inactive dead Load Kernel Module configfs
modprobe@drm.service
loaded inactive dead Load Kernel Module drm
--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@redhat9 tmp]# systemctl status ip6tables
```

Unit `ip6tables.service` could not be found.

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

```
[root@redhat9 tmp]# systemctl list-unit-files | more
UNIT FILE                                STATE      PRESET
proc-sys-fs-binfmt_misc.automount       static     -
-.mount                                  generated -
boot.mount                               generated -
dev-hugepages.mount                     static     -
dev-mqueue.mount                        static     -
mnt-sdb12.mount                         generated -
proc-sys-fs-binfmt_misc.mount            disabled   disabled
run-vmblock\x2dfuse.mount               disabled   disabled
sys-fs-fuse-connections.mount           static     -
sys-kernel-config.mount                 static     -
sys-kernel-debug.mount                 static     -
sys-kernel-tracing.mount                static     -
tmp.mount                                disabled   disabled
cups.path                                enabled    enabled
insights-client-results.path            disabled   disabled
ostree-finalize-staged.path             disabled   disabled
systemd-ask-password-console.path       static     -
systemd-ask-password-plymouth.path       static     -
systemd-ask-password-wall.path          static     -
session-4.scope                         transient  -
session-cl.scope                       transient  -
accounts-daemon.service                 enabled    enabled
alsa-restore.service                   static     -
alsa-state.service                     static     -
arp-ethers.service                     disabled   disabled
atd.service                             enabled    enabled
auditd.service                         enabled    enabled
autovt@.service                         alias      -
```

avahi-daemon.service	enabled	enabled
bar.service	enabled	disabled
blk-availability.service	disabled	disabled
bluetooth.service	enabled	enabled
bolt.service	static	-
brltty.service	disabled	disabled
canberra-system-bootup.service	disabled	disabled
canberra-system-shutdown-reboot.service	disabled	disabled
canberra-system-shutdown.service	disabled	disabled
chrony-wait.service	disabled	disabled
chronyd-restricted.service	disabled	disabled
chronyd.service	disabled	enabled
cni-dhcp.service	disabled	disabled
cockpit-motd.service	static	-
cockpit-wsinstance-http.service	static	-
cockpit-wsinstance-https-factory@.service	static	-
cockpit-wsinstance-https@.service	static	-
cockpit.service	static	-
colord.service	static	-
configure-printer@.service	static	-
console-getty.service	disabled	disabled
container-getty@.service	static	-
cpupower.service	disabled	disabled
crond.service	enabled	enabled
cups-browsed.service	disabled	disabled
cups.service	enabled	enabled
--More--		

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

```
[root@redhat9 tmp]# systemctl list-unit-files -t mount
UNIT FILE          STATE    PRESET
-.mount            generated -
boot.mount         generated -
```

```
dev-hugepages.mount      static -
dev-mqueue.mount         static -
mnt-sdb12.mount          generated -
proc-sys-fs-binfmt_misc.mount disabled disabled
run-vmblock\x2dfuse.mount disabled disabled
sys-fs-fuse-connections.mount static -
sys-kernel-config.mount  static -
sys-kernel-debug.mount   static -
sys-kernel-tracing.mount static -
tmp.mount                disabled disabled

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/system-generators/systemd-fstab-generator** :

```
[root@redhat9 tmp]# ls -l /lib/systemd/system-generators/systemd-fstab-generator
-rwxr-xr-x. 1 root root 57696 Jul 18 13:01 /lib/systemd/system-generators/systemd-fstab-generator
```

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

```
[root@redhat9 tmp]# ls -l /lib/systemd/system-generators
total 396
-rwxr-xr-x. 1 root root 541 Jul 24 06:08 kdump-dep-generator.sh
-rwxr-xr-x. 1 root root 15832 May 17 18:27 ostree-system-generator
lrwxrwxrwx. 1 root root 31 Aug 8 10:54 podman-system-generator -> ../../../../libexec/podman/quadlet
-rwxr-xr-x. 1 root root 1005 Feb 19 2024 selinux-autorelabel-generator.sh
```

```
-rwxr-xr-x. 1 root root 15624 Jul 18 13:01 systemd-bless-boot-generator
-rwxr-xr-x. 1 root root 40920 Jul 18 13:01 systemd-cryptsetup-generator
-rwxr-xr-x. 1 root root 24312 Jul 18 13:01 systemd-debug-generator
-rwxr-xr-x. 1 root root 57696 Jul 18 13:01 systemd-fstab-generator
-rwxr-xr-x. 1 root root 24096 Jul 18 13:01 systemd-getty-generator
-rwxr-xr-x. 1 root root 36496 Jul 18 13:01 systemd-gpt-auto-generator
-rwxr-xr-x. 1 root root 16096 Jul 18 13:01 systemd-hibernate-resume-generator
-rwxr-xr-x. 1 root root 24240 Jul 18 13:01 systemd-integritysetup-generator
-rwxr-xr-x. 1 root root 15632 Jul 18 13:01 systemd-rc-local-generator
-rwxr-xr-x. 1 root root 24296 Jul 18 13:01 systemd-run-generator
-rwxr-xr-x. 1 root root 15824 Jul 18 13:01 systemd-system-update-generator
-rwxr-xr-x. 1 root root 36424 Jul 18 13:01 systemd-sysv-generator
-rwxr-xr-x. 1 root root 36872 Jul 18 13:01 systemd-veritysetup-generator
```

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

```
[root@redhat9 tmp]# systemctl --help
systemctl [OPTIONS...] COMMAND ...
```

Query or send control commands to the system manager.

Unit Commands:

list-units [PATTERN...]	List units currently in memory
list-automounts [PATTERN...]	List automount units currently in memory, ordered by path
list-sockets [PATTERN...]	List socket units currently in memory, ordered by address
list-timers [PATTERN...]	List timer units currently in memory, ordered by next elapse
is-active PATTERN...	Check whether units are active
is-failed PATTERN...	Check whether units are failed
status [PATTERN... PID...]	Show runtime status of one or more units
show [PATTERN... JOB...]	Show properties of one or more units/jobs or the manager

cat PATTERN...	Show files and drop-ins of specified units
help PATTERN... PID...	Show manual for one or more units
list-dependencies [UNIT...]	Recursively show units which are required or wanted by the units or by which those units are required or wanted
start UNIT...	Start (activate) one or more units
stop UNIT...	Stop (deactivate) one or more units
reload UNIT...	Reload one or more units
restart UNIT...	Start or restart one or more units
try-restart UNIT...	Restart one or more units if active
reload-or-restart UNIT...	Reload one or more units if possible, otherwise start or restart
try-reload-or-restart UNIT...	If active, reload one or more units, if supported, otherwise restart
isolate UNIT	Start one unit and stop all others
kill UNIT...	Send signal to processes of a unit
clean UNIT...	Clean runtime, cache, state, logs or configuration of unit
freeze PATTERN...	Freeze execution of unit processes
thaw PATTERN...	Resume execution of a frozen unit
set-property UNIT PROPERTY=VALUE...	Sets one or more properties of a unit
bind UNIT PATH [PATH]	Bind-mount a path from the host into a unit's namespace
mount-image UNIT PATH [PATH [OPTS]]	Mount an image from the host into a unit's namespace
service-log-level SERVICE [LEVEL]	Get/set logging threshold for service
service-log-target SERVICE [TARGET]	Get/set logging target for service
reset-failed [PATTERN...]	Reset failed state for all, one, or more units
Unit File Commands:	
list-unit-files [PATTERN...]	List installed unit files
enable [UNIT... PATH...]	Enable one or more unit files
disable UNIT...	Disable one or more unit files
reenable UNIT...	Reenable one or more unit files

preset UNIT...	Enable/disable one or more unit files based on preset configuration
preset-all	Enable/disable all unit files based on preset configuration

lines 1-55

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@redhat9 tmp]# pkg-config systemd --variable=systemdsystemunitdir  
/usr/lib/systemd/system
```

```
[root@redhat9 tmp]# ls -l /usr/lib/systemd/system | more  
total 1640  
-rw-r--r--. 1 root root 729 Feb 24 2022 accounts-daemon.service  
-rw-r--r--. 1 root root 480 Jan 16 2024 alsa-restore.service  
-rw-r--r--. 1 root root 465 Jan 16 2024 alsa-state.service  
-rw-r--r--. 1 root root 275 Aug 10 2021 arp-ethers.service  
-rw-r--r--. 1 root root 274 Apr 4 2022 atd.service  
-rw-r--r--. 1 root root 1771 Nov 8 2023 auditd.service  
lrwxrwxrwx. 1 root root 14 Jul 18 13:00 autovt@.service -> getty@.service  
-rw-r--r--. 1 root root 1044 Nov 8 2023 avahi-daemon.service  
-rw-r--r--. 1 root root 870 Nov 8 2023 avahi-daemon.socket  
-rw-r--r--. 1 root root 224 Oct 23 14:00 bar.service  
-rw-r--r--. 1 root root 964 Jul 18 13:00 basic.target  
drwxr-xr-x. 2 root root 6 Jul 18 13:00 basic.target.wants  
-r--r--r--. 1 root root 384 Feb 3 2024 blk-availability.service  
-rw-r--r--. 1 root root 449 Oct 31 2022 blockdev@.target
```

```
-rw-r--r--. 1 root root 707 Jun 14 2022 bluetooth.service
-rw-r--r--. 1 root root 435 Oct 31 2022 bluetooth.target
-rw-r--r--. 1 root root 642 Jan 16 2023 bolt.service
-rw-r--r--. 1 root root 463 Oct 31 2022 boot-complete.target
-rw-r--r--. 1 root root 217 Aug 9 2021 brltty.service
-rw-r--r--. 1 root root 491 Jul 13 2023 canberra-system-bootup.service
-rw-r--r--. 1 root root 509 Jul 13 2023 canberra-system-shutdown-reboot.service
-rw-r--r--. 1 root root 466 Jul 13 2023 canberra-system-shutdown.service
-rw-r--r--. 1 root root 1811 Jan 23 2024 chronyd-restricted.service
-rw-r--r--. 1 root root 1468 Jan 23 2024 chronyd.service
-rw-r--r--. 1 root root 1082 Jan 23 2024 chrony-wait.service
-rw-r--r--. 1 root root 277 Jul 23 09:37 cni-dhcp.service
-rw-r--r--. 1 root root 302 Dec 4 2023 cni-dhcp.socket
-rw-r--r--. 1 root root 222 Apr 2 2024 cockpit-motd.service
-rw-r--r--. 1 root root 720 Apr 2 2024 cockpit.service
-rw-r--r--. 1 root root 349 Apr 2 2024 cockpit.socket
-rw-r--r--. 1 root root 221 Apr 2 2024 cockpit-wsinstance-http.service
-rw-r--r--. 1 root root 165 Apr 2 2024 cockpit-wsinstance-https-factory@.service
-rw-r--r--. 1 root root 244 Apr 2 2024 cockpit-wsinstance-https-factory.socket
-rw-r--r--. 1 root root 215 Apr 2 2024 cockpit-wsinstance-http.socket
-rw-r--r--. 1 root root 264 Apr 2 2024 cockpit-wsinstance-https@.service
-rw-r--r--. 1 root root 478 Apr 2 2024 cockpit-wsinstance-https@.socket
-rw-r--r--. 1 root root 295 Aug 9 2021 colord.service
-rw-r--r--. 1 root root 154 Aug 11 2021 configure-printer@.service
-rw-r--r--. 1 root root 1102 Jul 18 13:00 console-getty.service
-rw-r--r--. 1 root root 1254 Jul 18 13:00 container-getty@.service
-rw-r--r--. 1 root root 294 Sep 13 18:49 cpupower.service
-rw-r--r--. 1 root root 371 Nov 30 2023 crond.service
-rw-r--r--. 1 root root 473 Oct 31 2022 cryptsetup-pre.target
-rw-r--r--. 1 root root 420 Oct 31 2022 cryptsetup.target
lrwxrwxrwx. 1 root root 13 Jul 18 13:00 ctrl-alt-del.target -> reboot.target
-rw-r--r--. 1 root root 234 Jan 7 2021 cups-browsed.service
-rw-r--r--. 1 root root 142 Jun 19 11:00 cups.path
-rw-r--r--. 1 root root 298 Jun 19 11:00 cups.service
```

```
drwxr-xr-x. 2 root root 25 Sep 25 12:03 cups.service.d
-rw-r--r--. 1 root root 148 Jun 19 11:00 cups.socket
-rw-r--r--. 1 root root 529 Aug 23 2022 dbus-broker.service
-rw-r--r--. 1 root root 560 Jun 12 2023 dbus-daemon.service
lrwxrwxrwx. 1 root root 25 Jul 18 13:00 dbus-org.freedesktop.hostname1.service -> systemd-hostnamed.service
lrwxrwxrwx. 1 root root 23 Jul 18 13:00 dbus-org.freedesktop.locale1.service -> systemd-localed.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@redhat9 tmp]# 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@redhat9 tmp]# 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@redhat9 tmp]# 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/sysconfig/ssh
ExecStart=/usr/sbin/sshd -D $OPTIONS
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 essaiera 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,**
 - 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@redhat9 tmp]# systemctl show sshd
Type=notify
ExitType=main
Restart=on-failure
NotifyAccess=main
RestartUsec=42s
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
TimeoutAbortUsec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUsec=infinity
RuntimeRandomizedExtraUsec=0
WatchdogUsec=0
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
```

```
MainPID=1175
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
Result=success
ReloadResult=success
CleanResult=success
UID=[not set]
GID=[not set]
NRestarts=0
OOMPolicy=stop
ReloadSignal=1
ExecMainStartTimestamp=Thu 2024-10-24 09:29:16 CEST
ExecMainStartTimestampMonotonic=38779836
ExecMainExitTimestampMonotonic=0
ExecMainPID=1175
ExecMainCode=0
ExecMainStatus=0
ExecStart={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $OPTIONS ; ignore_errors=no ; start_time=[Thu
2024-10-24 09:29:16 CEST] ; stop_time=[n/a] ; pid=1175 ; code=(null) ; status=0/0 }
ExecStartEx={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $OPTIONS ; flags= ; start_time=[Thu 2024-10-24
09:29:16 CEST] ; stop_time=[n/a] ; pid=1175 ; 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 }
ExecReloadEx={ path=/bin/kill ; argv[]=/bin/kill -HUP $MAINPID ; flags= ; start_time=[n/a] ; stop_time=[n/a] ;
pid=0 ; code=(null) ; status=0/0 }
Slice=system.slice
ControlGroup=/system.slice/sshd.service
ControlGroupId=4041
MemoryCurrent=5337088
MemoryAvailable=infinity
CPUUsageNSec=63348000
TasksCurrent=1
```

```
IPIngressBytes=[no data]
IPIngressPackets=[no data]
IPEgressBytes=[no data]
IPEgressPackets=[no data]
IOReadBytes=18446744073709551615
IOReadOperations=18446744073709551615
IOWriteBytes=18446744073709551615
IOWriteOperations=18446744073709551615
lines 1-55
```

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

```
[root@redhat9 tmp]# 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
○ └─iscsi-onboot.service
○ └─iscsi-starter.service
● └─kmod-static-nodes.service
○ └─ldconfig.service
● └─lvm2-lvmpolld.socket
● └─lvm2-monitor.service
○ └─multipathd.service
● └─nis-domainname.service
● └─plymouth-read-write.service
● └─plymouth-start.service
● └─proc-sys-fs-binfmt_misc.automount
```

- |—selinux-autorelabel-mark.service
- |—sys-fs-fuse-connections.mount
- |—sys-kernel-config.mount
- |—sys-kernel-debug.mount
- |—sys-kernel-tracing.mount
- |—systemd-ask-password-console.path
- |—systemd-binfmt.service
- |—systemd-boot-random-seed.service
- |—systemd-boot-update.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-network-generator.service
- |—systemd-pcrmachine.service
- |—systemd-pcrphase-sysinit.service
- |—systemd-pcrphase.service
- |—systemd-random-seed.service
- |—systemd-repart.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
 - |—systemd-cryptsetup@sdb12.service
- |—integritysetup.target
- |—local-fs.target

lines 1-55

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@redhat9 tmp]# 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@redhat9 tmp]# ls -l /etc/systemd/system
total 8
drwxr-xr-x. 2 root root 65 Oct 19 2023 basic.target.wants
drwxr-xr-x. 2 root root 31 Oct 19 2023 bluetooth.target.wants
lrwxrwxrwx. 1 root root 37 Oct 19 2023 ctrl-alt-del.target -> /usr/lib/systemd/system/reboot.target
lrwxrwxrwx. 1 root root 41 Oct 19 2023 dbus-org.bluez.service -> /usr/lib/systemd/system/bluetooth.service
lrwxrwxrwx. 1 root root 41 Oct 19 2023 dbus-org.fedoraproject.FirewallD1.service ->
/usr/lib/systemd/system/firewalld.service
lrwxrwxrwx. 1 root root 44 Oct 19 2023 dbus-org.freedesktop.Avahi.service -> /usr/lib/systemd/system/avahi-
daemon.service
lrwxrwxrwx. 1 root root 44 Oct 19 2023 dbus-org.freedesktop.ModemManager1.service ->
/usr/lib/systemd/system/ModemManager.service
lrwxrwxrwx. 1 root root 57 Oct 19 2023 dbus-org.freedesktop.nm-dispatcher.service ->
/usr/lib/systemd/system/NetworkManager-dispatcher.service
lrwxrwxrwx. 1 root root 43 Oct 19 2023 dbus.service -> /usr/lib/systemd/system/dbus-broker.service
lrwxrwxrwx. 1 root root 40 Oct 19 2023 default.target -> /usr/lib/systemd/system/graphical.target
drwxr-xr-x. 2 root root 45 Oct 19 2023 default.target.wants
drwxr-xr-x. 2 root root 38 Oct 19 2023 'dev-virtio\x2dports-org.qemu.guest_agent.0.device.wants'
lrwxrwxrwx. 1 root root 35 Oct 19 2023 display-manager.service -> /usr/lib/systemd/system/gdm.service
drwxr-xr-x. 2 root root 32 Oct 19 2023 getty.target.wants
drwxr-xr-x. 2 root root 181 Oct 19 2023 graphical.target.wants
drwxr-xr-x. 2 root root 36 Oct 19 2023 local-fs.target.wants
drwxr-xr-x. 2 root root 4096 Oct 24 08:02 multi-user.target.wants
```

```
drwxr-xr-x. 2 root root 48 Oct 19 2023 network-online.target.wants
drwxr-xr-x. 2 root root 26 Oct 19 2023 printer.target.wants
drwxr-xr-x. 2 root root 27 Oct 19 2023 remote-fs.target.wants
drwxr-xr-x. 2 root root 186 Oct 19 2023 sockets.target.wants
drwxr-xr-x. 2 root root 4096 Sep 25 12:08 sysinit.target.wants
drwxr-xr-x. 2 root root 27 Oct 23 13:09 sysstat-collect.timer.d
drwxr-xr-x. 2 root root 64 Oct 23 12:58 sysstat.service.wants
drwxr-xr-x. 2 root root 27 Oct 21 16:32 systemd-tmpfiles-clean.timer.d
drwxr-xr-x. 2 root root 86 Oct 19 2023 timers.target.wants
drwxr-xr-x. 2 root root 29 Oct 19 2023 vmtoolsd.service.requires
```

LAB #3 - La Commande systemd-analyze

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

```
[root@redhat9 tmp]# systemd-analyze
Startup finished in 1.361s (kernel) + 4.348s (initrd) + 1min 21.082s (userspace) = 1min 26.792s
graphical.target reached after 53.162s 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@redhat9 tmp]# systemd-analyze blame
44.601s kdump.service
18.590s systemd-cryptsetup@sdb12.service
16.592s plymouth-quit-wait.service
9.801s dev-sdb8.device
9.801s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart8.device
9.801s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb8.device
9.801s dev-disk-by\x2dpartuuid-23a56c2e\x2d08.device
9.801s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart8.device
9.801s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1\x2dpart8.device
9.800s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb10.device
```

```
9.800s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart10.device
9.800s dev-sdb10.device
9.800s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1\x2dpart10.device
9.800s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart10.device
9.800s dev-disk-by\x2dpartuuid-23a56c2e\x2d0a.device
9.800s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb5.device
9.800s dev-sdb5.device
9.800s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1\x2dpart5.device
9.800s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart5.device
9.800s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart5.device
9.799s dev-disk-by\x2dpartuuid-23a56c2e\x2d05.device
9.782s sys-module-fuse.device
9.750s dev-ttyS3.device
9.750s sys-devices-platform-serial8250-tty-ttyS3.device
9.749s dev-ttyS0.device
9.749s sys-devices-platform-serial8250-tty-ttyS0.device
9.749s dev-ttyS1.device
9.749s sys-devices-platform-serial8250-tty-ttyS1.device
9.749s dev-ttyS2.device
9.749s sys-devices-platform-serial8250-tty-ttyS2.device
9.717s sys-devices-pci0000:00-0000:00:12.0-virtio1-net-ens18.device
9.717s sys-subsystem-net-devices-ens18.device
9.717s sys-module-configfs.device
9.680s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart6.device
9.680s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb6.device
9.680s dev-disk-by\x2dpartuuid-23a56c2e\x2d06.device
9.680s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart6.device
9.680s dev-sdb6.device
9.680s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1\x2dpart6.device
9.654s dev-disk-by\x2dpath-pci\x2d0000:01:01.0\x2dscsi\x2d0:0:0:0\x2dpart2.device
9.654s sys-devices-pci0000:00-0000:00:05.0-0000:01:01.0-virtio2-host0-target0:0:0-0:0:0:0-block-sda-sda2.device
9.654s dev-sda2.device
9.654s dev-disk-by\x2dpartuuid-d00dfc8a\x2d02.device
9.654s dev-disk-by\x2did-scsi\x2d0QEMU_QEMU_HARDDISK_drive\x2dscsi0\x2dpart2.device
```

```
9.647s dev-disk-by\x2dpartuuid-23a56c2e\x2d09.device
9.647s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb9.device
9.647s dev-sdb9.device
9.647s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart9.device
9.647s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1\x2dpart9.device
9.647s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart9.device
9.635s dev-sdb7.device
9.635s sys-devices-pci0000:00-0000:00:07.0-ata3-host3-target3:0:0-3:0:0:0-block-sdb-sdb7.device
9.635s dev-disk-by\x2dpath-pci\x2d0000:00:07.0\x2data\x2d1.0\x2dpart7.device
9.635s dev-disk-by\x2dpartuuid-23a56c2e\x2d07.device
9.635s dev-disk-by\x2did-ata\x2dQEMU_HARDDISK_QM00005\x2dpart7.device
lines 1-55
```

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@redhat9 tmp]# systemd-analyze critical-chain sshd.service
The time when unit became active or started is printed after the "@" character.
The time the unit took to start is printed after the "+" character.

sshd.service +348ms
└─network.target @33.014s
   └─wpa_supplicant.service @49.199s +119ms
      └─basic.target @27.729s
         └─dbus-broker.service @26.718s +1.007s
            └─dbus.socket @26.682s
               └─sysinit.target @26.678s
                  └─systemd-update-utmp.service @26.658s +19ms
                     └─auditd.service @26.278s +369ms
                        └─systemd-tmpfiles-setup.service @25.954s +313ms
                           └─local-fs.target @25.942s
                              └─run-user-1000.mount @1min 19.309s
                                 └─local-fs-pre.target @6.742s
                                    └─systemd-tmpfiles-setup-dev.service @3.994s +1.040s
```

```
└─kmod-static-nodes.service @3.795s +53ms
  └─systemd-journald.socket
    └─system.slice
      └─.slice
```

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

```
[root@redhat9 tmp]# systemd-analyze --help
systemd-analyze [OPTIONS...] COMMAND ...
```

Profile systemd, show unit dependencies, check unit files.

Commands:

[time]	Print time required to boot the machine
blame	Print list of running units ordered by time to init
critical-chain [UNIT...]	Print a tree of the time critical chain of units
plot	Output SVG graphic showing service initialization
dot [UNIT...]	Output dependency graph in dot(1) format
dump [PATTERN...]	Output state serialization of service manager
cat-config	Show configuration file and drop-ins
unit-files	List files and symlinks for units
unit-paths	List load directories for units
exit-status [STATUS...]	List exit status definitions
capability [CAP...]	List capability definitions
syscall-filter [NAME...]	List syscalls in seccomp filters
filesystems [NAME...]	List known filesystems
condition CONDITION...	Evaluate conditions and asserts
compare-versions VERSION1 [OP] VERSION2	Compare two version strings
verify FILE...	Check unit files for correctness

calendar SPEC...	Validate repetitive calendar time events
timestamp TIMESTAMP...	Validate a timestamp
timespan SPAN...	Validate a time span
security [UNIT...]	Analyze security of unit
inspect-elf FILE...	Parse and print ELF package metadata

Options:

--recursive-errors=MODE	Control which units are verified
--offline=BOOL	Perform a security review on unit file(s)
--threshold=N	Exit with a non-zero status when overall exposure level is over threshold value
--security-policy=PATH	Use custom JSON security policy instead of built-in one
--json=pretty short off	Generate JSON output of the security analysis table, or plot's raw time data
--no-pager	Do not pipe output into a pager
--no-legend	Disable column headers and hints in plot with either --table or --json=
--system	Operate on system systemd instance
--user	Operate on user systemd instance
--global	Operate on global user configuration
-H --host=[USER@]HOST	Operate on remote host
-M --machine=CONTAINER	Operate on local container
--order	Show only order in the graph
--require	Show only requirement in the graph
--from-pattern=GLOB	Show only origins in the graph
--to-pattern=GLOB	Show only destinations in the graph
--fuzz=SECONDS	Also print services which finished SECONDS

lines 1-55

LAB #4 - Les Cibles Systemd

Chaque Cible est décrite par un fichier de configuration :

```
[root@redhat9 tmp]# cat /usr/lib/systemd/system/graphical.target
# SPDX-License-Identifier: LGPL-2.1-or-later
#
# 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@redhat9 tmp]# systemctl list-dependencies multi-user.target
multi-user.target
● |atd.service
● |auditd.service
● |avahi-daemon.service
● |bar.service
● |crond.service
● |cups.path
● |cups.service
● |firewalld.service
● |foo.service
● |httpd.service
○ |insights-client-boot.service
● |irqbalance.service
● |kdump.service
● |libstoragemgmt.service
● |mcelog.service
● |mdmonitor.service
● |ModemManager.service
● |NetworkManager.service
○ |ostree-readonly-sysroot-migration.service
● |plymouth-quit-wait.service
○ |plymouth-quit.service
● |psacct.service
● |rhsmcertd.service
● |rsyslog.service
○ |smartd.service
```

```
● |—sshd.service
○ |—sssd.service
● |—sysstat.service
● |—systemd-ask-password-wall.path
● |—systemd-logind.service
○ |—systemd-update-utmp-runlevel.service
● |—systemd-user-sessions.service
○ |—vmtoolsd.service
● |—basic.target
● |   |— .mount
○ |   |—low-memory-monitor.service
○ |   |—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
○ |   |—multipathd.socket
● |   |—sssd-kcm.socket
● |   |—systemd-coredump.socket
● |   |—systemd-initctl.socket
● |   |—systemd-journald-dev-log.socket
● |   |—systemd-journald.socket
lines 1-55
```

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 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@redhat9 tmp]# systemctl --failed
UNIT LOAD ACTIVE SUB DESCRIPTION
0 loaded units listed.
```

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@redhat9 tmp]# ls -l /etc/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx. 1 root root 35 Oct 19 2023 atd.service -> /usr/lib/systemd/system/atd.service
lrwxrwxrwx. 1 root root 38 Oct 19 2023 auditd.service -> /usr/lib/systemd/system/auditd.service
lrwxrwxrwx. 1 root root 44 Oct 19 2023 avahi-daemon.service -> /usr/lib/systemd/system/avahi-daemon.service
lrwxrwxrwx. 1 root root 35 Oct 23 14:01 bar.service -> /usr/lib/systemd/system/bar.service
lrwxrwxrwx. 1 root root 37 Oct 19 2023 crond.service -> /usr/lib/systemd/system/crond.service
lrwxrwxrwx. 1 root root 33 Oct 19 2023 cups.path -> /usr/lib/systemd/system/cups.path
lrwxrwxrwx. 1 root root 36 Oct 19 2023 cups.service -> /usr/lib/systemd/system/cups.service
lrwxrwxrwx. 1 root root 41 Oct 19 2023 firewalld.service -> /usr/lib/systemd/system/firewalld.service
lrwxrwxrwx. 1 root root 35 Oct 23 13:59 foo.service -> /usr/lib/systemd/system/foo.service
lrwxrwxrwx. 1 root root 37 Oct 22 10:15 httpd.service -> /usr/lib/systemd/system/httpd.service
lrwxrwxrwx. 1 root root 52 Oct 19 2023 insights-client-boot.service -> /usr/lib/systemd/system/insights-client-boot.service
lrwxrwxrwx. 1 root root 42 Oct 19 2023 irqbalance.service -> /usr/lib/systemd/system/irqbalance.service
lrwxrwxrwx. 1 root root 37 Oct 19 2023 kdump.service -> /usr/lib/systemd/system/kdump.service
lrwxrwxrwx. 1 root root 46 Oct 19 2023 libstoragemgmt.service -> /usr/lib/systemd/system/libstoragemgmt.service
lrwxrwxrwx. 1 root root 38 Oct 19 2023 mcelog.service -> /usr/lib/systemd/system/mcelog.service
lrwxrwxrwx. 1 root root 41 Oct 19 2023 mdmonitor.service -> /usr/lib/systemd/system/mdmonitor.service
lrwxrwxrwx. 1 root root 44 Oct 19 2023 ModemManager.service -> /usr/lib/systemd/system/ModemManager.service
lrwxrwxrwx. 1 root root 46 Oct 19 2023 NetworkManager.service -> /usr/lib/systemd/system/NetworkManager.service
lrwxrwxrwx. 1 root root 38 Oct 22 15:41 psacct.service -> /usr/lib/systemd/system/psacct.service
```

```
lrwxrwxrwx. 1 root root 40 Oct 19 2023 remote-fs.target -> /usr/lib/systemd/system/remote-fs.target
lrwxrwxrwx. 1 root root 41 Oct 19 2023 rhsmcertd.service -> /usr/lib/systemd/system/rhsmcertd.service
lrwxrwxrwx. 1 root root 39 Oct 19 2023 rsyslog.service -> /usr/lib/systemd/system/rsyslog.service
lrwxrwxrwx. 1 root root 38 Oct 19 2023 smartd.service -> /usr/lib/systemd/system/smartd.service
lrwxrwxrwx. 1 root root 36 Oct 19 2023 sshd.service -> /usr/lib/systemd/system/sshd.service
lrwxrwxrwx. 1 root root 36 Oct 19 2023 sssd.service -> /usr/lib/systemd/system/sss.service
lrwxrwxrwx. 1 root root 39 Oct 23 12:58 sysstat.service -> /usr/lib/systemd/system/sysstat.service
lrwxrwxrwx. 1 root root 40 Oct 19 2023 vmtocls.service -> /usr/lib/systemd/system/vmtocls.service

[root@redhat9 tmp]# ls -l /usr/lib/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx. 1 root root 15 Jul 18 13:00 getty.target -> ../getty.target
lrwxrwxrwx. 1 root root 44 May 17 18:27 ostree-readonly-sysroot-migration.service -> ../ostree-readonly-sysroot-migration.service
lrwxrwxrwx. 1 root root 24 Jul 14 2023 plymouth-quit.service -> ../plymouth-quit.service
lrwxrwxrwx. 1 root root 29 Jul 14 2023 plymouth-quit-wait.service -> ../plymouth-quit-wait.service
lrwxrwxrwx. 1 root root 33 Jul 18 13:00 systemd-ask-password-wall.path -> ../systemd-ask-password-wall.path
lrwxrwxrwx. 1 root root 25 Jul 18 13:00 systemd-logind.service -> ../systemd-logind.service
lrwxrwxrwx. 1 root root 39 Jul 18 13:00 systemd-update-utmp-runlevel.service -> ../systemd-update-utmp-runlevel.service
lrwxrwxrwx. 1 root root 32 Jul 18 13:00 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@redhat9 tmp]# systemctl get-default
graphical.target
```

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

```
[root@redhat9 tmp]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40 Oct 19 2023 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/graphical.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@redhat9 tmp]# systemctl get-default
graphical.target

[root@redhat9 tmp]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40 Oct 19 2023 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/graphical.target

[root@redhat9 tmp]# 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@redhat9 tmp]# systemctl get-default
multi-user.target

[root@redhat9 tmp]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 41 Oct 24 11:47 /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-
user.target

[root@redhat9 tmp]# 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@redhat9 tmp]# systemctl get-default
graphical.target
```

```
[root@redhat9 tmp]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40 Oct 24 11:47 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/graphical.target
```

Modifier la Cible en Cours

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

```
[root@redhat9 tmp]# systemctl isolate rescue

[root@redhat9 tmp]# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
    rescue.target                loaded active active      Rescue Mode

[root@redhat9 tmp]# runlevel
5 1

[root@redhat9 tmp]# who -r
      run-level 1   2024-10-24 11:48                last=5
```

```
[root@redhat9 tmp]# systemctl isolate graphical

[root@redhat9 tmp]# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
    graphical.target            loaded active active    Graphical Interface

[root@redhat9 tmp]# runlevel
1 5

[root@redhat9 tmp]# who -r
      run-level 5   2024-10-24 11:50                last=1
```

Arrêt du Système

Sous Red Hat 9 les commandes **halt**, **poweroff**, **reboot** et **shutdown** sont des liens symboliques vers **/bin/systemctl** :

```
[root@redhat9 tmp]# ls -l /usr/sbin/shutdown /usr/sbin/halt /usr/sbin/poweroff /usr/sbin/reboot
lrwxrwxrwx. 1 root root 16 Jul 18 13:00 /usr/sbin/halt -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Jul 18 13:00 /usr/sbin/poweroff -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Jul 18 13:00 /usr/sbin/reboot -> ../bin/systemctl
lrwxrwxrwx. 1 root root 16 Jul 18 13:00 /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@redhat9 tmp]# shutdown --help
shutdown [OPTIONS...] [TIME] [WALL...]
```

Shut down the system.

Options:

```
--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
--show       Show pending shutdown
```

See the shutdown(8) man page for details.

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@redhat9 tmp]# date && shutdown -t 60 -k
Thu Oct 24 11:53:34 AM CEST 2024
Shutdown scheduled for Thu 2024-10-24 11:54:34 CEST, use 'shutdown -c' to cancel.
```

La Commande reboot

Cette commande redémarre le système.

Options de la commande

Les options de cette commande sont :

```
[root@redhat9 tmp]# reboot --help
reboot [OPTIONS...] [ARG]
```

Reboot the system.

Options:

--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

See the halt(8) man page for details.

La Commande halt

Cette commande arrête le système.

Options de la commande

Les options de cette commande sont :

```
[root@redhat9 tmp]# halt --help
halt [OPTIONS...]
```

Halt the system.

Options:

--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

See the halt(8) man page for details.

La Commande poweroff

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

Options de la commande

Les options de cette commande sont :

```
[root@redhat9 tmp]# poweroff --help
poweroff [OPTIONS...]
```

Power off the system.

Options:

--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

See the halt(8) man page for details.