

Version : **2022.01**

Dernière mise-à-jour : 2022/11/01 07:24

Topic 202: System Startup

- **Topic 202: System Startup**

- Présentation
- BIOS, EFI et OpenFirmware
 - Systèmes à base du BIOS
 - Charger de Démarrage
 - Systèmes à base de l'EFI
 - Autres Systèmes
- Gestionnaire d'amorçage
 - LILO
 - La Commande LILO
 - Codes Erreur de LILO
 - Grub Legacy sous RHEL/CentOS 6
 - Configurer GRUB Legacy
 - La Section Globale
 - Une Section spécifique à un OS
 - Configurer l'Authentification
 - Modifier la Configuration de GRUB Legacy en Ligne de Commande
 - GRUB 2 sous Debian 11
 - Le fichier /boot/grub2/device.map
 - Le fichier /etc/default/grub
 - Les fichiers du répertoire /etc/grub.d
 - 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 sous RHEL/CentOS 6
 - Examiner l'image existante
 - Le script init
 - La Commande dracut
- Initramfs sous Debian 11
 - LAB #1 - Travailler avec Initramfs
 - 1.1 - Examiner l'image existante
 - 1.2 - La Commande mkinitramfs
- Processus de Démarrage du Noyau Linux
- Processus Init
- Démarrer RHEL/CentOS 5 avec SysVinit
 - Niveaux d'exécution sous RHEL/CentOS 5
 - Inittab
 - Scripts de Démarrage
 - rc.sysinit sous RHEL/CentOS 5
 - Répertoire init.d
 - Répertoires rcx.d
 - rc.local
 - La Commande chkconfig
- Démarrer Debian 6 avec SysVinit
 - Niveaux d'exécution sous Debian 6
 - Inittab
 - Scripts de Démarrage
 - Le script rc.S sous Debian 6
 - Répertoire init.d
 - Répertoires rcx.d
 - rc.local
 - La Commande chkconfig
 - La Commande update-rc.d sous Debian 6
 - Linux Standard Base

- Codes Retour Standardisés
- Scripts
- Démarrer RHEL/CentOS 6 avec Upstart
 - Inittab
 - Initialisation du Système
 - Runlevels
 - [CTL]-[ALT]-[DEL]
 - mingetty
 - Gestion des Services
 - La Commande initctl
 - Jobs
 - Événements
 - Etats
 - Démarrer et Arrêter les Jobs
 - La Commande status
 - La Commande start
 - La Commande stop
 - La Commande restart
- Démarrer Debian 11 avec systemd
 - LAB #2 - La Commande systemctl
 - LAB #3 - Fichiers de Configuration
 - 3.1 - Fichiers de Configuration par Défaut
 - 3.2 - Surchargement des Fichiers par Défaut
 - 3.3 - Les Fichiers d'Unités
 - LAB #4 - La Commande systemd-analyze
 - LAB #5 - Les Cibles
 - 5.1 - Contrôler les dépendances d'une Cible
 - 5.2 - La Cible par Défaut
 - LAB #6 - Gestion des Services
 - 6.1 - Gestion des Instances Uniques
 - 6.2 - Gestion d'Instances Multiples
 - 6.3 - Interdire la Modification du Statut d'un Service

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**.

Charger de Démarrage

La première partie du Charger 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 parfaitement 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

Des gestionnaires d'amorçage sous Linux, un se distingue comme étant le plus utilisé :

- GRUB (Grand Unified Boot Loader)

Cependant il en existe d'autres :

- LILO (Linux LOader)
- SysLinux
- LoadLin
- ...

LILO

LILO (*Linux LOader*) est configuré par le fichier **/etc/lilo.conf**.

La Commande LILO

La commande **lilo** peut prendre une de plusieurs options. Les options les plus importantes sont :

Option	Description
-M	Permet d'écrire sur le MBR
-d	Permet de réduire ou augmenter le temps d'attente avant le lancement du noyau par défaut
-D	Permet de sélectionner un noyau par défaut en indiquant son label

Option	Description
-u	Permet de désinstaller LILO
-v	Permet d'activer le mode verbose
-m	Permet de modifier le fichier map par défaut (/boot/map)
-i	Permet de spécifier un nouveau fichier à utiliser comme secteur de boot (/boot/boot.b)
-C	Permet de modifier le fichier de configuration par défaut
-q	Permet de créer le fichier /boot/map qui contient l'emplacement des noyaux qui peuvent être booter

Codes Erreur de LILO

Lors du démarrage, LILO permet d'identifier les éventuelles erreurs :

Affichage	Erreur
(rien)	Aucun morceau de LILO n'a été chargé. Soit LILO n'est pas installé, soit la partition sur laquelle son secteur d'amorce se trouve n'est pas active.
L	Le premier morceau du chargeur d'amorce a été chargé et démarré, mais il ne peut charger le second morceau. Les codes d'erreur à deux chiffres indiquent le type de problème. (Voir également la section "Codes d'erreur disque".) Ce cas indique en général une panne de périphérique ou une incohérence de géométrie (c'est à dire de mauvais paramètres disques).
LI	Le premier morceau du chargeur d'amorce a pu charger le second morceau, mais n'a pas réussi à l'exécuter. Cela peut être causé par une incohérence de géométrie ou par le déplacement de /boot/boot.b sans lancer l'installateur de carte.
LIL	Le second morceau du chargeur d'amorce a été démarré, mais il ne trouve pas la table de descripteurs dans le fichier carte. C'est en général dû à une panne de périphérique ou une incohérence de géométrie.
LIL?	Le second morceau du chargeur d'amorce a été chargé à un adresse incorrecte. C'est en général causé par une subtile incohérence de géométrie, ou par le déplacement de /boot/boot.b sans lancer l'installateur de carte.
LIL-	La table de descripteurs est corrompue. Cela peut être dû à une incohérence de géométrie ou au déplacement de /boot/map sans lancer l'installateur.
LILO	Tous les éléments de LILO ont été correctement chargés.

Si le BIOS signale une erreur lorsque LILO essaye de charger une image d'amorce, le code d'erreur correspondant est affiché. Ces codes vont de 0x00 à 0xbb. Reportez-vous au Guide Utilisateur de LILO pour leur explication.





Important : LILO ne gère pas les systèmes UEFI.

Grub Legacy sous RHEL/CentOS 6



A retenir : Grub Legacy a été créé en **1999**.

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

```
# grub-install /dev/sda [Entrée]
```

où **sda** est le nom du périphérique où grub doit s'installer dans le MBR. Notez cependant que le MBR a une taille trop petite pour contenir tout le Charger de Démarrage. Pour cette raison, le gestionnaire est divisé en deux. Le gestionnaire de niveau 1 est stocké dans le MBR et indique où se trouve le gestionnaire de niveau 2, c'est-à-dire le répertoire /boot.

Il est aussi possible d'utiliser la commande :

```
# grub-install '(hd0)' [Entrée]
```

où **hd0** indique à grub le premier disque.

Il est à noter que la première partie du Charger de Démarrage peut également être installé dans un **PBR** (*Partition Boot Record*) dans le cas d'un système dual-boot voire multi-boot au quel cas on parle de **chainloading** :

```
# grub-install /dev/sda1 [Entrée]
```

ou :

```
# grub-install 'hd(0,0)' [Entrée]
```

où **(0,0)** est la nomenclature grub pour la première partition du premier disque.

Le gestionnaire de niveau 1 a pour seul but le lancement du gestionnaire de niveau 2. Le gestionnaire de niveau 2 charge le noyau en mémoire, monte l'image **initrd** et charge les modules nécessaires pendant que le noyau monte la partition racine / en lecture seule.

Pour désinstaller grub du MBR, utilisez une disquette DOS pour démarrer la machine puis taper la commande suivante au prompt :

```
A> fdisk /mbr [Entrée]
```

Configurer GRUB Legacy

grub se configure grâce au fichier **/boot/grub/menu.lst**. Pour visualiser ce fichier, il convient de saisir la commande suivante :

```
[root@centos6 ~]# cat /boot/grub/menu.lst
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/sda2
#           initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS Linux (2.6.32-71.29.1.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.29.1.el6.i686 ro root=UUID=e73735d6-c14b-4a40-8735-f34fc868da8a rd_NO_LUKS
rd_NO_LVM rd_NO_MD rd_NO_DM LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=fr-latin9
crashkernel=auto rhgb quiet
    initrd /initramfs-2.6.32-71.29.1.el6.i686.img
```

```
title centos (2.6.32-71.el6.i686)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.el6.i686 ro root=UUID=e73735d6-c14b-4a40-8735-f34fc868da8a rd_NO_LUKS rd_NO_LVM
rd_NO_MD rd_NO_DM LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=fr-latin9 crashkernel=auto
rhgb quiet
    initrd /initramfs-2.6.32-71.el6.i686.img
```



Important : Sous les distributions dérivées de Red Hat, par exemple Fedora, le fichier menu.lst s'appelle **grub.conf**.

Ce fichier comporte plusieurs sections :

- * la section des ****paramètres globaux****,
- * une ou plusieurs sections pour chaque système d'opération installé sur la machine.

La Section Globale

Paramètre	Explication
default=0	Ce paramètre désigne le numéro de l'entrée à charger par défaut. La valeur de 0 indique la première section commençant par le mot clef title
timeout=5	Ce paramètre indique le délai en secondes après lequel l'entrée par défaut sera chargée.
splashimage=(hd0,0)/grub/splash.xpm.gz	Ce paramètre indique l'emplacement de l'image de fond du menu de GRUB Legacy
hiddenmenu	Ce paramètre cache le menu de GRUB Legacy pendant le chargement de l'entrée par défaut, sauf si l'utilisateur appuie sur une touche
color	Ce paramètre prend la forme de deux paires de couleurs. Le premier, par exemple white/blue, définit les couleurs de l'avant-plan et de l'arrière-plan des entrées normales du menu, tandis que le deuxième, par exemple yellow/blue définit les couleurs de l'avant-plan et de l'arrière-plan des entrées sélectionnées du menu
password=motdepasse	Cette option n'est présente que dans le cas où un mot de passe a été spécifié pour protéger GRUB Legacy

Une Section spécifique à un OS

Paramètre	Explication
title CentOS Linux (2.6.32-71.29.1.el6.i686)	Ce paramètre indique le début d'une section de configuration d'une entrée pour un système d'exploitation ainsi que le nom qui apparaît dans le menu de GRUB Legacy
root (hd0,0)	Ce paramètre indique la partition contenant le noyau de Linux. Dans l'exemple hd0,0 indique la première partition du premier disque dur. Cette partition est ensuite montée en tant que /boot .
kernel /vmlinuz-2.6.32-71.29.1.el6.i686	Ce paramètre indique le nom du noyau à charger pour démarrer la machine. Son chemin est relatif à la partition de démarrage (hd0,0). Certaines options peuvent être passées au noyau en les spécifiant en tant qu'arguments telles rhgb ou Red Hat Graphical Boot et quiet qui supprime les messages de démarrage qui apparaissent avant le lancement de l'animation graphique activée par l'option rhgb
initrd	Ce paramètre stipule l'emplacement du disque initial chargé en mémoire lors du démarrage. Son chemin est relatif à la partition de démarrage (hd0,0).
lock	
rootnoverify (hd0,1)	Ce paramètre indique une section concernant un système d'exploitation non Linux, tel que Windows™

Configurer l'Authentification

Grub Legacy peut être protégé par un mot de passe en incluant la directive suivante dans le fichier `/boot/grub/menu.lst` :

```
password --md5 <mot de passe>
```

Le mot de passe doit être chiffré avec MD5 :

```
[root@centos6 ~]# grub-md5-crypt
Password:
Retype password:
$1$VL0zG$ibdk0my4IHny/XtNIGRhv1
```

Editez ensuite le fichier **`/boot/grub/menu.lst`** :

```
[root@centos6 ~]# cat /boot/grub/menu.lst
# grub.conf generated by anaconda
```

```
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/sda2
#           initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
password --md5 $1$VL0zG$ibdk0my4IHny/XtNIGRhv1
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-504.1.3.el6.i686)
    lock
    root (hd0,0)
    kernel /vmlinuz-2.6.32-504.1.3.el6.i686 ro root=UUID=b9f29672-c84e-4d3b-b132-189758a084eb rd_NO_LUKS rd_NO_MD
LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=fr-latin9 rd_NO_LVM
rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-504.1.3.el6.i686.img
title CentOS (2.6.32-358.18.1.el6.i686)
    lock
    root (hd0,0)
    kernel /vmlinuz-2.6.32-358.18.1.el6.i686 ro root=UUID=b9f29672-c84e-4d3b-b132-189758a084eb rd_NO_LUKS
rd_NO_MD LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=fr-latin9
rd_NO_LVM rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-358.18.1.el6.i686.img
title CentOS (2.6.32-358.6.1.el6.i686)
    lock
    root (hd0,0)
    kernel /vmlinuz-2.6.32-358.6.1.el6.i686 ro root=UUID=b9f29672-c84e-4d3b-b132-189758a084eb rd_NO_LUKS rd_NO_MD
LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=fr-latin9 rd_NO_LVM
rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-358.6.1.el6.i686.img
```

```
title CentOS (2.6.32-279.el6.i686)
    lock
    root (hd0,0)
    kernel /vmlinuz-2.6.32-279.el6.i686 ro root=UUID=b9f29672-c84e-4d3b-b132-189758a084eb rd_NO_LUKS rd_NO_MD
LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=fr-latin9 rd_NO_LVM
rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-279.el6.i686.img
```



Important : Notez l'addition de la ligne **password -md5**
\$1\$VLOzG\$ibdk0my4IHny/XtNIGRhv1 ainsi que le mot clef **lock** sur une ligne située
après chaque ligne commençant par **title**.

Modifier la Configuration de GRUB Legacy en Ligne de Commande

Lors du démarrage de GRUB Legacy, il est possible de voir son menu en appuyant sur n'importe quelle touche. Si GRUB Legacy a été protégé par un mot de passe, il convient d'appuyer sur la touche **p** puis de rentrer le mot de passe. A ce stade il est possible d'utiliser deux autres touches :

- la touche **e** pour accéder à l'éditeur de l'interface,
- la touche **c** pour accéder à la ligne de commande.

En mode édition notez l'utilisation des touches suivantes :

- **e** : éditer une ligne,
- **d** : supprimer une ligne,
- **o** : ajouter une ligne après la ligne courante,
- **O** : ajouter une ligne avant la ligne courante,
- **b** : démarrer avec la configuration modifiée,
- **echap** : abandonner les modifications et retourner à l'interface de GRUB.

GRUB 2 sous Debian 11

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/grub/**, 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 :

```
# grub-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/grub/grub.cfg**. Pour visualiser ce fichier, il convient de saisir la commande suivante :

```
root@debian11:~# cat /boot/grub/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
  set have_grubenv=true
  load_env
fi
if [ "${next_entry}" ] ; then
```

```
    set default="${next_entry}"
    set next_entry=
    save_env next_entry
    set boot_once=true
else
    set default="0"
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_uga
    insmod ieee1275_fb
    insmod vbe
    insmod vga
    insmod video_bochs
    insmod video_cirrus
  fi
}

if [ x$feature_default_font_path = xy ] ; then
  font=unicode
else
  insmod part_msdos
  insmod ext2
  set root='hd0,msdos1'
  if [ x$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 9887a74f-a680-4bde-8f04-db5ae9ea186e
  else
    search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
  fi
  font="/usr/share/grub/unicode.pf2"
fi

if loadfont $font ; then
  set gfxmode=auto
  load_video
  insmod gfxterm
  set locale_dir=$prefix/locale
  set lang=en_US
  insmod gettext
fi
terminal_output gfxterm
```

```
if [ "${recordfail}" = 1 ] ; then
    set timeout=30
else
    if [ x$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
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/05_debian_theme ###
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 9887a74f-a680-4bde-8f04-db5ae9ea186e
else
    search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
fi
insmod png
if background_image /usr/share/desktop-base/homeworld-theme/grub/grub-4x3.png; then
    set color_normal=white/black
    set color_highlight=black/white
else
    set menu_color_normal=cyan/blue
    set menu_color_highlight=white/blue
fi
### END /etc/grub.d/05_debian_theme ###
```

```
### BEGIN /etc/grub.d/10_linux ###
function gfxmode {
    set gfxpayload="${1}"
}
set linux_gfx_mode=
export linux_gfx_mode
menuentry 'Debian GNU/Linux' --class debian --class gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-simple-9887a74f-a680-4bde-8f04-db5ae9ea186e' {
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    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 9887a74f-a680-4bde-8f04-db5ae9ea186e
    else
        search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
    fi
    echo    'Loading Linux 5.10.0-13-amd64 ...'
    linux   /boot/vmlinuz-5.10.0-13-amd64 root=UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e ro quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/initrd.img-5.10.0-13-amd64
}
submenu 'Advanced options for Debian GNU/Linux' $menuentry_id_option 'gnulinux-advanced-9887a74f-a680-4bde-8f04-
db5ae9ea186e' {
    menuentry 'Debian GNU/Linux, with Linux 5.10.0-13-amd64' --class debian --class gnu-linux --class gnu --
class os $menuentry_id_option 'gnulinux-5.10.0-13-amd64-advanced-9887a74f-a680-4bde-8f04-db5ae9ea186e' {
        load_video
        insmod gzio
        if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
        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-
baremetal=ahci0,msdos1 9887a74f-a680-4bde-8f04-db5ae9ea186e
        else
            search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
        fi
        echo 'Loading Linux 5.10.0-13-amd64 ...'
        linux /boot/vmlinuz-5.10.0-13-amd64 root=UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e ro quiet
        echo 'Loading initial ramdisk ...'
        initrd /boot/initrd.img-5.10.0-13-amd64
    }
    menuentry 'Debian GNU/Linux, with Linux 5.10.0-13-amd64 (recovery mode)' --class debian --class gnu-linux
--class gnu --class os $menuentry_id_option 'gnulinux-5.10.0-13-amd64-recovery-9887a74f-a680-4bde-8f04-
db5ae9ea186e' {
        load_video
        insmod gzio
        if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
        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 9887a74f-a680-4bde-8f04-db5ae9ea186e
        else
            search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
        fi
        echo 'Loading Linux 5.10.0-13-amd64 ...'
        linux /boot/vmlinuz-5.10.0-13-amd64 root=UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e ro single
        echo 'Loading initial ramdisk ...'
        initrd /boot/initrd.img-5.10.0-13-amd64
    }
}
```

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

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

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

### 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 ###
```

Prenons le cas des paramètres de Grub Legacy et comparons-les aux paramètres de GRUB 2 :

Grub Legacy	GRUB 2
title	Menuentry
root (hd0,0)	set root=hd(0,1). Notez que GRUB 2 commence toujours la numérotation des disques à 0 mais numérote les partitions à partir de 1
kernel	linux
initrd	initrd

Grub Legacy	GRUB 2
lock	Ce paramètre n'existe plus sous GRUB 2.
rootnoverify (hd0,1)	Ce paramètre n'existe plus sous GRUB 2. Les paramètres des systèmes d'exploitation non Linux sont définis avec le paramètre root

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

- grub-mkconfig -o /boot/grub/grub.cfg
- grub-mkconfig -o /boot/efi/EFI/debian/grub.cfg

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

Le fichier /boot/grub2/device.map

```
root@debian11:~# cat /boot/grub/device.map
cat: /boot/grub/device.map: No such file or directory
root@debian11:~# grub-mkdevicemap
root@debian11:~# cat /boot/grub/device.map
(hd0)    /dev/disk/by-id/scsi-0QEMU_QEMU_HARDDISK_drive-scsi2
(hd1)    /dev/disk/by-id/scsi-0QEMU_QEMU_HARDDISK_drive-scsi0
(hd2)    /dev/disk/by-id/scsi-0QEMU_QEMU_HARDDISK_drive-scsi1
(hd3)    /dev/disk/by-id/lvm-pv-uuid-1J010Q-CM90-tKxI-0sM6-0vbe-3eDG-S10H6d
(hd4)    /dev/disk/by-id/lvm-pv-uuid-GEk0IP-S7ce-8S1G-K0TX-ocxE-Ud6y-IY3f0Z
(hd5)    /dev/disk/by-id/lvm-pv-uuid-J7UiEX-m983-j1fp-rU7x-TuCh-MFKh-s105M0
```

Le fichier /etc/default/grub

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

```
root@debian11:~# cat /etc/default/grub
```

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```



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

Dans ce fichier les directives les plus importantes sont :

Directive	Description
GRUB_DEFAULT	Entrée du menu sélectionner par défaut
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_CMDLINE_LINUX_DEFAULT	Paramètres passés au noyau lors d'un démarrage normal (Hors donc le mode secours)
GRUB_CMDLINE_LINUX	Paramètres passés au noyau peu importe le type de démarrage
GRUB_BADRAM	Permet de spécifier de la mémoire défaillante
GRUB_TERMINAL	Si non commentée, cette directive désactive le démarrage graphique
GRUB_GFXMODE	Indique la résolution utilisée lors d'un démarrage graphique
GRUB_DISABLE_LINUX_UUID	Si true , cette directive empêche l'utilisation de l'UUID de la partition
GRUB_DISABLE_LINUX_RECOVERY	Si true , cette directive empêche la génération des entrées en mode recovery
GRUB_INIT_TUNE	Permet d'obtenir un beep au démarrage de GRUB 2

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@debian11:~# ls -l /etc/grub.d
total 80
-rwxr-xr-x 1 root root 10046 Jul 11 2021 00_header
-rwxr-xr-x 1 root root 6260 Jul 11 2021 05_debian_theme
-rwxr-xr-x 1 root root 13664 Jul 11 2021 10_linux
-rwxr-xr-x 1 root root 13726 Jul 11 2021 20_linux_xen
-rwxr-xr-x 1 root root 12059 Jul 11 2021 30_os-prober
```

```
-rwxr-xr-x 1 root root 1416 Jul 11 2021 30_uefi-firmware
-rwxr-xr-x 1 root root 214 Jul 11 2021 40_custom
-rwxr-xr-x 1 root root 216 Jul 11 2021 41_custom
-rw-r--r-- 1 root root 483 Jul 11 2021 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.

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.



Important : Certaines distributions, telle qu'Ubuntu, cache le menu de GRUB 2 derrière une interface graphique. Afin de voir ce menu, il convient d'appuyer sur la touche **Shift** pendant que la machine démarre.

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 sous RHEL/CentOS 6

Un fichier Initramfs *INITial Ram File System* est une image d'un système minimal initialisée au démarrage du système.

Ce fichier utilise le système de fichier **cramFS** qui est un système de fichier compressé au format gzip et archivé via cpio.

L'image est chargée en mémoire vive et permet ainsi d'avoir un système minimal pouvant ensuite charger le système de fichier principal.

Examiner l'image existante

Pour examiner une image initramfs, il convient d'abord de la copier vers /tmp et de la décompresser :

```
[root@centos6 ~]# cp /boot/initramfs-2.6.32-71.29.1.el6.i686.img /tmp/custom.gz
[root@centos6 ~]# gunzip /tmp/custom.gz
```

Ensuite il convient d'extraire l'image grâce à la commande **cpio** :

```
[root@centos6 ~]# cd /tmp
[root@centos6 tmp]# mkdir initrd
[root@centos6 tmp]# cd initrd
[root@centos6 initrd]# cpio -cid -I ../custom
59631 blocs
```

Installez maintenant le paquet **tree** :

```
[root@centos6 initrd]# yum install tree
```

Utilisez maintenant la commande **tree** pour examiner le contenu de l'image :

```
[root@centos6 initrd]# tree | more
```

```
.
├── bin
│   ├── basename
│   ├── cat
│   ├── cp
│   ├── dash
│   ├── dd
│   ├── dmesg
│   ├── grep
│   ├── gzip
│   ├── ln
│   ├── loadkeys
│   ├── ls
│   ├── mkdir
│   ├── mknod
│   ├── mount
│   ├── mv
│   ├── plymouth
│   └── plymouthd
```

```
|  └─ readlink
|  └─ rm
|  └─ sed
|  └─ setfont
|  └─ sh -> dash
|  └─ sleep
|  └─ umount
|  └─ uname
└─ cmdline
--Plus--
```

Comme vous pouvez le constater, l'image contient une arborescence Linux minimaliste :

```
[root@centos6 initrd]# ls
bin      dev          emergency  init      initqueue-finished  lib    pre-pivot  pre-udev  sbin
sysroot  usr
cmdline  dracut-004-33.2.el6_0  etc      initqueue  initqueue-settled  mount  pre-trigger  proc      sys    tmp
var
```



A faire : Utilisez le manuel de la commande **cpio** pour comprendre les options utilisées.

Le script **init**

Le script **init** est lancé lors du chargement de l'image :

```
[root@centos6 initrd]# more init
#!/bin/sh
#
# Licensed under the GPLv2
#
```

```
# Copyright 2008-2009, Red Hat, Inc.
# Harald Hoyer <harald@redhat.com>
# Jeremy Katz <katzj@redhat.com>

wait_for_loginit()
{
    if getarg rdinitdebug; then
        set +x
        exec 0<>/dev/console 1<>/dev/console 2<>/dev/console
        # wait for loginit
        i=0
        while [ $i -lt 10 ]; do
            j=$(jobs)
            [ -z "$j" ] && break
            [ -z "${j##*Running*}" ] || break
            sleep 0.1
            i=$((i+1))
        done
        [ $i -eq 10 ] && kill %1 >/dev/null 2>&1

        while pidof -x /sbin/loginit >/dev/null 2>&1; do
            for pid in $(pidof -x /sbin/loginit); do
                kill $HARD $pid >/dev/null 2>&1
            done
        done
    --Plus-- (8%)
}
```



A faire : Passez en revue le contenu du script.

La Commande dracut

La commande **dracut** permet de créer facilement une image initramfs. Les options de la commande sont :

```
[root@centos6 initrd]# dracut --help
Usage: /sbin/dracut [OPTION]... <initramfs> <kernel-version>
Creates initial ramdisk images for preloading modules

-f, --force                Overwrite existing initramfs file.
-m, --modules [LIST]       Specify a space-separated list of dracut modules to
                             call when building the initramfs. Modules are located
                             in /usr/share/dracut/modules.d.
-o, --omit [LIST]          Omit a space-separated list of dracut modules.
-a, --add [LIST]           Add a space-separated list of dracut modules.
-d, --drivers [LIST]       Specify a space-separated list of kernel modules to
                             exclusively include in the initramfs.
--add-drivers [LIST]       Specify a space-separated list of kernel
                             modules to add to the initramfs.
--filesystems [LIST]       Specify a space-separated list of kernel filesystem
                             modules to exclusively include in the generic
                             initramfs.
-k, --kmoddir [DIR]        Specify the directory, where to look for kernel
                             modules
--fwdir [DIR]              Specify additional directories, where to look for
                             firmwares, separated by :
--kernel-only              Only install kernel drivers and firmware files
--no-kernel                Do not install kernel drivers and firmware files
--strip                    Strip binaries in the initramfs
--nostrip                  Do not strip binaries in the initramfs (default)
--mdadmconf                Include local /etc/mdadm.conf
--nomdadmconf              Do not include local /etc/mdadm.conf
--lvmconf                  Include local /etc/lvm/lvm.conf
--nolvmconf                Do not include local /etc/lvm/lvm.conf
-h, --help                 This message
--debug                    Output debug information of the build process
-v, --verbose               Verbose output during the build process
```

-c, --conf [FILE]	Specify configuration file to use. Default: /etc/dracut.conf
-l, --local	Local mode. Use modules from the current working directory instead of the system-wide installed in /usr/share/dracut/modules.d. Useful when running dracut from a git checkout.
-H, --hostonly	Host-Only mode: Install only what is needed for booting the local host instead of a generic host.
-i, --include [SOURCE] [TARGET]	Include the files in the SOURCE directory into the Target directory in the final initramfs.
-I, --install [LIST]	Install the space separated list of files into the initramfs.

Le fichier de configuration de dracut est **/etc/dracut.conf** :

```
[root@centos6 initrd]# cat /etc/dracut.conf
# Sample dracut config file

# Specific list of dracut modules to use
#dracutmodules+="

# Dracut modules to omit
#omit_dracutmodules+="

# Dracut modules to add to the default
#add_dracutmodules+="

# additional kernel modules to the default
add_drivers+="

# list of kernel filesystem modules to be included in the generic initramfs
#filesystems+="
```

```
# build initrd only to boot current hardware
#hostonly="yes"
#

# install local /etc/mdadm.conf
mdadmconf="yes"

# install local /etc/lvm/lvm.conf
lvmconf="yes"
```

Exécutez maintenant la commande suivante afin de générer le fichier **initramfs** :

```
# dracut -v initramfs [Entrée]
```

Notez la présence de votre nouvelle image **/tmp/initrd/initramfs**.

Déplacez votre fichier initramfs au répertoire /boot :

```
[root@centos6 initrd]# mv initramfs /boot
```

Editez maintenant votre fichier **/boot/grub/menu.lst** et **ajoutez** une **nouvelle section** qui utilise votre initramfs.

```
...
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS Linux (initramfs)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.29.1.el6.i686 ro root=UUID=e73735d6-c14b-4a40-8735-f34fc868da8a rd_NO_LUKS
rd_NO_LVM rd_NO_MD rd_NO_DM LANG=fr_FR.UTF-8 SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=fr-latin9
crashkernel=auto rhgb quiet
    initrd /initramfs
title CentOS Linux (2.6.32-71.29.1.el6.i686)
...
```



A faire : Re-démarrez votre machine pour tester votre configuration.

Initramfs sous Debian 11

Le fichier Initramfs *INITial Ram File System* est une image d'un système minimal initialisée au démarrage du système.

====LAB #1 - Travailler avec Initramfs

1.1 - Examiner l'image existante

Pour examiner une image initramfs, il convient d'abord de la copier vers /tmp et de la décompresser :

```
root@debian11:~# cp /boot/initrd.img-5.10.0-13-amd64 /tmp/custom.gz
root@debian11:~# gunzip /tmp/custom.gz
```

Ensuite il convient d'extraire l'image grâce à la commande **cpio** :

```
root@debian11:~# cd /tmp
root@debian11:/tmp# mkdir initrd
root@debian11:/tmp# cd initrd
root@debian11:/tmp/initrd# cpio -idvB < ../custom
...
var
var/cache
var/cache/fontconfig
var/cache/fontconfig/2cf0bf67-ac23-47f4-9d76-913901374fea-le64.cache-7
var/cache/fontconfig/42771ecd-ce02-464a-8263-bede4e63b34-le64.cache-7
var/cache/fontconfig/CACHEDIR.TAG
```

```
var/cache/fontconfig/ce5f225d-fe60-43b7-ad9d-497d8ccce03f-le64.cache-7  
var/cache/fontconfig/f8222393-3a24-44e9-bff6-c3f977041ad4-le64.cache-7  
31991 blocks
```

Installez maintenant le paquet **tree** :

```
root@debian11:/tmp/initrd# apt-get -y install tree
```

Utilisez maintenant la commande **tree** pour examiner le contenu de l'image :

```
root@debian11:/tmp/initrd# tree | more
```

```
.  
├── bin -> usr/bin  
├── conf  
│   ├── arch.conf  
│   ├── conf.d  
│   │   └── resume  
│   └── initramfs.conf  
├── cryptroot  
│   └── crypttab  
├── etc  
│   ├── console-setup  
│   │   ├── cached_UTF-8_del.kmap  
│   │   └── null  
│   ├── default  
│   │   └── keyboard  
│   ├── fonts  
│   │   ├── conf.d  
│   │   │   └── 60-latin.conf  
│   │   └── fonts.conf  
│   ├── fstab  
│   ├── ld.so.cache  
│   ├── ld.so.conf  
│   └── ld.so.conf.d
```

```
├── libc.conf
├── x86_64-linux-gnu.conf
├── lvm
│   ├── archive
│   │   ├── vg0_00000-267942700.vg
│   │   ├── vg0_00001-854434220.vg
│   │   ├── vg0_00002-520659205.vg
│   │   ├── vg0_00003-1606608177.vg
│   │   ├── vg0_00004-458787361.vg
│   │   ├── vg0_00005-1786773709.vg
│   │   ├── vg0_00006-196117920.vg
│   │   ├── vg0_00007-2024993792.vg
│   │   ├── vg0_00008-297779072.vg
│   │   ├── vg0_00009-1557237202.vg
│   │   ├── vg0_00010-550024633.vg
│   │   ├── vg0_00011-155655591.vg
│   │   └── vg0_00012-1101644815.vg
│   ├── backup
│   │   └── vg0
│   ├── lvm.conf
│   ├── lvmlocal.conf
│   └── profile
│       ├── cache-mq.profile
│       ├── cache-smq.profile
│       ├── command_profile_template.profile
│       ├── lvmdbusd.profile
│       ├── metadata_profile_template.profile
│       ├── thin-generic.profile
│       ├── thin-performance.profile
│       └── vdo-small.profile
├── modprobe.d
├── motd
├── mtab -> /proc/mounts
└── nsswitch.conf
```

```
| |— os-release
--More--
[q]
```

Comme vous pouvez le constater, l'image contient une arborescence Linux minimaliste :

```
root@debian11:/tmp/initrd# ls -l
total 36
lrwxrwxrwx 1 root root 7 Apr 30 11:02 bin -> usr/bin
drwxr-xr-x 3 root root 4096 Apr 30 11:02 conf
drwxr-xr-x 2 root root 4096 Apr 30 11:02 cryptroot
drwxr-xr-x 10 root root 4096 Apr 30 11:02 etc
-rwxr-xr-x 1 root root 6301 Apr 30 11:02 init
lrwxrwxrwx 1 root root 7 Apr 30 11:02 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Apr 30 11:02 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Apr 30 11:02 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Apr 30 11:02 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4096 Apr 30 11:02 run
lrwxrwxrwx 1 root root 8 Apr 30 11:02 sbin -> usr/sbin
drwxr-xr-x 10 root root 4096 Apr 30 11:02 scripts
drwxr-xr-x 10 root root 4096 Apr 30 11:02 usr
drwxr-xr-x 3 root root 4096 Apr 30 11:02 var
```

Le script **init** est lancé lors du chargement de l'image :

```
root@debian11:/tmp/initrd# more init
#!/bin/sh

# Default PATH differs between shells, and is not automatically exported
# by klibc dash. Make it consistent.
export PATH=/sbin:/usr/sbin:/bin:/usr/bin

[ -d /dev ] || mkdir -m 0755 /dev
[ -d /root ] || mkdir -m 0700 /root
```

```
[ -d /sys ] || mkdir /sys
[ -d /proc ] || mkdir /proc
[ -d /tmp ] || mkdir /tmp
mkdir -p /var/lock
mount -t sysfs -o nodev,noexec,nosuid sysfs /sys
mount -t proc -o nodev,noexec,nosuid proc /proc

# shellcheck disable=SC2013
for x in $(cat /proc/cmdline); do
    case $x in
        initramfs.clear)
            clear
            ;;
        quiet)
            quiet=y
            ;;
    esac
done

if [ "$quiet" != "y" ]; then
    quiet=n
    echo "Loading, please wait..."
fi
export quiet

# Note that this only becomes /dev on the real filesystem if udev's scripts
# are used; which they will be, but it's worth pointing out
mount -t devtmpfs -o nosuid,mode=0755 udev /dev
mkdir /dev/pts
mount -t devpts -o noexec,nosuid,gid=5,mode=0620 devpts /dev/pts || true

# Export the dpkg architecture
export DPKG_ARCH=
. /conf/arch.conf
```

```
# Set modprobe env
export MODPROBE_OPTIONS="-qb"

# Export relevant variables
export ROOT=
export ROOTDELAY=
export ROOTFLAGS=
export ROOTFSTYPE=
export IP=
export DEVICE=
export BOOT=
export BOOTIF=
export UBIMTD=
export break=
export init=/sbin/init
--More-- (19%)
```



A faire : Passez en revue le contenu du script.

1.2 - La commande mkinitramfs

La commande **mkinitramfs** permet de créer facilement une image initramfs. Les options de la commande sont :

```
root@debian11:~# mkinitramfs --help
```

```
Usage: mkinitramfs [option]... -o outfile [version]
```

Options:

```
-c compress  Override COMPRESS setting in initramfs.conf.
-d confdir   Specify an alternative configuration directory.
```

```
-k      Keep temporary directory used to make the image.
-o outfile Write to outfile.
-r root  Override R00T setting in initramfs.conf.
```

See `mkinitramfs(8)` for further details.

Le fichier de configuration de `mkinitramfs` est **/etc/initramfs-tools/modules**. Editez ce fichier pour spécifier des modules noyau supplémentaires à inclure dans le fichier image générée :

```
root@debian11:/tmp/initrd# vi /etc/initramfs-tools/modules
root@debian11:/tmp/initrd# cat /etc/initramfs-tools/modules
# List of modules that you want to include in your initramfs.
# They will be loaded at boot time in the order below.
#
# Syntax:  module_name [args ...]
#
# You must run update-initramfs(8) to effect this change.
#
# Examples:
#
# raid1
# sd_mod
usbcore
uhci
ehci-hcd
usb-ohci
usb-uhci
usb-storage
scsi_mod
sd_mod
```

Exécutez maintenant la commande suivante afin de générer le fichier **usbinitramfs** :

```
root@debian11:/tmp/initrd# mkinitramfs -o usbinitramfs-`uname -r`.img
```

Notez la présence de votre nouvelle image **/tmp/initrd/usbinitramfs** :

```
root@debian11:/tmp/initrd# ls -l /tmp/initrd/usbinitramfs-5.10.0-13-amd64.img
-rw-r--r-- 1 root root 47695493 Apr 30 11:12 /tmp/initrd/usbinitramfs-5.10.0-13-amd64.img
```

Déplacez votre fichier usbinitramfs au répertoire /boot :

```
root@debian11:/tmp/initrd# mv usbinitramfs-5.10.0-13-amd64.img /boot
```

Créez maintenant le fichier **/etc/grub.d/09_usbdebian** :

```
root@debian11:/tmp/initrd# vi /etc/grub.d/09_usbdebian
root@debian11:/tmp/initrd# cat /etc/grub.d/09_usbdebian
#!/bin/sh -e
cat << EOF
menuentry 'Debian GNU/Linux' --class debian --class gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-simple-9887a74f-a680-4bde-8f04-db5ae9ea186e' {
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    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 9887a74f-a680-4bde-8f04-db5ae9ea186e
    else
        search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
    fi
    echo    'Loading Linux 5.10.0-13-amd64 ...'
    linux   /boot/vmlinuz-5.10.0-13-amd64 root=UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e ro quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/usbinitramfs-5.10.0-13-amd64.img
}
```

EOF

Rendez ce fichier exécutable :

```
root@debian11:/tmp/initrd# chmod +x /etc/grub.d/09_usbdebian
```

Mettez à jour grub afin que celui-ci prend en compte le nouveau fichier :

```
root@debian11:/tmp/initrd# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-5.10.0-13-amd64
Found initrd image: /boot/initrd.img-5.10.0-13-amd64
done
```

Controlez le fichier /boot/grub/grub.cfg :

```
...
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/09_usbdebian ###
menuentry 'Debian GNU/Linux' --class debian --class gnu-linux --class gnu --class os 'gnulinux-simple-9887a74f-a680-4bde-8f04-db5ae9ea186e' {
    load_video
    insmod gzio
    if [ x = xxen ]; then insmod xzio; insmod lzopio; fi
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 9887a74f-a680-4bde-8f04-db5ae9ea186e
    else
        search --no-floppy --fs-uuid --set=root 9887a74f-a680-4bde-8f04-db5ae9ea186e
    fi
}
```

```
    fi
    echo    'Loading Linux 5.10.0-13-amd64 ...'
    linux   /boot/vmlinuz-5.10.0-13-amd64 root=UUID=9887a74f-a680-4bde-8f04-db5ae9ea186e ro  quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/usbinitramfs-5.10.0-13-amd64.img
}
### END /etc/grub.d/09_usbdebian ###

### BEGIN /etc/grub.d/10_linux ###
...
```

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.

Processus Init

Le premier processus lancé par le noyau est **init**. L'exécutable lancé est **/sbin/init**. Son rôle est de d'initialiser le système et de lancer certains autres services. Les tâches accomplies par init sont :

- le montage de /proc et de /sys,
- configuration des paramètres du noyau présents dans **/etc/sysctl.conf**,
- l'activation de SELinux,
- la mise à l'heure du système,
- la définition des consoles textes,
- la définition du nom de la machine,
- la détection des périphériques USB,
- la mise en place du support RAID et LVM,
- l'activation des quotas de disque,
- le montages des systèmes de fichiers,
- le re-montage du système de fichiers racine en lecture/écriture,
- l'activation du swap,
- le lancement de syslog,
- le chargement des modules du noyau,
- le nettoyage des fichiers temporaires,
- la définition des variables d'environnement tels PATH et RUNLEVEL

Démarrer RHEL/CentOS 5 avec SysVinit

Niveaux d'exécution sous RHEL/CentOS 5

Il existe 8 niveaux d'exécution ou **RUNLEVELS** sous RHEL/CentOS 5. Quatre des 8 sont réservés :

RUNLEVEL	Description
0	Arrêt de la machine
1	Mode mono-utilisateur pour la maintenance
6	Redémarrage de la machine
S ou s	Mode mono-utilisateur avec seul la partition racine montée

Les autres quatre RUNLEVELS sont définis par chaque distribution. Par exemple, sous RedHat, ils sont :

RUNLEVEL	Description
2	Mode multi-utilisateur sans NFS
3	Mode multi-utilisateur
4	Non-utilisé
5	Mode multi-utilisateur avec session graphique

Il existe aussi 3 pseudo-niveaux d'exécution **a**, **b** et **c**. Ces pseudo-niveaux permettent à init de faire quelque chose sans changer de niveau d'exécution.

Pour connaître le niveau d'exécution actuel de la machine, saisissez la commande suivante :

```
[root@centos5 ~]# runlevel
N 5
```

La lettre N indique que le système n'a pas changé de niveau d'exécution depuis son démarrage.

Pour modifier le niveau d'exécution courant, il convient d'utiliser la commande **init** ou **telinit** suivie du numéro du nouveau niveau d'exécution. Ces commandes peuvent prendre plusieurs options :

Option	Description
Q ou q	Demande à Init de relire le fichier /etc/inittab
-t	Permet de modifier le temps accordé par Init aux processus entre l'envoi du signal SIGTERM et l'envoi du signal SIGKILL

Inittab

Le fichier **/etc/inittab** permet de définir les services à démarrer en fonction du RUNLEVEL :

```
[root@centos5 ~]# cat /etc/inittab
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#
# Author:          Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#                  Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
```

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"


# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6


# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Dans l'exemple ci-dessus, chaque ligne non-commentée est composée de quatre champs, séparés par le caractère :

Champ	Nom	Description
1	Identifiant	Identifiant unique de la ligne composé de 1 à 4 caractères
2	RUNLEVELS	Liste des niveaux d'exécution concernés par la ligne
3	Action	Méthode utilisé pour lancer la commande se trouvant dans le champ 4
4	Commande	Commande à lancer

Le champ **action** prend une des directives suivantes :

Directive	Description
respawn	Le processus est relancé en cas d'arrêt de celui-ci
mingetty	Assure la gestion du terminal texte
once	Le processus n'est exécuté qu'une fois
wait	Le processus n'est exécuté qu'une fois. Init attend la fin du processus avant de passer à la ligne suivante
boot	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance
bootwait	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance. Init attend la fin du processus avant de passer à la ligne suivante
off	Revient à commenter la ligne
ondemand	La même chose que respawn mais la commande est exécuté dans un des 3 pseudo-niveaux d'exécution
initdefault	Définit le niveau d'exécution par défaut
sysinit	La commande est exécutée au démarrage de la machine avant les lignes boot et bootwait
powerfail	La commande est exécutée quand init reçoit un signal SIGPWR d'un onduleur
powerwait	La commande est exécutée quand init reçoit un signal SIGPWR d'un onduleur. Init attend la fin du processus avant de passer à la ligne suivante
powerokwait	La commande est exécutée si Init reçoit un signal de rétablissement du courant
powerfailnow	La commande est exécutée quand Init reçoit un signal comme quoi la batterie de l'onduleur est presque vide
ctrlaltdel	La commande est exécutée quand Init reçoit un signal SIGINT. Ce signal est envoyé par la combinaison de touches [CTRL] [ALT] [SUPPR]
kbrequest	La commande est exécutée suivant des séquences de touches saisies au clavier

L'analyse de notre fichier d'exemple sous RHEL/CentOS 5 indique :

Ligne	Description
id:5:initdefault:	Le niveau d'exécution par défaut est 5
si::sysinit:/etc/rc.d/rc.sysinit	La commande /etc/rc.d/rc.sysinit est lancée au démarrage de la machine
l0:0:wait:/etc/rc.d/rc 0	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 0
l1:1:wait:/etc/rc.d/rc 1	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 1
l2:2:wait:/etc/rc.d/rc 2	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 2
l3:3:wait:/etc/rc.d/rc 3	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 3
l4:4:wait:/etc/rc.d/rc 4	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 4
l5:5:wait:/etc/rc.d/rc 5	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 5
l6:6:wait:/etc/rc.d/rc 6	La commande /etc/rc.d/rc est lancée pour le niveau d'exécution 6

Ligne	Description
ca::ctrlaltdel:/sbin/shutdown -t3 -r now	La commande /sbin/shutdown -t3 -r now est lancée si les touches [CTRL] [ALT] [SUPPR] sont appuyées simultanément
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"	La commande /sbin/shutdown -f -h +2 est lancée quand Init reçoit le signal SIGPWR
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"	La commande /sbin/shutdown -c est lancée quand Init reçoit un signal de rétablissement du courant
1:2345:respawn:/sbin/mingetty tty1	Le terminal tty1 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F1]
2:2345:respawn:/sbin/mingetty tty2	Le terminal tty2 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F2]
3:2345:respawn:/sbin/mingetty tty3	Le terminal tty3 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F3]
4:2345:respawn:/sbin/mingetty tty4	Le terminal tty4 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F4]
5:2345:respawn:/sbin/mingetty tty5	Le terminal tty5 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F5]
6:2345:respawn:/sbin/mingetty tty6	Le terminal tty6 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F6]
x:5:respawn:/etc/X11/prefdm -nodaemon	xdm est lancé dans le niveau d'exécution 5

Scripts de Démarrage

Naviguez à **/etc/rc.d** et saisissez la commande **ls** :

```
[root@centos5 ~]# cd /etc/rc.d
[root@centos5 rc.d]# ls
init.d  rc0.d  rc2.d  rc4.d  rc6.d      rc.sysinit
rc      rc1.d  rc3.d  rc5.d  rc.local
```

rc.sysinit sous RHEL/CentOS 5

D'après l'étude du fichier **inittab** nous savons que le script rc.sysinit est exécuté en premier.

Répertoire init.d

Le répertoire **/etc/rc.d/init.d** contient les scripts permettant de lancer les services du système :

```
[root@centos5 rc.d]# ls init.d/*
init.d/abrtd          init.d/cpuspeed      init.d/iptables      init.d/netfs          init.d/portreserve
init.d/sandbox        init.d/vboxadd
init.d/acpid          init.d/crond         init.d/irqbalance    init.d/network        init.d/postfix
init.d/saslauthd      init.d/vboxadd-service
init.d/atd            init.d/cups          init.d/jexec          init.d/NetworkManager init.d/psacct
init.d/single         init.d/vboxadd-x11
init.d/auditd         init.d/dnsmasq       init.d/kdump          init.d/nfs             init.d/rdisc
init.d/smartd         init.d/wpa_supplicant
init.d/autofs         init.d/firstboot     init.d/killall        init.d/nfslock        init.d/restorecond
init.d/snmpd          init.d/ypbind
init.d/avahi-daemon   init.d/functions     init.d/lvm2-monitor   init.d/nscd           init.d/rpcbind
init.d/snmptrapd
init.d/bluetooth      init.d/haldaemon     init.d/mdmonitor      init.d/nslcd          init.d/rpcgssd
init.d/ssh
init.d/certmonger     init.d/halt          init.d/messagebus     init.d/ntpd           init.d/rpcidmapd
init.d/sss
init.d/cgconfig       init.d/httpd         init.d/microcode_ctl  init.d/ntpdate        init.d/rpcsvcgssd
init.d/sysstat
init.d/cgred          init.d/ip6tables     init.d/netconsole     init.d/odjjobd        init.d/rsyslog
init.d/udev-post
```

Répertoires rcx.d

Les répertoires **rc0.d** à **rc6.d** contiennent des liens vers les scripts du répertoire **init.d**.

Pour mieux comprendre, saisissez les commandes suivantes :

```
[root@centos5 rc.d]# for rep in rc[345].d; do echo "dans $rep :"; ls $rep/S*; done
```

dans rc3.d :

rc3.d/S00microcode_ctl	rc3.d/S10network	rc3.d/S13irqbalance	rc3.d/S24avahi-daemon	rc3.d/S25netfs
rc3.d/S30vboxadd	rc3.d/S80postfix			
rc3.d/S01sysstat	rc3.d/S11auditd	rc3.d/S13rpcbind	rc3.d/S24nfslock	rc3.d/S26acpid
rc3.d/S30vboxadd-x11	rc3.d/S82abrt			
rc3.d/S02lvm2-monitor	rc3.d/S11portreserve	rc3.d/S15mdmonitor	rc3.d/S24rpcgssd	rc3.d/S26haldaemon
rc3.d/S35vboxadd-service	rc3.d/S90crond			
rc3.d/S08ip6tables	rc3.d/S12rsyslog	rc3.d/S22messagebus	rc3.d/S24rpcidmapd	rc3.d/S26udev-post
rc3.d/S50bluetooth	rc3.d/S95atd			
rc3.d/S08iptables	rc3.d/S13cpuspeed	rc3.d/S23NetworkManager	rc3.d/S25cups	rc3.d/S28autofs
rc3.d/S55sshd	rc3.d/S99local			

dans rc4.d :

rc4.d/S00microcode_ctl	rc4.d/S10network	rc4.d/S13irqbalance	rc4.d/S24avahi-daemon	rc4.d/S25netfs
rc4.d/S30vboxadd	rc4.d/S90crond			
rc4.d/S01sysstat	rc4.d/S11auditd	rc4.d/S13rpcbind	rc4.d/S24nfslock	rc4.d/S26acpid
rc4.d/S35vboxadd-service	rc4.d/S95atd			
rc4.d/S02lvm2-monitor	rc4.d/S11portreserve	rc4.d/S15mdmonitor	rc4.d/S24rpcgssd	rc4.d/S26haldaemon
rc4.d/S50bluetooth	rc4.d/S99local			
rc4.d/S08ip6tables	rc4.d/S12rsyslog	rc4.d/S22messagebus	rc4.d/S24rpcidmapd	rc4.d/S26udev-post
rc4.d/S55sshd				
rc4.d/S08iptables	rc4.d/S13cpuspeed	rc4.d/S23NetworkManager	rc4.d/S25cups	rc4.d/S28autofs
rc4.d/S80postfix				

dans rc5.d :

rc5.d/S00microcode_ctl	rc5.d/S10network	rc5.d/S13irqbalance	rc5.d/S24avahi-daemon	rc5.d/S25netfs
rc5.d/S30vboxadd	rc5.d/S80postfix			
rc5.d/S01sysstat	rc5.d/S11auditd	rc5.d/S13rpcbind	rc5.d/S24nfslock	rc5.d/S26acpid
rc5.d/S30vboxadd-x11	rc5.d/S82abrt			
rc5.d/S02lvm2-monitor	rc5.d/S11portreserve	rc5.d/S15mdmonitor	rc5.d/S24rpcgssd	rc5.d/S26haldaemon
rc5.d/S35vboxadd-service	rc5.d/S90crond			
rc5.d/S08ip6tables	rc5.d/S12rsyslog	rc5.d/S22messagebus	rc5.d/S24rpcidmapd	rc5.d/S26udev-post
rc5.d/S50bluetooth	rc5.d/S95atd			

rc5.d/S08iptables	rc5.d/S13cpuspeed	rc5.d/S23NetworkManager	rc5.d/S25cups	rc5.d/S28autofs
rc5.d/S55sshd	rc5.d/S99local			



Important : Notez que chaque répertoire correspondant à un niveau d'exécution contient des liens pointant vers un script dans le répertoire **/etc/init.d**. La lettre **S** indique au script **rc** que le script dans **/etc/rc.d/init.d** doit être exécutée avec l'option **start**. De cette façon les processus sont lancés dans le niveau d'exécution spécifié. Le numéro qui suit la lettre **S** indique l'ordre de lancement par le script **rc**. Si deux scripts dans un répertoire **/etc/rc.d/rcX.d** ont le même numéro, l'ordre alphabétique prime. Notez aussi la présence du lien **S99local** qui lance le script **rc.local** en dernier. Le script **rc.local** est lancé dans les niveaux d'exécution **2, 3, 4 et 5**. C'est dans ce script que **root** peut ajouter des commandes.

Rappelez la commande précédente et modifiez la lettre S en **K** :

```
[root@centos5 rc.d]# for rep in rc[345].d; do echo "dans $rep :"; ls $rep/K*; done
dans rc3.d :
rc3.d/K01certmonger rc3.d/K10saslauthd rc3.d/K50snmpd rc3.d/K73ypbind rc3.d/K80kdump
rc3.d/K87restorecond rc3.d/K95firstboot
rc3.d/K01smartd rc3.d/K15httpd rc3.d/K50snmptrapd rc3.d/K74nscd rc3.d/K80sssd
rc3.d/K88nslcd
rc3.d/K02oddjobd rc3.d/K50dnsmasq rc3.d/K60nfs rc3.d/K74ntpd rc3.d/K84wpa_suppllicant
rc3.d/K89rdisc
rc3.d/K10psacct rc3.d/K50netconsole rc3.d/K69rpcsvcgssd rc3.d/K75ntpddate rc3.d/K86cgred
rc3.d/K95cgconfig
dans rc4.d :
rc4.d/K01certmonger rc4.d/K10saslauthd rc4.d/K50netconsole rc4.d/K69rpcsvcgssd rc4.d/K74ntpd
rc4.d/K84wpa_suppllicant rc4.d/K89rdisc
rc4.d/K01smartd rc4.d/K15httpd rc4.d/K50snmpd rc4.d/K70vboxadd-x11 rc4.d/K75ntpddate
rc4.d/K86cgred rc4.d/K95cgconfig
rc4.d/K02oddjobd rc4.d/K16abrtd rc4.d/K50snmptrapd rc4.d/K73ypbind rc4.d/K80kdump
```

```
rc4.d/K87restorecond      rc4.d/K95firstboot
rc4.d/K10psacct           rc4.d/K50dnsmasq      rc4.d/K60nfs           rc4.d/K74nscd           rc4.d/K80sssd
rc4.d/K88nslcd
dans rc5.d :
rc5.d/K01certmonger      rc5.d/K10saslauthd    rc5.d/K50snmpd         rc5.d/K73ypbind         rc5.d/K80kdump
rc5.d/K87restorecond     rc5.d/K95firstboot
rc5.d/K01smartd          rc5.d/K15httpd        rc5.d/K50snmptrapd     rc5.d/K74nscd           rc5.d/K80sssd
rc5.d/K88nslcd
rc5.d/K02oddjobd         rc5.d/K50dnsmasq      rc5.d/K60nfs           rc5.d/K74ntpd           rc5.d/K84wpa_suplicant
rc5.d/K89rdisc
rc5.d/K10psacct          rc5.d/K50netconsole   rc5.d/K69rpcsvcgssd    rc5.d/K75ntpdate        rc5.d/K86cgred
rc5.d/K95cgconfig
```



Important : Ici le principe est le même sauf que la lettre **K** indique au script **rc** que le script dans **/etc/rc.d/init.d** doit être lancé avec l'option **stop**.

rc.local

Le script rc.local est lancé dans les niveaux d'exécution **2, 3, 4 et 5**. C'est dans ce script que **root** peut ajouter des commandes.

La Commande chkconfig

Pour avoir une vue globale des services lancés par niveau d'exécution nous pouvons utiliser la commande **chkconfig**. Saisissez la commande suivante :

```
[root@centos5 rc.d]# chkconfig --list
NetworkManager 0:arrêt 1:arrêt 2:marche 3:marche 4:marche 5:marche 6:arrêt
abrtcd          0:arrêt 1:arrêt 2:arrêt 3:marche 4:arrêt 5:marche 6:arrêt
```

acpid	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
atd	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
auditd	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
autofs	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
avahi-daemon	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
bluetooth	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
certmonger	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
cgconfig	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
cgred	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
cpuspeed	0:arrêt	1:marche	2:marche	3:marche	4:marche	5:marche	6:arrêt
crond	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
cups	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
dnsmasq	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
firstboot	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
haldaemon	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
httpd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
ip6tables	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
iptables	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
irqbalance	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
kdump	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
lvm2-monitor	0:arrêt	1:marche	2:marche	3:marche	4:marche	5:marche	6:arrêt
mdmonitor	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
messagebus	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
microcode_ctl	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
netconsole	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
netfs	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
network	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
nfs	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
nfslock	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
nscd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
nsLCD	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
ntpd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
ntpddate	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
odddjobd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt

portreserve	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
postfix	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
psacct	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
rdisc	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
restorecond	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
rpcbind	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
rpcgssd	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
rpcidmapd	0:arrêt	1:arrêt	2:arrêt	3:marche	4:marche	5:marche	6:arrêt
rpcsvcgssd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
rsyslog	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
saslauthd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
smartd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
snmpd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
snmptrapd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
sshd	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
sssd	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
sysstat	0:arrêt	1:marche	2:marche	3:marche	4:marche	5:marche	6:arrêt
udev-post	0:arrêt	1:marche	2:marche	3:marche	4:marche	5:marche	6:arrêt
vboxadd	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
vboxadd-service	0:arrêt	1:arrêt	2:marche	3:marche	4:marche	5:marche	6:arrêt
vboxadd-x11	0:arrêt	1:arrêt	2:arrêt	3:marche	4:arrêt	5:marche	6:arrêt
wpa_supplicant	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt
ypbind	0:arrêt	1:arrêt	2:arrêt	3:arrêt	4:arrêt	5:arrêt	6:arrêt

Options de la commande

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

```
[root@centos5 ~]# chkconfig --help
chkconfig version 1.3.47 - Copyright (C) 1997-2000 Red Hat, Inc.
Ce logiciel peut être librement distribué selon les termes de la licence publique GNU (GPL).
```

```
utilisation :  chkconfig [--list] [--type <type>] [nom]
```

```
chkconfig --add <nom>
chkconfig --del <nom>
chkconfig --override <name>
chkconfig [--level <niveaux>] [--type <type>] <nom> <on|off|reset|resetpriorities>
```

Les options les plus importantes sont :

Option	Description
- -add (nom)	Permet d'ajouter un service aux runlevels définis par le service lui-même
- -del (nom)	Permet de supprimer un service aux runlevels définis par le service lui-même
- -level [niveau] (nom) [on off reset]	Permet d'activer, de désactiver ou de réinitialiser un service inscrit

Démarrer Debian 6 avec SysVinit

Niveaux d'exécution sous Debian 6

Il existe 8 niveaux d'exécution ou **RUNLEVELS** sous Linux. Quatre des 8 sont réservés :

RUNLEVEL	Description
0	Arrêt de la machine
1	Mode mono-utilisateur pour la maintenance
6	Redémarrage de la machine
S ou s	Mode mono-utilisateur avec seul la partition racine montée

Les autres quatre RUNLEVELS sont définis par chaque distribution. Par exemple, sous Debian, ils sont :

RUNLEVEL	Description
2	Mode multi-utilisateur
3	Mode multi-utilisateur - Non-utilisé
4	Mode multi-utilisateur - Non-utilisé
5	Mode multi-utilisateur - Non-utilisé

Il existe aussi 3 pseudo-niveaux d'exécution **a**, **b** et **c**. Ces pseudo-niveaux permettent à init de faire quelque chose sans changer de niveau d'exécution.

Pour connaître le niveau d'exécution actuel de la machine, saisissez la commande suivante :

```
root@debian6:~# runlevel
N 2
```

La lettre N indique que le système n'a pas changé de niveau d'exécution depuis son démarrage.

Pour modifier le niveau d'exécution courant, il convient d'utiliser la commande **init** ou **telinit** suivie du numéro du nouveau niveau d'exécution. Ces commandes peuvent prendre plusieurs options :

Option	Description
Q ou q	Demande à Init de relire le fichier /etc/inittab
-t	Permet de modifier le temps accordé par Init aux processus entre l'envoi du signal SIGTERM et l'envoi du signal SIGKILL

Inittab

Le fichier **/etc/inittab** permet de définir les services à démarrer en fonction du RUNLEVEL :

```
root@debian6:~# cat /etc/inittab
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $

# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
```

```
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Action on special keypress (ALT-UpArrow).
#kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab to let this work."

# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
```

```
# characters of the device (after "tty").
#
# Format:
# <id>:<runlevels>:<action>:<process>
#
# Note that on most Debian systems tty7 is used by the X Window System,
# so if you want to add more getty's go ahead but skip tty7 if you run X.
#
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
```

Dans l'exemple ci-dessus, chaque ligne non-commentée est composée de quatre champs, séparés par le caractère :

Champ	Nom	Description
1	Identifiant	Identifiant unique de la ligne composé de 1 à 4 caractères
2	RUNLEVELS	Liste des niveaux d'exécution concernés par la ligne
3	Action	Méthode utilisé pour lancer la commande se trouvant dans le champ 4
4	Commande	Commande à lancer

Le champ **action** prend une des directives suivantes :

Directive	Description
respawn	Le processus est relancé en cas d'arrêt de celui-ci
mingetty	Assure la gestion du terminal texte
once	Le processus n'est exécuté qu'une fois
wait	Le processus n'est exécuté qu'une fois. Init attend la fin du processus avant de passer à la ligne suivante
boot	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance
bootwait	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance. Init attend la fin du processus avant de passer à la ligne suivante
off	Revient à commenter la ligne
ondemand	La même chose que respawn mais la commande est exécuté dans un des 3 pseudo-niveaux d'exécution
initdefault	Définit le niveau d'exécution par défaut
sysinit	La commande est exécutée au démarrage de la machine avant les lignes boot et bootwait
powerfail	La commande est exécutée quand init reçoit un signal SIGPWR d'un onduleur
powerwait	La commande est exécutée quand init reçoit un signal SIGPWR d'un onduleur. Init attend la fin du processus avant de passer à la ligne suivante
powerokwait	La commande est exécutée si Init reçoit un signal de rétablissement du courant
powerfailnow	La commande est exécutée quand Init reçoit un signal comme quoi la batterie de l'onduleur est presque vide
ctrlaltdel	La commande est exécutée quand Init reçoit un signal SIGINT. Ce signal est envoyé par la combinaison de touches [CTRL] [ALT] [SUPPR]
kbrequest	La commande est exécutée suivant des séquences de touches saisies au clavier

L'analyse de notre fichier d'exemple sous Debian 6 indique :

Ligne	Description
id:2:initdefault:	Le niveau d'exécution par défaut est 2
si::sysinit:/etc/init.d/rcS	Le script /etc/init.d/rcS est lancé au démarrage de la machine
~~:S:wait:/sbin/sulogin	La commande /sbin/sulogin est lancée au démarrage de la machine en niveau d'exécution S
l0:0:wait:/etc/init.d/rc 0	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 0
l1:1:wait:/etc/init.d/rc 1	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 1
l2:2:wait:/etc/init.d/rc 2	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 2
l3:3:wait:/etc/init.d/rc 3	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 3
l4:4:wait:/etc/init.d/rc 4	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 4
l5:5:wait:/etc/init.d/rc 5	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 5

Ligne	Description
l6:6:wait:/etc/init.d/rc 6	Le script /etc/rc.d/rc est lancé pour le niveau d'exécution 6
z6:6:respawn:/sbin/sulogin	La commande /sbin/sulogin est lancée au redémarrage de la machine si nécessaire
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now	La commande /sbin/shutdown -t1 -a -r now est lancée si les touches [CTRL] [ALT] [SUPPR] sont appuyées simultanément
pf::powerwait:/etc/init.d/powerfail start	Le script /etc/init.d/powerfail start est lancée quand Init reçoit le signal SIGPWR
pn::powerfailnow:/etc/init.d/powerfail now	Le script /etc/init.d/powerfail now est lancée quand Init reçoit un signal comme quoi la batterie de l'onduleur est presque vide
po::powerokwait:/etc/init.d/powerfail stop	Le script /etc/init.d/powerfail stop est lancée quand Init reçoit un signal de rétablissement du courant
1:2345:respawn:/sbin/getty 38400 tty1	Le terminal tty1 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F1]
2:23:respawn:/sbin/getty 38400 tty2	Le terminal tty2 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F2]
3:23:respawn:/sbin/getty 38400 tty3	Le terminal tty3 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F3]
4:23:respawn:/sbin/getty 38400 tty4	Le terminal tty4 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F4]
5:23:respawn:/sbin/getty 38400 tty5	Le terminal tty5 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F5]
6:23:respawn:/sbin/getty 38400 tty6	Le terminal tty6 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F6]

Scripts de Démarrage

Le script rc.S sous Debian 6

D'après l'étude du fichier **inittab**, nous savons que le script /etc/init.d/rcS est exécuté en premier. Ce script appelle tous les scripts dans /etc/init.d qui sont référencés par les liens dans /etc/rcS.d/ et ceci dans un ordre numérique/alphabétique :

```
root@debian6:/etc/init.d# cat /etc/init.d/rcS
#!/bin/sh
#
```

```
# rcS
#
# Call all S??* scripts in /etc/rcS.d/ in numerical/alphabetical order
#

exec /etc/init.d/rc S
```

Répertoire init.d

Le répertoire **/etc/init.d** contient les scripts permettant de lancer les services du système :

```
root@debian6:/etc# cd init.d
root@debian6:/etc/init.d# ls
acpid          hwclockfirst.sh    rc.local
alsa-utils     hwclock.sh         rcS
anacron        ifupdown           README
atd            ifupdown-clean     reboot
avahi-daemon   kerneloops         rmnologin
binfmt-support keyboard-setup      rsyslog
bluetooth     killprocs          saned
bootlogd      lm-sensors         sendsigs
bootlogs      loadcpufreq        single
bootmisc.sh   module-init-tools  skeleton
checkfs.sh    mountall-bootclean.sh stop-bootlogd
checkroot.sh  mountall.sh        stop-bootlogd-single
console-screen.sh mountdevsubfs.sh  sudo
console-setup mountkernfs.sh     udev
cpufrequtils  mountnfs-bootclean.sh udev-mtab
cron          mountnfs.sh        umountfs
cups          mountoverflowtmp   umountnfs.sh
dbus         mtab.sh            umountroot
exim4        networking         unattended-upgrades
fancontrol   network-manager    urandom
```

fuse	nfs-common	vboxadd
gdm3	portmap	vboxadd-service
halt	pppd-dns	vboxadd-x11
hdparm	procps	x11-common
hostname.sh	rc	

rc.local

Le script rc.local est lancé dans les niveaux d'exécution **2, 3, 4 et 5**. C'est dans ce script que **root** peut ajouter des commandes.

La commande chkconfig

Sous Debian la commande **chkconfig** n'est pas installée par défaut. Il convient donc de l'installer en utilisant la commande **apt-get** :

```
root@debian6:/etc/init.d# apt-get install chkconfig
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  chkconfig
0 mis à jour, 1 nouvellement installés, 0 à enlever et 219 non mis à jour.
Il est nécessaire de prendre 9 182 o dans les archives.
Après cette opération, 69,6 ko d'espace disque supplémentaires seront utilisés.
Réception de : 1 http://ftp.fr.debian.org/debian/ squeeze/main chkconfig all 11.0-79.1-2 [9 182 B]
9 182 o réceptionnés en 50s (183 o/s)
(Lecture de la base de données... 130287 fichiers et répertoires déjà installés.)
Dépaquetage de chkconfig (à partir de .../chkconfig_11.0-79.1-2_all.deb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de chkconfig (11.0-79.1-2) ...
```

Saisissez maintenant la commande suivante :

```
root@debian6:/tmp# chkconfig --list
acpid                0:off  1:off  2:on   3:on   4:on   5:on   6:off
alsa-utils           0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
anacron              0:off  1:off  2:on   3:on   4:on   5:on   6:off
atd                  0:off  1:off  2:on   3:on   4:on   5:on   6:off
avahi-daemon         0:off  1:off  2:on   3:on   4:on   5:on   6:off
binfmt-support       0:off  1:off  2:on   3:on   4:on   5:on   6:off
bluetooth            0:off  1:off  2:on   3:on   4:on   5:on   6:off
bootlogd             0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
bootlogs             0:off  1:on   2:on   3:on   4:on   5:on   6:off
bootmisc.sh          0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
checkfs.sh           0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
checkroot.sh         0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
console-screen.sh    0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
console-setup        0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
cpufrequtils         0:off  1:off  2:on   3:on   4:on   5:on   6:off
cron                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
cups                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
dbus                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
exim4                0:off  1:off  2:on   3:on   4:on   5:on   6:off
fancontrol           0:off  1:off  2:on   3:on   4:on   5:on   6:off
fuse                 0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
gdm3                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
hdparm               0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
hostname.sh          0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
hwclock.sh           0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
hwclockfirst.sh      0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
ifupdown             0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
ifupdown-clean       0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
kerneloops           0:off  1:off  2:on   3:on   4:on   5:on   6:off
keyboard-setup       0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
killprocs            0:off  1:on   2:off  3:off  4:off  5:off  6:off
lm-sensors            0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
loadcpufreq          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

module-init-tools	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountall-bootclean.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountall.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountdevsubfs.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountkernfs.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountnfs-bootclean.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountnfs.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mountoverflowtmp	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
mtab.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
network-manager	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
networking	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
nfs-common	0:off	1:off	2:on	3:on	4:on	5:on	6:off	S:on
portmap	0:off	1:off	2:on	3:on	4:on	5:on	6:off	S:on
pppd-dns	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
procps	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
rc.local	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
rcS	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
rmnologin	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
rsyslog	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
saned	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
sendsigs	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
stop-bootlogd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
stop-bootlogd-single	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
sudo	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
udev	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
udev-mtab	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
umountfs	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
umountnfs.sh	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
umountroot	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
unattended-upgrades	0:off	1:off	2:off	3:off	4:off	5:off	6:off	
urandom	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
vboxadd	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
vboxadd-service	0:off	1:off	2:on	3:on	4:on	5:on	6:off	
vboxadd-x11	0:off	1:off	2:off	3:off	4:off	5:off	6:off	

x11-common	0:off	1:off	2:off	3:off	4:off	5:off	6:off	S:on
------------	-------	-------	-------	-------	-------	-------	-------	------

Options de la commande

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

```
root@debian6:/tmp# chkconfig --help
usage:
    chkconfig -A|--allservices           (together with -l: show all services)
    chkconfig -t|--terse [names]         (shows the links)
    chkconfig -e|--edit [names]          (configure services)
    chkconfig -s|--set [name state]...   (configure services)
    chkconfig -l|--list [--deps] [names] (shows the links)
    chkconfig -c|--check name [state]    (check state)
    chkconfig -a|--add [names]           (runs insserv)
    chkconfig -d|--del [names]           (runs insserv -r)
    chkconfig -h|--help                  (print usage)
    chkconfig -f|--force ...              (call insserv with -f)

    chkconfig [name]                     same as chkconfig -t
    chkconfig name state...               same as chkconfig -s name state
```

Les options les plus importantes sont :

Option	Description
- -add (nom)	Permet d'ajouter un service aux runlevels définis par le service lui-même
- -del (nom)	Permet de supprimer un service aux runlevels définis par le service lui-même
- -level [niveau] (nom) [on off reset]	Permet d'activer, de désactiver ou de réinitialiser un service inscrit

La commande **update-rc.d** sous Debian 6

La commande **update-rc.d** est utilisée pour gérer les liens dans les répertoires rc[x].d. Cette commande permet d'insérer les liens vers un script dans

/etc/init.d et prend la forme suivante :

```
update-rc.d <service> start <priorité de démarrage> <runlevels de démarrage> . stop <priorité d'arrêt> <runlevels d'arrêt> .
```

Par exemple la commande suivante crée les liens **S** dans rc2.d à rc5.d avec une priorité de 20 et les liens **K** dans rc0.d, rc1.d et rc6.d avec une priorité de 20 pour le script /etc/init.d/ssh :

```
# update-rc.d ssh start 20 2 3 4 5 . stop 20 0 1 6 . [Entrée]
```



Important : Il existe aussi un mot clef : **default**. Ce mot clef indique à **update-rc.d** d'utiliser les valeurs spécifiées dans la command précédente.

Pour supprimer les liens il convient d'utiliser la commande suivante :

```
# update-rc.d -f ssh remove [Entrée]
```



Important : Cette commande laisse le script ssh dans le répertoire /etc/init.d.

Options de la commande

Les options de cette commande sont :

```
root@debian:/etc/init.d# update-rc.d --help
update-rc.d: using dependency based boot sequencing
update-rc.d: error: --help
usage: update-rc.d [-n] [-f] <basename> remove
```

```
update-rc.d [-n] <basename> defaults [NN | SS KK]
update-rc.d [-n] <basename> start|stop NN runlvl [runlvl] [...] .
update-rc.d [-n] <basename> disable|enable [S|2|3|4|5]
-n: not really
-f: force
```

The disable|enable API is not stable and might change in the future.

Linux Standard Base

Linux Standard Base (LSB) fut introduit par le **Linux Foundation** dans un but de permettre la portabilité des scripts init entre distributions différentes.

Les scripts init qui sont conformes au standard LSB doivent fournir :

- au moins les actions **start**, **stop**, **restart**, **force-reload** et **status**,
- des codes retours standardisés,
- des informations sur des dépendances.

Les scripts init conformes au standard LSB peuvent aussi fournir :

- les actions **reload** et **try-restart**,
- des messages de journalisation en utilisant les fonctions Init.d **log_success_msg**, **log_failure_msg** et **log_warning_msg**.

Les fonctions proposées par défaut par LSB sont contenues dans le fichier **/lib/lsb/init-functions** :

```
[root@centos5 ~]# cat /lib/lsb/init-functions
#!/bin/sh

# LSB initscript functions, as defined in the LSB Spec 1.1.0
#
# Lawrence Lim <llim@redhat.com> - Tue, 26 June 2007
# Updated to the latest LSB 3.1 spec
```

```
# http://refspecs.freestdards.org/LSB_3.1.0/LSB-Core-generic/LSB-Core-generic_lines.txt

start_daemon () {
    /etc/redhat-lsb/lsb_start_daemon "$@"
}

killproc () {
    /etc/redhat-lsb/lsb_killproc "$@"
}

pidofproc () {
    /etc/redhat-lsb/lsb_pidofproc "$@"
}

log_success_msg () {
    /etc/redhat-lsb/lsb_log_message success "$@"
}

log_failure_msg () {
    /etc/redhat-lsb/lsb_log_message failure "$@"
}

log_warning_msg () {
    /etc/redhat-lsb/lsb_log_message warning "$@"
}
```

Ces fonctions sont des scripts se trouvant dans le répertoire **/etc/redhat-lsb** :

```
[root@centos5 ~]# ls -l /etc/redhat-lsb/
total 16
-rwxr-xr-x. 1 root root 70 22 févr. 2013 lsb_killproc
-rwxr-xr-x. 1 root root 243 22 févr. 2013 lsb_log_message
-rwxr-xr-x. 1 root root 59 22 févr. 2013 lsb_pidofproc
```

```
-rwxr-xr-x. 1 root root 650 22 févr. 2013 lsb_start_daemon
```

L'examen de ces scripts nous démontre que le système appelle le fichier **/etc/init.d/functions** :

```
[root@centos5 ~]# cat /etc/redhat-lsb/lsb_killproc
#!/bin/bash

. /etc/init.d/functions

LSB=LSB-1.1 killproc $*
exit $?

[root@centos ~]#
[root@centos ~]# cat /etc/redhat-lsb/lsb_pidofproc
#!/bin/bash

. /etc/init.d/functions

pidofproc $*
exit $?
```

Le fichier **/etc/init.d/functions** contient les fonctions à exécuter, par exemple :

```
...
# A function to stop a program.
killproc() {
    local RC killlevel= base pid pid_file= delay try binary=

    RC=0; delay=3; try=0
    # Test syntax.
    if [ "$#" -eq 0 ]; then
        echo $"Usage: killproc [-p pidfile] [ -d delay] {program} [-signal]"
        return 1
    fi
    if [ "$1" = "-p" ]; then
```

```

        pid_file=$2
        shift 2
    fi
    if [ "$1" = "-b" ]; then
        if [ -z $pid_file ]; then
            echo "$-b option can be used only with -p"
            echo "Usage: killproc -p pidfile -b binary program"
            return 1
        fi
        binary=$2
        shift 2
    fi
    if [ "$1" = "-d" ]; then
        delay=$(echo $2 | awk -v RS=' ' -v IGNORECASE=1 '{if($1!~/^[0-9.]+[smhd]?$/ ) exit
1;d=$1~/s$/^[0-9.]*$/?1:$1~/m$/?60:$1~/h$/?60*60:$1~/d$/?24*60*60:-1;if(d==-1) exit 1;delay+=d*$1} END
{printf("%d",delay+0.5)}}')
        if [ "$?" -eq 1 ]; then
            echo "Usage: killproc [-p pidfile] [-d delay] {program} [-signal]"
            return 1
        fi
        shift 2
    ...

```

et

```

...
pidofproc() {
    local RC pid pid_file=

    # Test syntax.
    if [ "$#" = 0 ]; then
        echo "Usage: pidofproc [-p pidfile] {program}"
        return 1
    fi

```

```
    if [ "$1" = "-p" ]; then
        pid_file=$2
        shift 2
    fi
    fail_code=3 # "Program is not running"

    # First try "/var/run/*.pid" files
    __pids_var_run "$1" "$pid_file"
    RC=$?
    if [ -n "$pid" ]; then
        echo $pid
        return 0
    fi

    [ -n "$pid_file" ] && return $RC
    __pids_pidof "$1" || return $RC
}
...
```

Codes Retour Standardisés

Les codes retour standardisés sont :

Code Retour	Description
0	Le programme fonctionne et le service est correctement démarré
1	Le programme est mort et le fichier pid dans /var/run existe
2	Le programme est mort et le fichier verrou dans /var/lock existe
3	Le programme ne fonctionne pas et le service n'est pas correctement démarré
4	Le statut du programme ou du service est inconnu
5 - 99	Réservés pour LSB
100-149	Réserves pour la distribution
150-199	Réservés pour l'application

Code Retour	Description
200-254	Réservés

Scripts

LSB stipule un format *rigide* de script qui commence par une section délimitée par deux clauses :

- **### BEGIN INIT INFO,**
- **### END INIT INFO.**

Par exemple :

```
[root@centos5 ~]# more /etc/rc.d/init.d/sshd
#!/bin/bash
#
# sshd      Start up the OpenSSH server daemon
#
# chkconfig: 2345 55 25
# description: SSH is a protocol for secure remote shell access. \
#              This service starts up the OpenSSH server daemon.
#
# processname: sshd
# config: /etc/ssh/ssh_host_key
# config: /etc/ssh/ssh_host_key.pub
# config: /etc/ssh/ssh_random_seed
# config: /etc/ssh/sshd_config
# pidfile: /var/run/sshd.pid

### BEGIN INIT INFO
# Provides: sshd
# Required-Start: $local_fs $network $syslog
# Required-Stop: $local_fs $syslog
# Should-Start: $syslog
# Should-Stop: $network $syslog
```

```
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start up the OpenSSH server daemon
# Description:      SSH is a protocol for secure remote shell access.
#                  This service starts up the OpenSSH server daemon.
### END INIT INFO

# source function library
. /etc/rc.d/init.d/functions

# pull in sysconfig settings
[ -f /etc/sysconfig/sshd ] && . /etc/sysconfig/sshd
--Plus-- (20%)
```

Les lignes se trouvant entre les deux clauses ont un format spécifique :

```
# {MotClef}: valeur1 [valeur2...]
```



Important : Notez qu'à part la deuxième ligne de la description, chaque ligne **doit** commencer par le caractère **#** suivi par un espace.

Les Mots Clefs sont :

Mot Clef	Description
Provides	Indique le service -(<i>boot facilities</i> en anglais) démarré par le script. Le nom doit être unique.
Required-start	Indique d'autres services qui doivent être démarrés avant le démarrage de celui indiqué par le mot-clef Provides .
Required-Stop	Indique d'autres services qui doivent être arrêtés après l'arrêt de celui indiqué par le mot-clef Provides .
Should-Start	Indique d'autres services qui, s'ils sont présents sur le système, être démarrés avant le démarrage de celui indiqué par le mot-clef Provides .
Should-Stop	Indique d'autres services qui, s'ils sont présents sur le système, doivent être arrêtés après l'arrêt de celui indiqué par le mot-clef Provides .

Mot Clef	Description
Default-Start	Indique les niveaux d'exécution dans lesquels le service doit être démarré.
Default-Stop	Indique les niveaux d'exécution dans lesquels le service doit être arrêté.
Short-Description	Indique une description du service en une seule ligne .
Description	Indique une description multi-lignes du service.

Il existe des groupements de services, appelés en anglais des *Virtual Facilities*. Les plus importants sont :

Virtual Facility	Description
\$local_fs	Tous les systèmes de fichiers locaux doivent être montés.
\$network	La carte Ethernet doit fonctionner
\$named	Les daemons, si présents, responsables de la résolution des noms tels DNS, NIS+ ou LDAP doivent être démarrés.
\$portmap	Les daemons qui fournissent le <i>SunRPC/ONCRPC port mapping</i> doivent être démarrés.
\$remote_fs	Tous les systèmes de fichiers doivent être montés.
\$syslog	Syslog, Syslog-ng ou Rsyslog doit être démarré.
\$time	L'heure du système doit avoir été fixé soit par NTP, soit par rdate soit par l'horloge système.

Démarrer RHEL/CentOS 6 avec Upstart

Upstart est un processus Init mieux adapté aux exigences des périphériques modernes de type plug'n'play. **Upstart** peut faire tout ce que fait le processus Init **SysVinit** mais supporte d'avantage d'**actions**. Par exemple, Upstart est capable de démarrer un service quand un périphérique spécifique est branché à chaud. Upstart est néanmoins compatible avec les scripts de démarrage du système **SysVinit** et ceux-ci fonctionnent normalement en attendant que les éditeurs de logiciels tiers portent les scripts SysVinit vers Upstart.

Inittab

Sous Upstart, le fichier `/etc/inittab` ne permet **plus** de définir les services à démarrer en fonction du RUNLEVEL, seule la définition d'**initdefault** y est spécifiée :

```
[root@centos6 ~]# cat /etc/inittab
```

```
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

```
[root@centos6 ~]# cat /etc/inittab
# inittab is only used by upstart for the default runlevel.
#
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# System initialization is started by /etc/init/rcS.conf
#
```

```
# Individual runlevels are started by /etc/init/rc.conf
#
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

Sous Upstart, les définitions des autres valeurs habituellement incluses dans `/etc/inittab` se trouvent dans des fichiers de configuration supplémentaires qui se trouvent dans le répertoire **/etc/init/** :

```
[root@centos6 ~]# ls -l /etc/init
total 68
-rw-r--r--. 1 root root 412 22 juil. 2014 control-alt-delete.conf
-rw-r--r--. 1 root root 130 12 mars 2014 init-system-dbus.conf
-rw-r--r--. 1 root root 463 22 juil. 2014 kexec-disable.conf
-rw-r--r--. 1 root root 560 22 juil. 2014 plymouth-shutdown.conf
-rw-r--r--. 1 root root 357 22 juil. 2014 prefdm.conf
-rw-r--r--. 1 root root 505 22 juil. 2014 quit-plymouth.conf
-rw-r--r--. 1 root root 417 22 juil. 2014 rc.conf
-rw-r--r--. 1 root root 1046 22 juil. 2014 rcS.conf
```

```
-rw-r--r--. 1 root root 430 22 juil. 2014 rcS-emergency.conf
-rw-r--r--. 1 root root 725 22 juil. 2014 rcS-sulogin.conf
-rw-r--r--. 1 root root 2915 23 nov. 2013 readahead-collector.conf
-rw-r--r--. 1 root root 1559 23 nov. 2013 readahead.conf
-rw-r--r--. 1 root root 726 23 nov. 2013 readahead-disable-services.conf
-rw-r--r--. 1 root root 1302 22 juil. 2014 serial.conf
-rw-r--r--. 1 root root 791 22 juil. 2014 splash-manager.conf
-rw-r--r--. 1 root root 473 22 juil. 2014 start-ttys.conf
-rw-r--r--. 1 root root 335 22 juil. 2014 tty.conf
```

Initialisation du Système

L'initialisation du système est configurée dans le fichier **/etc/init/rcS.conf** :

```
[root@centos6 init]# cat /etc/init/rcS.conf
# rcS - runlevel compatibility
#
# This task runs the old sysv-rc startup scripts.

start on startup

stop on runlevel

task

# Note: there can be no previous runlevel here, if we have one it's bad
# information (we enter rc1 not rcS for maintenance). Run /etc/rc.d/rc
# without information so that it defaults to previous=N runlevel=S.
console output
exec /etc/rc.d/rc.sysinit
post-stop script
    if [ "$UPSTART_EVENTS" = "startup" ]; then
        [ -f /etc/inittab ] && runlevel=$(/bin/awk -F ':' '$3 == "initdefault" && $1 !~ "^#" { print $2 }')
```

```
/etc/inittab)
    [ -z "$runlevel" ] && runlevel="3"
    for t in $(cat /proc/cmdline); do
        case $t in
            -s|single|S|s) runlevel="S" ;;
            [1-9])         runlevel="$t" ;;
            *)              ;;
        esac
    done
    exec telinit $runlevel
fi
end script
```

Runlevels

La gestion des Runlevels est configurée dans le fichier **/etc/init/rc.conf** :

```
[root@centos6 init]# cat /etc/init/rc.conf
# rc - System V runlevel compatibility
#
# This task runs the old sysv-rc runlevel scripts. It
# is usually started by the telinit compatibility wrapper.

start on runlevel [0123456]

stop on runlevel [!$RUNLEVEL]

task

export RUNLEVEL
console output
exec /etc/rc.d/rc $RUNLEVEL
```

[CTL]-[ALT]-[DEL]

Le comportement associé avec la combinaison de touches [CTL]-[ALT]-[DEL] est configuré dans le fichier **/etc/init/control-alt-delete.conf** :

```
[root@centos6 init]# cat /etc/init/control-alt-delete.conf
# control-alt-delete - emergency keypress handling
#
# This task is run whenever the Control-Alt-Delete key combination is
# pressed. Usually used to shut down the machine.

start on control-alt-delete

exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

mingetty

Le génération des terminaux getty est spécifiée dans les fichiers **/etc/init/tty.conf** et **/etc/init/serial.conf** :

```
[root@centos6 init]# cat /etc/init/tty.conf
# tty - getty
#
# This service maintains a getty on the sepcified device.

stop on runlevel [016]

respawn
instance $TTY
exec /sbin/mingetty $TTY
[root@centos rc.d]# cat /etc/init/serial.conf
# Automatically start a configured serial console
#
# How this works:
```

```
#
# On boot, a udev helper examines /dev/console. If a serial console is the
# primary console (last console on the commandline in grub), the event
# 'fedora.serial-console-available <port name> <speed>' is emitted, which
# triggers this script. It waits for the runlevel to finish, ensures
# the proper port is in /etc/securetty, and starts the getty.
#
# If your serial console is not the primary console, or you want a getty
# on serial even if it's not the console, create your own event by copying
# /etc/init/tty.conf, and changing the getty line in that file.

start on fedora.serial-console-available DEV=* and stopped rc RUNLEVEL=[2345]
stop on runlevel [016]

instance $DEV
respawn
pre-start exec /sbin/securetty $DEV
exec /sbin/agetty /dev/$DEV $SPEED vt100-nav
```

La configuration des terminaux fait partie du fichier **/etc/sysconfig/init**

```
[root@centos6 init]# cat /etc/sysconfig/init
# color => new RH6.0 bootup
# verbose => old-style bootup
# anything else => new style bootup without ANSI colors or positioning
BOOTUP=color
# column to start "[ OK ]" label in
RES_COL=60
# terminal sequence to move to that column. You could change this
# to something like "tput hpa ${RES_COL}" if your terminal supports it
MOVE_TO_COL="echo -en \\033[${RES_COL}G"
# terminal sequence to set color to a 'success' color (currently: green)
SETCOLOR_SUCCESS="echo -en \\033[0;32m"
# terminal sequence to set color to a 'failure' color (currently: red)
```

```
SETCOLOR_FAILURE="echo -en \\033[0;31m"  
# terminal sequence to set color to a 'warning' color (currently: yellow)  
SETCOLOR_WARNING="echo -en \\033[0;33m"  
# terminal sequence to reset to the default color.  
SETCOLOR_NORMAL="echo -en \\033[0;39m"  
# Set to anything other than 'no' to allow hotkey interactive startup...  
PROMPT=yes  
# Set to 'yes' to allow probing for devices with swap signatures  
AUTOSWAP=no  
# What ttys should gettys be started on?  
ACTIVE_CONSOLES=/dev/tty[1-6]  
# Set to '/sbin/sulogin' to prompt for password on single-user mode  
# Set to '/sbin/sushell' otherwise  
SINGLE=/sbin/sushell
```

Gestion des Services

Sous Upstart, les services sont appelés des **jobs**. Les scripts de démarrage de jobs au format Upstart sont placés dans le répertoire **/etc/init/** et ont une forme **nom.conf** où *nom* est le nom du job :

```
[root@centos6 ~]# ls /etc/init  
control-alt-delete.conf  rcS-sulogin.conf  
init-system-dbus.conf    readahead-collector.conf  
kexec-disable.conf       readahead.conf  
plymouth-shutdown.conf   readahead-disable-services.conf  
prefdm.conf              serial.conf  
quit-plymouth.conf        splash-manager.conf  
rc.conf                  start-ttys.conf  
rcS.conf                 tty.conf  
rcS-emergency.conf
```

Pour créer un fichier job, il convient de respecter un certain format. Par exemple créez le fichier **/etc/init/testjob.conf** :

[testjob.conf](#)

```
description "Un job pour tester Upstart"  
author "Linux E-Learning"  
start on runlevel [2345]  
exec echo Le job test a été lancé le `date` >> /var/log/testjob.log
```

La Commande initctl

Pour obtenir une liste de tous les jobs et leurs états, il convient d'utiliser la commande `initctl`. Vérifiez donc que le job `testjob` se trouve dans la sortie de cette commande :

```
[root@centos6 ~]# initctl list  
rc stop/waiting  
tty (/dev/tty3) start/running, process 1833  
tty (/dev/tty2) start/running, process 1828  
tty (/dev/tty6) start/running, process 1854  
tty (/dev/tty5) start/running, process 1845  
tty (/dev/tty4) start/running, process 1838  
plymouth-shutdown stop/waiting  
control-alt-delete stop/waiting  
rcS-emergency stop/waiting  
readahead-collector stop/waiting  
kexec-disable stop/waiting  
quit-plymouth stop/waiting  
testjob stop/waiting  
rcS stop/waiting  
prefdm start/running, process 1814  
init-system-dbus stop/waiting  
readahead stop/waiting  
splash-manager stop/waiting  
start-ttys stop/waiting
```

```
readahead-disable-services stop/waiting
rcS-sulogin stop/waiting
serial stop/waiting
```

Option de la Commande **initctl**

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

```
[root@centos6 ~]# initctl --help
Usage: initctl [OPTION]... COMMAND [OPTION]... [ARG]...

Options:
  --system          use D-Bus system bus to connect to init daemon
  --dest=NAME       destination well-known name on system bus
  -q, --quiet       reduce output to errors only
  -v, --verbose     increase output to include informational messages
  --help           display this help and exit
  --version        output version information and exit
```

For a list of commands, try `initctl help`.

Report bugs at <<https://launchpad.net/upstart/+bugs>>

Jobs

Il existe trois types de jobs sous Upstart :

- **task**
- **service**
 - un service job peut fonctionner en arrière plan
- **abstract**

- un abstract job est un service qui fonctionnent jusqu'à ce que l'administrateur l'arrête

Événements

Un événement ou *event* en anglais est un signal envoyé vers un job pour déclencher une action. Par exemple :

- **starting**
- **started**
- **stopping**
- **stopped**

Etats

L'objectif d'un job est de démarrer ou de s'arrêter. Entre ces deux objectifs se trouvent des états intermédiaires :

- **waiting** - l'état initial d'un job
- **starting** - le job est sur le point de démarrer
- **pre-start** - la section pre-start est chargée
- **spawned** - une section script est sur le point de démarrer
- **post-start** - les opérations détaillées dans la section post-start ont lieu
- **running** - le job est opérationnel
- **pre-stop** - la section pre-stop est chargée
- **stopping** - le job est en cours d'arrêt
- **killed** - le job est arrêté
- **post-stop** - les opérations détaillées dans la section post-stop ont lieu

Démarrer et Arrêter les Jobs

Compte tenu de la compatibilité avec le système SysVinit, les commandes traditionnellement utilisées avec ce dernier sont compris par Upstart. Ceci étant Upstart fournit ses propres commandes pour le contrôle des jobs.

La Commande status

Pour voir le status d'un job spécifique, utilisez la commande suivante :

```
[root@centos6 ~]# status testjob
testjob stop/waiting
```

Options de la Commande status

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

```
[root@centos6 ~]# status --help
Usage: status [OPTION]... JOB [KEY=VALUE]...
Query status of job.

Options:
  --system                use D-Bus system bus to connect to init daemon
  --dest=NAME             destination well-known name on system bus
  -q, --quiet             reduce output to errors only
  -v, --verbose           increase output to include informational messages
  --help                 display this help and exit
  --version               output version information and exit
```

JOB is the name of the job that is to be queried, this may be followed by zero or more environment variables to distinguish between job instances.

Report bugs at <<https://launchpad.net/upstart/+bugs>>

La Commande start

```
[root@centos6 ~]# start testjob
testjob start/running, process 2981
[root@centos ~]# cat /var/log/testjob.log
Le job test a été lancé le Tue Jun 30 15:26:10 CEST 2015
```

Options de la Commande start

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

```
[root@centos6 ~]# start --help
Usage: start [OPTION]... JOB [KEY=VALUE]...
Start job.

Options:
  -n, --no-wait          do not wait for job to start before exiting
  --system              use D-Bus system bus to connect to init daemon
  --dest=NAME           destination well-known name on system bus
  -q, --quiet            reduce output to errors only
  -v, --verbose          increase output to include informational messages
  --help                display this help and exit
  --version             output version information and exit
```

JOB is the name of the job that is to be started, this may be followed by zero or more environment variables to be defined in the new job.

The environment may also serve to distinguish between job instances, and thus decide whether a new instance will be started or an error returned if an existing instance is already running.

Report bugs at <<https://launchpad.net/upstart/+bugs>>

La Commande stop

Pour arrêter un job, utilisez la commande **stop**.

Options de la Commande stop

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

```
[root@centos6 ~]# stop --help
Usage: stop [OPTION]... JOB [KEY=VALUE]...
Stop job.
```

Options:

-n, --no-wait	do not wait for job to stop before exiting
--system	use D-Bus system bus to connect to init daemon
--dest=NAME	destination well-known name on system bus
-q, --quiet	reduce output to errors only
-v, --verbose	increase output to include informational messages
--help	display this help and exit
--version	output version information and exit

JOB is the name of the job that is to be stopped, this may be followed by zero or more environment variables to be passed to the job's pre-stop and post-stop processes.

The environment also serves to distinguish between job instances, and thus decide which of multiple instances will be stopped.

Report bugs at <<https://launchpad.net/upstart/+bugs>>

La Commande restart

Pour redémarrer un job, utilisez la commande **restart**.

Options de la Commande restart

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

```
[root@centos6 ~]# restart --help
Usage: restart [OPTION]... JOB [KEY=VALUE]...
Restart job.

Options:
  -n, --no-wait          do not wait for job to restart before exiting
  --system              use D-Bus system bus to connect to init daemon
  --dest=NAME           destination well-known name on system bus
  -q, --quiet            reduce output to errors only
  -v, --verbose          increase output to include informational messages
  --help                display this help and exit
  --version             output version information and exit
```

JOB is the name of the job that is to be restarted, this may be followed by zero or more environment variables to be defined in the job after restarting.

The environment also serves to distinguish between job instances, and thus decide which of multiple instances will be restarted.

Report bugs at <<https://launchpad.net/upstart/+bugs>>



Il est aussi possible d'utiliser les commandes **initctl start**, **initctl stop** et **initctl restart** pour gérer les jobs.

Démarrer Debian 11 avec 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@debian11:/tmp/initrd# ls -l /usr/lib/systemd/system/runlevel*
lrwxrwxrwx 1 root root 15 Mar 20 20:55 /usr/lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Mar 20 20:55 /usr/lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Mar 20 20:55 /usr/lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Mar 20 20:55 /usr/lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Mar 20 20:55 /usr/lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Mar 20 20:55 /usr/lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Mar 20 20:55 /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 #2 - La Commande systemctl

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

```
root@debian11:/tmp/initrd# cd ~
root@debian11:~# 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 QEMU_DVD-ROM Debian_11.3.0_amd64_n			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:0-block-sdb-sdb1.device	loaded	active	
plugged QEMU_HARDDISK 1			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:0-block-sdb-sdb2.device	loaded	active	
plugged QEMU_HARDDISK 2			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:0-block-sdb-sdb5.device	loaded	active	
plugged QEMU_HARDDISK 5			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:0-block-sdb.device	loaded	active	
plugged QEMU_HARDDISK			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc1.device	loaded	active	
plugged QEMU_HARDDISK 1			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc10.device	loaded	active	
plugged QEMU_HARDDISK 10			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc11.device	loaded	active	
plugged QEMU_HARDDISK 11			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc12.device	loaded	active	
plugged QEMU_HARDDISK 12			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc2.device	loaded	active	
plugged QEMU_HARDDISK 2			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc3.device	loaded	active	
plugged QEMU_HARDDISK 3			
sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc4.device	loaded	active	

plugged	QEMU_HARDDISK 4		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc5.device	loaded	active
plugged	QEMU_HARDDISK 5		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc6.device	loaded	active
plugged	QEMU_HARDDISK 6		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc7.device	loaded	active
plugged	QEMU_HARDDISK 7		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc8.device	loaded	active
plugged	QEMU_HARDDISK my_reiserfs		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc-sdc9.device	loaded	active
plugged	QEMU_HARDDISK 9		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:1-block-sdc.device	loaded	active
plugged	QEMU_HARDDISK		
	sys-devices-pci0000:00-0000:00:05.0-virtio1-host2-target2:0:0-2:0:0:2-block-sda.device	loaded	active
plugged	QEMU_HARDDISK		
	sys-devices-pci0000:00-0000:00:12.0-virtio2-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-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-ens18.device	loaded active
plugged	Virtio network device	
	-.mount	loaded active
mounted	Root Mount	
	dev-hugepages.mount	loaded active
mounted	Huge Pages File System	
	dev-mqueue.mount	loaded active
mounted	POSIX Message Queue File System	
	mnt-sdc11.mount	loaded active
mounted	/mnt/sdc11	
	mnt-sdc12.mount	loaded active
mounted	/mnt/sdc12	
	run-user-1000.mount	loaded active
mounted	/run/user/1000	
	run-user-113.mount	loaded active
mounted	/run/user/113	
	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-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-180.scope                                loaded active
running  Session 180 of user trainee
         session-cl.scope                                loaded active
running  Session cl of user lightdm
         apparmor.service                                loaded active
exited   Load AppArmor profiles
         auditd.service                                  loaded active
running  Security Auditing Service
         avahi-daemon.service                            loaded active
running  Avahi mDNS/DNS-SD Stack
         blk-availability.service                        loaded active
exited   Availability of block devices
         console-setup.service                          loaded active
exited   Set console font and keymap
         cron.service                                    loaded active
running  Regular background program processing daemon
         cups-browsed.service                            loaded active
running  Make remote CUPS printers available locally
         cups.service                                    loaded active
running  CUPS Scheduler
lines 1-58
[q]
```

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

```
root@debian11:~# systemctl list-units --all | grep inactive | more
● home.mount
not-found inactive dead      home.mount
    proc-sys-fs-binfmt_misc.mount
loaded    inactive dead      Arbitrary Executable File Formats File System
● tmp.mount
not-found inactive dead      tmp.mount
    systemd-ask-password-console.path
```

loaded	inactive	dead	Dispatch Password Requests to Console Directory Watch
anacron.service			
loaded	inactive	dead	Run anacron jobs
apt-daily-upgrade.service			
loaded	inactive	dead	Daily apt upgrade and clean activities
apt-daily.service			
loaded	inactive	dead	Daily apt download activities
● connman.service			
not-found	inactive	dead	connman.service
● console-screen.service			
not-found	inactive	dead	console-screen.service
dm-event.service			
loaded	inactive	dead	Device-mapper event daemon
e2scrub_all.service			
loaded	inactive	dead	Online ext4 Metadata Check for All Filesystems
e2scrub_reap.service			
loaded	inactive	dead	Remove Stale Online ext4 Metadata Check Snapshots
emergency.service			
loaded	inactive	dead	Emergency Shell
● fcoe.service			
not-found	inactive	dead	fcoe.service
fstrim.service			
loaded	inactive	dead	Discard unused blocks on filesystems from /etc/fstab
getty-static.service			
loaded	inactive	dead	getty on tty2-tty6 if dbus and logind are not available
● iscsi-shutdown.service			
not-found	inactive	dead	iscsi-shutdown.service
● iscsi.service			
not-found	inactive	dead	iscsi.service
● iscsid.service			
not-found	inactive	dead	iscsid.service
● kbd.service			
not-found	inactive	dead	kbd.service
logrotate.service			

```
loaded    inactive dead    Rotate log files
● lvm2-activation.service
not-found inactive dead    lvm2-activation.service
  lvm2-lvmpolld.service
loaded    inactive dead    LVM2 poll daemon
  man-db.service
loaded    inactive dead    Daily man-db regeneration
  mlocate.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
  modprobe@fuse.service
loaded    inactive dead    Load Kernel Module fuse
● nslcd.service
not-found inactive dead    nslcd.service
  plymouth-quit.service
loaded    inactive dead    Terminate Plymouth Boot Screen
● rbdmap.service
not-found inactive dead    rbdmap.service
  rc-local.service
loaded    inactive dead    /etc/rc.local Compatibility
  rescue.service
loaded    inactive dead    Rescue Shell
  rsync.service
loaded    inactive dead    fast remote file copy program daemon
● sssd.service
not-found inactive dead    sssd.service
  systemd-ask-password-console.service
loaded    inactive dead    Dispatch Password Requests to Console
  systemd-ask-password-plymouth.service
loaded    inactive dead    Forward Password Requests to Plymouth
  systemd-ask-password-wall.service
```

```
loaded    inactive dead    Forward Password Requests to Wall
systemd-binsfmt.service
loaded    inactive dead    Set Up Additional Binary Formats
systemd-boot-system-token.service
loaded    inactive dead    Store a System Token in an EFI Variable
systemd-cryptsetup@sdcl1.service
loaded    inactive dead    Cryptography Setup for sdcl1
systemd-fsck-root.service
loaded    inactive dead    File System Check on Root Device
systemd-fsck@dev-mapper-sdcl1.service
loaded    inactive dead    File System Check on /dev/mapper/sdcl1
systemd-fsckd.service
loaded    inactive dead    File System Check Daemon to report status
systemd-hwdb-update.service
loaded    inactive dead    Rebuild Hardware Database
systemd-initctl.service
loaded    inactive dead    initctl Compatibility Daemon
systemd-machine-id-commit.service
loaded    inactive dead    Commit a transient machine-id on disk
systemd-networkd.service
loaded    inactive dead    Network Service
systemd-pstore.service
loaded    inactive dead    Platform Persistent Storage Archival
systemd-quotacheck.service
loaded    inactive dead    File System Quota Check
systemd-rfkill.service
loaded    inactive dead    Load/Save RF Kill Switch Status
systemd-tmpfiles-clean.service
loaded    inactive dead    Cleanup of Temporary Directories
● systemd-update-done.service
not-found inactive dead    systemd-update-done.service
systemd-update-utmp-runlevel.service
loaded    inactive dead    Update UTMP about System Runlevel Changes
● systemd-vconsole-setup.service
```

```
not-found inactive dead      systemd-vconsole-setup.service
● ypbind.service
not-found inactive dead      ypbind.service
  system-systemd\x2dcryptsetup.slice
loaded    inactive dead      Cryptsetup Units Slice
  system-systemd\x2dfsck.slice
loaded    inactive dead      system-systemd\x2dfsck.slice
--More--
[q]
```

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@debian11:~# systemctl status connman
Unit connman.service could not be found.
```

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

```
root@debian11:~# systemctl list-unit-files | more
```

UNIT FILE	STATE	VENDOR PRESET
proc-sys-fs-binfmt_misc.automount	static	-
-.mount	generated	-
dev-hugepages.mount	static	-
dev-mqueue.mount	static	-
media-cdrom0.mount	generated	-
mnt-sdc11.mount	generated	-
proc-sys-fs-binfmt_misc.mount	disabled	disabled
sys-fs-fuse-connections.mount	static	-
sys-kernel-config.mount	static	-
sys-kernel-debug.mount	static	-
sys-kernel-tracing.mount	static	-
cups.path	enabled	enabled
systemd-ask-password-console.path	static	-
systemd-ask-password-plymouth.path	static	-

systemd-ask-password-wall.path	static	-
session-180.scope	transient	-
session-cl.scope	transient	-
alsa-restore.service	static	-
alsa-state.service	static	-
alsa-utils.service	masked	enabled
anacron.service	enabled	enabled
apparmor.service	enabled	enabled
apt-daily-upgrade.service	static	-
apt-daily.service	static	-
auditd.service	enabled	enabled
autovt@.service	alias	-
avahi-daemon.service	enabled	enabled
blk-availability.service	enabled	enabled
colord.service	static	-
configure-printer@.service	static	-
console-getty.service	disabled	disabled
console-setup.service	enabled	enabled
container-getty@.service	static	-
cron.service	enabled	enabled
cryptdisks-early.service	masked	enabled
cryptdisks.service	masked	enabled
cups-browsed.service	enabled	enabled
cups.service	enabled	enabled
dbus-fi.wl.wpa_supplicant1.service	alias	-
dbus-org.freedesktop.Avahi.service	alias	-
dbus-org.freedesktop.hostname1.service	alias	-
dbus-org.freedesktop.locale1.service	alias	-
dbus-org.freedesktop.login1.service	alias	-
dbus-org.freedesktop.ModemManager1.service	alias	-
dbus-org.freedesktop.nm-dispatcher.service	alias	-
dbus-org.freedesktop.timedate1.service	alias	-
dbus-org.freedesktop.timesync1.service	alias	-
dbus.service	static	-

debug-shell.service	disabled	disabled
display-manager.service	alias	-
dm-event.service	static	-
e2scrub@.service	static	-
e2scrub_all.service	static	-
e2scrub_fail@.service	static	-
e2scrub_reap.service	enabled	enabled
emergency.service	static	-
fstrim.service	static	-
--More--		
[q]		

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

```
root@debian11:~# systemctl list-unit-files -t mount
UNIT FILE                STATE    VENDOR PRESET
-.mount                  generated -
dev-hugepages.mount      static   -
dev-mqueue.mount         static   -
media-cdrom0.mount       generated -
mnt-sdc11.mount          generated -
proc-sys-fs-binfmt_misc.mount disabled disabled
sys-fs-fuse-connections.mount static    -
sys-kernel-config.mount  static   -
sys-kernel-debug.mount   static   -
sys-kernel-tracing.mount static    -

10 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@debian11:~# ls -l /lib/systemd/system-generators/systemd-fstab-generator
-rwxr-xr-x 1 root root 43440 Mar 20 20:55 /lib/systemd/system-generators/systemd-fstab-generator
```

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

```
root@debian11:~# ls -l /lib/systemd/system-generators
total 420
-rwxr-xr-x 1 root root 146872 Feb 22 2021 lvm2-activation-generator
-rwxr-xr-x 1 root root 14488 Mar 20 20:55 systemd-bless-boot-generator
-rwxr-xr-x 1 root root 35176 Mar 20 20:55 systemd-cryptsetup-generator
-rwxr-xr-x 1 root root 14632 Mar 20 20:55 systemd-debug-generator
-rwxr-xr-x 1 root root 43440 Mar 20 20:55 systemd-fstab-generator
-rwxr-xr-x 1 root root 14480 Mar 20 20:55 systemd-getty-generator
-rwxr-xr-x 1 root root 34968 Mar 20 20:55 systemd-gpt-auto-generator
-rwxr-xr-x 1 root root 14624 Mar 20 20:55 systemd-hibernate-resume-generator
-rwxr-xr-x 1 root root 14480 Mar 20 20:55 systemd-rc-local-generator
-rwxr-xr-x 1 root root 14616 Mar 20 20:55 systemd-run-generator
-rwxr-xr-x 1 root root 14480 Mar 20 20:55 systemd-system-update-generator
-rwxr-xr-x 1 root root 30936 Mar 20 20:55 systemd-sysv-generator
-rwxr-xr-x 1 root root 14624 Mar 20 20:55 systemd-veritysetup-generator
```

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

```
root@debian11:~# systemctl --help
systemctl [OPTIONS...] COMMAND ...
```

Query or send control commands to the system manager.

Unit Commands:

<code>list-units [PATTERN...]</code>	List units currently in memory
<code>list-sockets [PATTERN...]</code>	List socket units currently in memory, ordered by address
<code>list-timers [PATTERN...]</code>	List timer units currently in memory, ordered by next elapse
<code>is-active PATTERN...</code>	Check whether units are active
<code>is-failed PATTERN...</code>	Check whether units are failed
<code>status [PATTERN... PID...]</code>	Show runtime status of one or more units
<code>show [PATTERN... JOB...]</code>	Show properties of one or more units/jobs or the manager
<code>cat PATTERN...</code>	Show files and drop-ins of specified units
<code>help PATTERN... PID...</code>	Show manual for one or more units
<code>list-dependencies [UNIT...]</code>	Recursively show units which are required or wanted by the units or by which those units are required or wanted
<code>start UNIT...</code>	Start (activate) one or more units
<code>stop UNIT...</code>	Stop (deactivate) one or more units
<code>reload UNIT...</code>	Reload one or more units
<code>restart UNIT...</code>	Start or restart one or more units
<code>try-restart UNIT...</code>	Restart one or more units if active
<code>reload-or-restart UNIT...</code>	Reload one or more units if possible, otherwise start or restart
<code>try-reload-or-restart UNIT...</code>	If active, reload one or more units, if supported, otherwise restart
<code>isolate UNIT</code>	Start one unit and stop all others
<code>kill UNIT...</code>	Send signal to processes of a unit
<code>clean UNIT...</code>	Clean runtime, cache, state, logs or configuration of unit
<code>freeze PATTERN...</code>	Freeze execution of unit processes
<code>thaw PATTERN...</code>	Resume execution of a frozen unit
<code>set-property UNIT PROPERTY=VALUE...</code>	Sets one or more properties of a unit
<code>service-log-level SERVICE [LEVEL]</code>	Get/set logging threshold for service
<code>service-log-target SERVICE [TARGET]</code>	Get/set logging target for service
<code>reset-failed [PATTERN...]</code>	Reset failed state for all, one, or more

```
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
  is-enabled UNIT...                 Check whether unit files are enabled
  mask UNIT...                       Mask one or more units
  unmask UNIT...                     Unmask one or more units
  link PATH...                       Link one or more units files into
                                     the search path
  revert UNIT...                     Revert one or more unit files to vendor
                                     version
  add-wants TARGET UNIT...           Add 'Wants' dependency for the target
                                     on specified one or more units

lines 1-58
[q]
```

LAB #3 - Fichiers de Configuration

3.1 - 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 **/lib/systemd/system** :

```
root@debian11:~# pkg-config systemd --variable=systemdsystemunitdir
/lib/systemd/system
```

```
root@debian11:~# ls -l /lib/systemd/system | more
total 1076
-rw-r--r-- 1 root root 576 Dec 7 2020 alsa-restore.service
-rw-r--r-- 1 root root 528 Dec 7 2020 alsa-state.service
lrwxrwxrwx 1 root root 9 Dec 7 2020 alsa-utils.service -> /dev/null
-rw-r--r-- 1 root root 776 Feb 6 2021 anacron.service
-rw-r--r-- 1 root root 154 Feb 6 2021 anacron.timer
-rw-r--r-- 1 root root 1162 Apr 3 2021 apparmor.service
-rw-r--r-- 1 root root 326 Jun 10 2021 apt-daily.service
-rw-r--r-- 1 root root 156 Jun 10 2021 apt-daily.timer
-rw-r--r-- 1 root root 389 Jun 10 2021 apt-daily-upgrade.service
-rw-r--r-- 1 root root 184 Jun 10 2021 apt-daily-upgrade.timer
-rw-r--r-- 1 root root 1491 Jan 6 2021 auditd.service
lrwxrwxrwx 1 root root 14 Mar 20 20:55 autovt@.service -> getty@.service
-rw-r--r-- 1 root root 1044 Feb 6 2021 avahi-daemon.service
-rw-r--r-- 1 root root 870 Feb 6 2021 avahi-daemon.socket
-rw-r--r-- 1 root root 927 Feb 2 2021 basic.target
-rw-r--r-- 1 root root 380 Feb 22 2021 blk-availability.service
-rw-r--r-- 1 root root 449 Feb 2 2021 blockdev@.target
-rw-r--r-- 1 root root 427 Feb 2 2021 bluetooth.target
-rw-r--r-- 1 root root 463 Feb 2 2021 boot-complete.target
-rw-r--r-- 1 root root 295 Nov 12 2020 colord.service
-rw-r--r-- 1 root root 150 Jan 13 2021 configure-printer@.service
-rw-r--r-- 1 root root 1073 Mar 20 20:55 console-getty.service
-rw-r--r-- 1 root root 312 Oct 29 2018 console-setup.service
-rw-r--r-- 1 root root 1254 Mar 20 20:55 container-getty@.service
-rw-r--r-- 1 root root 316 Feb 22 2021 cron.service
lrwxrwxrwx 1 root root 9 Mar 20 20:55 cryptdisks-early.service -> /dev/null
lrwxrwxrwx 1 root root 9 Mar 20 20:55 cryptdisks.service -> /dev/null
-rw-r--r-- 1 root root 473 Feb 2 2021 cryptsetup-pre.target
-rw-r--r-- 1 root root 420 Feb 2 2021 cryptsetup.target
lrwxrwxrwx 1 root root 13 Mar 20 20:55 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 May 27 2021 cups.path
```

```
-rw-r--r-- 1 root root 278 May 27 2021 cups.service
-rw-r--r-- 1 root root 132 May 27 2021 cups.socket
lrwxrwxrwx 1 root root 25 Mar 20 20:55 dbus-org.freedesktop.hostname1.service -> systemd-hostnamed.service
lrwxrwxrwx 1 root root 23 Mar 20 20:55 dbus-org.freedesktop.locale1.service -> systemd-localed.service
lrwxrwxrwx 1 root root 22 Mar 20 20:55 dbus-org.freedesktop.login1.service -> systemd-logind.service
lrwxrwxrwx 1 root root 25 Mar 20 20:55 dbus-org.freedesktop.timedate1.service -> systemd-timedated.service
-rw-r--r-- 1 root root 380 Feb 21 2021 dbus.service
-rw-r--r-- 1 root root 102 Feb 21 2021 dbus.socket
-rw-r--r-- 1 root root 1065 Mar 20 20:55 debug-shell.service
lrwxrwxrwx 1 root root 16 Mar 20 20:55 default.target -> graphical.target
-rw-r--r-- 1 root root 758 Feb 2 2021 dev-hugepages.mount
-rw-r--r-- 1 root root 701 Feb 2 2021 dev-mqueue.mount
-rw-r--r-- 1 root root 341 Feb 22 2021 dm-event.service
-rw-r--r-- 1 root root 248 Feb 22 2021 dm-event.socket
-rw-r--r-- 1 root root 297 Jun 7 2021 e2scrub_all.service
-rw-r--r-- 1 root root 251 Jun 7 2021 e2scrub_all.timer
-rw-r--r-- 1 root root 245 Jun 7 2021 e2scrub_fail@.service
-rw-r--r-- 1 root root 550 Jun 7 2021 e2scrub_reap.service
-rw-r--r-- 1 root root 438 Jun 7 2021 e2scrub@.service
-rw-r--r-- 1 root root 805 Mar 20 20:55 emergency.service
-rw-r--r-- 1 root root 479 Feb 2 2021 emergency.target
-rw-r--r-- 1 root root 549 Feb 2 2021 exit.target
-rw-r--r-- 1 root root 488 Feb 2 2021 final.target
-rw-r--r-- 1 root root 461 Feb 2 2021 first-boot-complete.target
-rw-r--r-- 1 root root 477 Jan 20 21:10 fstrim.service
--More--
[q]
```

3.2 - Surchargement des Fichiers par Défaut

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@debian11:~# 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@debian11:~# 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 **/lib/systemd/system**.

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

```
root@debian11:~# pkg-config systemd --variable=systemdsystemunitpath  
/etc/systemd/system:/etc/systemd/system:/run/systemd/system:/usr/local/lib/systemd/system:/lib/systemd/system:/usr/lib/systemd/system:/lib/systemd/system
```

3.3 - Les Fichiers d'Unités

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

```
root@debian11:~# cat /lib/systemd/system/ssh.service  
[Unit]  
Description=OpenBSD Secure Shell server  
Documentation=man:sshd(8) man:sshd_config(5)  
After=network.target auditd.service  
ConditionPathExists=!/etc/ssh/ssh_not_to_be_run
```

```
[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

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

- **Description=OpenBSD Secure Shell server,**
 - 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 auditd.service,**
 - 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,

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 \$SSHD_OPTS,**
 - Cette directive définit l'exécutable à lancer,
- **ExecReload=/usr/sbin/sshd -t,**

- 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.

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.

```
root@debian11:~# ls -l /etc/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx 1 root root 35 Apr 25 06:47 anacron.service -> /lib/systemd/system/anacron.service
lrwxrwxrwx 1 root root 34 Apr 29 14:24 auditd.service -> /lib/systemd/system/auditd.service
lrwxrwxrwx 1 root root 40 Apr 25 06:51 avahi-daemon.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root 41 Apr 25 06:32 console-setup.service -> /lib/systemd/system/console-setup.service
lrwxrwxrwx 1 root root 32 Apr 25 06:31 cron.service -> /lib/systemd/system/cron.service
lrwxrwxrwx 1 root root 40 Apr 25 06:52 cups-browsed.service -> /lib/systemd/system/cups-browsed.service
lrwxrwxrwx 1 root root 29 Apr 25 06:52 cups.path -> /lib/systemd/system/cups.path
lrwxrwxrwx 1 root root 38 Apr 25 06:51 lm-sensors.service -> /lib/systemd/system/lm-sensors.service
lrwxrwxrwx 1 root root 40 Apr 25 06:52 ModemManager.service -> /lib/systemd/system/ModemManager.service
lrwxrwxrwx 1 root root 38 Apr 25 06:31 networking.service -> /lib/systemd/system/networking.service
lrwxrwxrwx 1 root root 42 Apr 25 06:52 NetworkManager.service -> /lib/systemd/system/NetworkManager.service
lrwxrwxrwx 1 root root 36 Apr 25 06:30 remote-fs.target -> /lib/systemd/system/remote-fs.target
lrwxrwxrwx 1 root root 33 Apr 28 13:36 rsync.service -> /lib/systemd/system/rsync.service
lrwxrwxrwx 1 root root 35 Apr 25 06:31 rsyslog.service -> /lib/systemd/system/rsyslog.service
lrwxrwxrwx 1 root root 31 Apr 25 07:04 ssh.service -> /lib/systemd/system/ssh.service
lrwxrwxrwx 1 root root 42 Apr 25 06:51 wpa_supplicant.service -> /lib/systemd/system/wpa_supplicant.service
```

Dernièrement, la ligne suivante de la section **[Install]** indique que le fichier ssh.service est référencé par un alias dénommé **sshd.service** :

- **Alias=sshd.service**

Les alias se trouvent dans la répertoire **/etc/systemd/system/** :

```
root@debian11:~# ls -l /etc/systemd/system/
total 36
lrwxrwxrwx 1 root root 42 Apr 25 06:51 dbus-fi.wl.wpa_supplicant1.service ->
/lib/systemd/system/wpa_supplicant.service
lrwxrwxrwx 1 root root 40 Apr 25 06:51 dbus-org.freedesktop.Avahi.service -> /lib/systemd/system/avahi-
daemon.service
lrwxrwxrwx 1 root root 40 Apr 25 06:52 dbus-org.freedesktop.ModemManager1.service ->
/lib/systemd/system/ModemManager.service
lrwxrwxrwx 1 root root 53 Apr 25 06:52 dbus-org.freedesktop.nm-dispatcher.service ->
/lib/systemd/system/NetworkManager-dispatcher.service
lrwxrwxrwx 1 root root 45 Apr 25 06:48 dbus-org.freedesktop.timesync1.service -> /lib/systemd/system/systemd-
timesyncd.service
drwxr-xr-x 2 root root 4096 Apr 25 06:30 default.target.wants
lrwxrwxrwx 1 root root 35 Apr 25 06:53 display-manager.service -> /lib/systemd/system/lightdm.service
drwxr-xr-x 2 root root 4096 Apr 25 06:30 getty.target.wants
drwxr-xr-x 2 root root 4096 Apr 25 06:50 graphical.target.wants
drwxr-xr-x 2 root root 4096 Apr 29 14:24 multi-user.target.wants
drwxr-xr-x 2 root root 4096 Apr 25 06:52 network-online.target.wants
drwxr-xr-x 2 root root 4096 Apr 25 06:52 printer.target.wants
drwxr-xr-x 2 root root 4096 Apr 26 13:47 sockets.target.wants
lrwxrwxrwx 1 root root 31 Apr 25 07:04 sshd.service -> /lib/systemd/system/ssh.service
drwxr-xr-x 2 root root 4096 Apr 26 13:47 sysinit.target.wants
lrwxrwxrwx 1 root root 35 Apr 25 06:31 syslog.service -> /lib/systemd/system/rsyslog.service
drwxr-xr-x 2 root root 4096 Apr 29 14:38 timers.target.wants
```

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@debian11:~# systemctl show sshd
Type=notify
Restart=on-failure
NotifyAccess=main
```

```
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
TimeoutAbortUsec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUsec=infinity
WatchdogUsec=0
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
RestartPreventExitStatus=255
MainPID=449
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
Result=success
ReloadResult=success
CleanResult=success
UID=[not set]
GID=[not set]
NRestarts=0
OOMPolicy=stop
ExecMainStartTimestamp=Tue 2022-04-26 13:08:21 CEST
ExecMainStartTimestampMonotonic=7283003
ExecMainExitTimestampMonotonic=0
ExecMainPID=449
ExecMainCode=0
ExecMainStatus=0
ExecStartPre={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; ignore_errors=no ; start_time=[n/a] ;
stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }
ExecStartPreEx={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; flags= ; start_time=[n/a] ; stop_time=[n/a] ;
```

```
pid=0 ; code=(null) ; status=0/0 }
ExecStart={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $SSHD_OPTS ; ignore_errors=no ; start_time=[n/a] ;
stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }
ExecStartEx={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $SSHD_OPTS ; flags= ; start_time=[n/a] ;
stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }
ExecReload={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; 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 }
ExecReloadEx={ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; flags= ; 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/ssh.service
MemoryCurrent=5832704
CPUUsageNSec=664103000
EffectiveCPUs=
EffectiveMemoryNodes=
TasksCurrent=1
IPIngressBytes=[no data]
IPIngressPackets=[no data]
IPEgressBytes=[no data]
IPEgressPackets=[no data]
IOReadBytes=18446744073709551615
IOReadOperations=18446744073709551615
IOWriteBytes=18446744073709551615
IOWriteOperations=18446744073709551615
Delegate=no
lines 1-58
[q]
```

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

```
root@debian11:~# systemctl list-dependencies sshd.service
```

```
sshd.service
```

```
● |-- .mount
● |-- system.slice
● |-- sysinit.target
●   |-- apparmor.service
●   |-- blk-availability.service
●   |-- dev-hugepages.mount
●   |-- dev-mqueue.mount
●   |-- keyboard-setup.service
●   |-- kmod-static-nodes.service
●   |-- lvm2-lvmpolld.socket
●   |-- lvm2-monitor.service
●   |-- plymouth-read-write.service
●   |-- plymouth-start.service
●   |-- proc-sys-fs-binfmt_misc.automount
●   |-- 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-system-token.service
●   |-- systemd-hwdb-update.service
●   |-- systemd-journal-flush.service
●   |-- systemd-journald.service
●   |-- systemd-machine-id-commit.service
●   |-- systemd-modules-load.service
●   |-- systemd-pstore.service
●   |-- systemd-random-seed.service
●   |-- systemd-sysctl.service
●   |-- systemd-sysusers.service
●   |-- systemd-timesyncd.service
●   |-- systemd-tmpfiles-setup-dev.service
```

- └systemd-tmpfiles-setup.service
- └systemd-udev-trigger.service
- └systemd-udev.service
- └systemd-update-utmp.service
- └cryptsetup.target
 - └systemd-cryptsetup@sdcl1.service
- └local-fs.target
 - └- .mount
 - └mnt-sdcl1.mount
 - └systemd-fsck-root.service
 - └systemd-remount-fs.service
- └swap.target
 - └dev-disk-by\x2duuid-1f9439f5\x2d4b19\x2d49b1\x2db292\x2d60c2c674cee9.swap

LAB #4 - La Commande systemd-analyze

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

```
root@debian11:~# systemd-analyze
Startup finished in 3.197s (kernel) + 4.577s (userspace) = 7.775s
graphical.target reached after 4.567s 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@debian11:~# systemd-analyze blame
1.781s dev-sdb1.device
1.645s man-db.service
1.304s apparmor.service
1.254s udisks2.service
1.212s systemd-random-seed.service
1.155s systemd-journal-flush.service
1.027s NetworkManager-wait-online.service
```

```
1.005s systemd-udevd.service
876ms ModemManager.service
645ms ssh.service
608ms NetworkManager.service
584ms polkit.service
542ms logrotate.service
469ms avahi-daemon.service
449ms wpa_supplicant.service
445ms systemd-logind.service
441ms lightdm.service
433ms plymouth-quit-wait.service
410ms user@113.service
372ms apt-daily.service
358ms apt-daily-upgrade.service
329ms packagekit.service
313ms keyboard-setup.service
299ms networking.service
288ms systemd-timesyncd.service
226ms systemd-tmpfiles-setup.service
197ms systemd-modules-load.service
196ms rsyslog.service
184ms systemd-udev-trigger.service
165ms e2scrub_reap.service
164ms systemd-journald.service
152ms user@1000.service
108ms hddtemp.service
101ms systemd-tmpfiles-setup-dev.service
94ms systemd-sysusers.service
93ms plymouth-start.service
91ms lvm2-pvscan@8:41.service
89ms systemd-update-utmp.service
86ms systemd-remount-fs.service
80ms lm-sensors.service
72ms lvm2-pvscan@8:39.service
```

```
68ms systemd-sysctl.service
67ms plymouth-read-write.service
65ms lvm2-monitor.service
63ms dev-hugepages.mount
63ms sys-kernel-debug.mount
62ms dev-mqueue.mount
60ms sys-kernel-tracing.mount
57ms lvm2-pvscan@8:38.service
41ms systemd-user-sessions.service
33ms modprobe@fuse.service
32ms sys-kernel-config.mount
32ms ifupdown-pre.service
27ms auditd.service
27ms console-setup.service
22ms rtkit-daemon.service
20ms modprobe@configfs.service
19ms sys-fs-fuse-connections.mount
lines 1-58
[q]
```

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@debian11:~# 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 +645ms
└─network.target @3.425s
   └─NetworkManager.service @2.816s +608ms
      └─dbus.service @2.812s
         └─basic.target @2.789s
            └─sockets.target @2.789s
               └─dbus.socket @2.789s
```

```
└─sysinit.target @2.786s
  └─systemd-timesyncd.service @2.497s +288ms
    └─systemd-tmpfiles-setup.service @2.268s +226ms
      └─systemd-journal-flush.service @1.112s +1.155s
        └─systemd-journald.service @946ms +164ms
          └─systemd-journald.socket @925ms
            └─system.slice @487ms
              └─.slice @487ms
```

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

```
root@debian11:~# 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	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...]	Print list of syscalls in seccomp filter
condition CONDITION...	Evaluate conditions and asserts
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
```

Options:

```
-h --help             Show this help
--version             Show package version
--no-pager            Do not pipe output into a pager
--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 earlier
                    than the latest in the branch
--man[=BOOL]         Do [not] check for existence of man pages
--generators[=BOOL] Do [not] run unit generators (requires privileges)
--iterations=N       Show the specified number of iterations
--base-time=TIMESTAMP Calculate calendar times relative to specified time
```

See the `systemd-analyze(1)` man page for details.

LAB #5 - Les Cibles

Chaque Cible est décrite par un fichier de configuration :

```
root@debian11:~# 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.

5.1 - Contrôler les dépendances d'une Cible

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

```
root@debian11:~# systemctl list-dependencies multi-user.target
multi-user.target
● └─anacron.service
```

- |—auditd.service
- |—avahi-daemon.service
- |—console-setup.service
- |—cron.service
- |—cups-browsed.service
- |—cups.path
- |—dbus.service
- |—hddtemp.service
- |—lm-sensors.service
- |—ModemManager.service
- |—networking.service
- |—NetworkManager.service
- |—plymouth-quit-wait.service
- |—plymouth-quit.service
- |—rsync.service
- |—rsyslog.service
- |—ssh.service
- |—systemd-ask-password-wall.path
- |—systemd-logind.service
- |—systemd-update-utmp-runlevel.service
- |—systemd-user-sessions.service
- |—wpa_supplicant.service
- |—basic.target
 - |— .mount
 - |—tmp.mount
 - |—paths.target
 - |—slices.target
 - |— .slice
 - |—system.slice
 - |—sockets.target
 - |—avahi-daemon.socket
 - |—cups.socket
 - |—dbus.socket
 - |—dm-event.socket

```
● | | | systemd-initctl.socket
● | | | systemd-journald-audit.socket
● | | | systemd-journald-dev-log.socket
● | | | systemd-journald.socket
● | | | systemd-udevd-control.socket
● | | | systemd-udevd-kernel.socket
● | | | sysinit.target
● | | | apparmor.service
● | | | blk-availability.service
● | | | dev-hugepages.mount
● | | | dev-mqueue.mount
● | | | keyboard-setup.service
● | | | kmod-static-nodes.service
● | | | lvm2-lvmpolld.socket
● | | | lvm2-monitor.service
● | | | plymouth-read-write.service
● | | | plymouth-start.service
● | | | proc-sys-fs-binfmt_misc.automount
● | | | sys-fs-fuse-connections.mount
● | | | sys-kernel-config.mount
● | | | sys-kernel-debug.mount
● | | | sys-kernel-tracing.mount
lines 1-58
[q]
```

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@debian11:~# 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@debian11:~# ls -l /etc/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx 1 root root 35 Apr 25 06:47 anacron.service -> /lib/systemd/system/anacron.service
lrwxrwxrwx 1 root root 34 Apr 29 14:24 auditd.service -> /lib/systemd/system/auditd.service
lrwxrwxrwx 1 root root 40 Apr 25 06:51 avahi-daemon.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root 41 Apr 25 06:32 console-setup.service -> /lib/systemd/system/console-setup.service
lrwxrwxrwx 1 root root 32 Apr 25 06:31 cron.service -> /lib/systemd/system/cron.service
lrwxrwxrwx 1 root root 40 Apr 25 06:52 cups-browsed.service -> /lib/systemd/system/cups-browsed.service
lrwxrwxrwx 1 root root 29 Apr 25 06:52 cups.path -> /lib/systemd/system/cups.path
lrwxrwxrwx 1 root root 38 Apr 25 06:51 lm-sensors.service -> /lib/systemd/system/lm-sensors.service
lrwxrwxrwx 1 root root 40 Apr 25 06:52 ModemManager.service -> /lib/systemd/system/ModemManager.service
lrwxrwxrwx 1 root root 38 Apr 25 06:31 networking.service -> /lib/systemd/system/networking.service
lrwxrwxrwx 1 root root 42 Apr 25 06:52 NetworkManager.service -> /lib/systemd/system/NetworkManager.service
lrwxrwxrwx 1 root root 36 Apr 25 06:30 remote-fs.target -> /lib/systemd/system/remote-fs.target
lrwxrwxrwx 1 root root 33 Apr 28 13:36 rsync.service -> /lib/systemd/system/rsync.service
lrwxrwxrwx 1 root root 35 Apr 25 06:31 rsyslog.service -> /lib/systemd/system/rsyslog.service
lrwxrwxrwx 1 root root 31 Apr 25 07:04 ssh.service -> /lib/systemd/system/ssh.service
lrwxrwxrwx 1 root root 42 Apr 25 06:51 wpa_supplicant.service -> /lib/systemd/system/wpa_supplicant.service

root@debian11:~# ls -l /usr/lib/systemd/system/multi-user.target.wants
total 0
lrwxrwxrwx 1 root root 15 Feb 21 2021 dbus.service -> ../dbus.service
lrwxrwxrwx 1 root root 15 Mar 20 20:55 getty.target -> ../getty.target
lrwxrwxrwx 1 root root 24 Mar 2 2021 plymouth-quit.service -> ../plymouth-quit.service
lrwxrwxrwx 1 root root 29 Mar 2 2021 plymouth-quit-wait.service -> ../plymouth-quit-wait.service
lrwxrwxrwx 1 root root 33 Mar 20 20:55 systemd-ask-password-wall.path -> ../systemd-ask-password-wall.path
lrwxrwxrwx 1 root root 25 Mar 20 20:55 systemd-logind.service -> ../systemd-logind.service
lrwxrwxrwx 1 root root 39 Mar 20 20:55 systemd-update-utmp-runlevel.service -> ../systemd-update-utmp-
```

```
runlevel.service  
lrwxrwxrwx 1 root root 32 Mar 20 20:55 systemd-user-sessions.service -> ../systemd-user-sessions.service
```

5.2 - 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@debian11:~# systemctl get-default
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@debian11:~# ls -l /lib/systemd/system/multi-user.target
-rw-r--r-- 1 root root 540 Feb  2 2021 /lib/systemd/system/multi-user.target
root@debian11:~# systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target → /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@debian11:~# systemctl isolate rescue
```

```
root@debian11:~# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
```

rescue.target		loaded	active	active	Rescue Mode
---------------	--	--------	--------	--------	-------------

```
root@debian11:~# runlevel
5 1

root@debian11:~# who -r
      run-level 1  2022-04-30 12:19                last=5

root@debian11:~# systemctl isolate multi-user

root@debian11:~# systemctl list-units --type target | egrep "eme|res|gra|mul" | head -1
multi-user.target                                loaded active active Multi-User System

root@debian11:~# runlevel
1 3

root@debian11:~# who -r
      run-level 3  2022-04-30 12:21                last=1
```

LAB #6 - Gestion des Services

6.1 - Gestion des Instances Uniques

Commencez par installer le paquet **httpd** :

```
root@debian11:~# apt install -y apache2
```

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

```
root@debian11:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 12:22:50 CEST; 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
```

```
Main PID: 56037 (apache2)
  Tasks: 55 (limit: 4663)
  Memory: 8.8M
  CPU: 38ms
  CGroup: /system.slice/apache2.service
          └─56037 /usr/sbin/apache2 -k start
          └─56039 /usr/sbin/apache2 -k start
          └─56040 /usr/sbin/apache2 -k start
```

```
Apr 30 12:22:50 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 12:22:50 debian11 systemd[1]: Started The Apache HTTP Server.
```

Dans le cas du service apache2 ci-dessus, on peut constater que le statut est **enabled**. 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 la commande **systemctl is-enabled** :

```
root@debian11:~# systemctl is-enabled apache2.service
enabled
```

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

```
root@debian11:~# systemctl disable apache2.service
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
```

```
root@debian11:~# systemctl is-enabled apache2.service
disabled
```

```
root@debian11:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
```

```
Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: enabled)
Active: active (running) since Sat 2022-04-30 12:22:50 CEST; 3min 6s ago
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 56037 (apache2)
Tasks: 55 (limit: 4663)
Memory: 8.8M
CPU: 46ms
CGroup: /system.slice/apache2.service
├─56037 /usr/sbin/apache2 -k start
├─56039 /usr/sbin/apache2 -k start
└─56040 /usr/sbin/apache2 -k start
```

Apr 30 12:22:50 debian11 systemd[1]: Starting The Apache HTTP Server...

Apr 30 12:22:50 debian11 systemd[1]: Started The Apache HTTP Server.

Dans le cas du service httpd ci-dessus, on peut constater que l'état est **active (running)**. 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@debian11:~# systemctl is-active apache2.service
active
```

Pour rendre l'état **inactive(dead)**, utilisez la commande suivante :

```
root@debian11:~# systemctl stop apache2.service
```

Vérifiez ensuite l'état du service :

```
root@debian11:~# systemctl is-active apache2.service
inactive

root@debian11:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: enabled)
   Active: inactive (dead) since Sat 2022-04-30 12:27:53 CEST; 1min 26s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 56435 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
    Main PID: 56037 (code=exited, status=0/SUCCESS)
      CPU: 77ms

Apr 30 12:22:50 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 12:22:50 debian11 systemd[1]: Started The Apache HTTP Server.
Apr 30 12:27:53 debian11 systemd[1]: Stopping The Apache HTTP Server...
Apr 30 12:27:53 debian11 systemd[1]: apache2.service: Succeeded.
Apr 30 12:27:53 debian11 systemd[1]: Stopped The Apache HTTP Server.
```

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

```
root@debian11:~# systemctl start apache2.service

root@debian11:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 12:30:17 CEST; 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 56448 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 56452 (apache2)
      Tasks: 55 (limit: 4663)
     Memory: 8.7M
        CPU: 35ms
```

```
CGroup: /system.slice/apache2.service
├─56452 /usr/sbin/apache2 -k start
├─56453 /usr/sbin/apache2 -k start
└─56454 /usr/sbin/apache2 -k start
```

Apr 30 12:30:17 debian11 systemd[1]: Starting The Apache HTTP Server...

Apr 30 12:30:17 debian11 systemd[1]: Started The Apache HTTP Server.

Pour activer un service au prochain démarrage du système, utilisez l'option **enable** :

```
root@debian11:~# systemctl enable apache2.service
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service →
/lib/systemd/system/apache2.service.
```

```
root@debian11:~# systemctl status apache2.service
```

● apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)

Active: active (running) since Sat 2022-04-30 12:30:17 CEST; 1min 5s ago

Docs: <https://httpd.apache.org/docs/2.4/>

Main PID: 56452 (apache2)

Tasks: 55 (limit: 4663)

Memory: 8.7M

CPU: 39ms

```
CGroup: /system.slice/apache2.service
├─56452 /usr/sbin/apache2 -k start
├─56453 /usr/sbin/apache2 -k start
└─56454 /usr/sbin/apache2 -k start
```

Apr 30 12:30:17 debian11 systemd[1]: Starting The Apache HTTP Server...

Apr 30 12:30:17 debian11 systemd[1]: Started The Apache HTTP Server.

6.2 - 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@debian11:~# cat /usr/lib/systemd/system/apache2@.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
ConditionPathIsDirectory=/etc/apache2-%i
Documentation=https://httpd.apache.org/docs/2.4/

[Service]
Type=forking
Environment=APACHE_CONFDIR=/etc/apache2-%i APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl graceful-stop
ExecReload=/usr/sbin/apachectl graceful
KillMode=mixed
PrivateTmp=true
Restart=on-abort

[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.
- %l : 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/apache2@.service** :

```
root@debian11:~# cp /usr/lib/systemd/system/apache2@.service /usr/lib/systemd/system/apache2@instance01.service
root@debian11:~# cp /usr/lib/systemd/system/apache2@.service /usr/lib/systemd/system/apache2@instance02.service
```

Copiez le répertoire **/etc/apache2** vers **/etc/apache2-instance01** et **/etc/apache2-instance02** :

```
root@debian11:~# cp -r /etc/apache2/ /etc/apache2-instance01
root@debian11:~# cp -r /etc/apache2/ /etc/apache2-instance02
```

Editez le fichier **vi /etc/apache2-instance01/ports.conf** en modifiant le port d'écoute à 81 :

```
root@debian11:~# vi /etc/apache2-instance01/ports.conf
root@debian11:~# cat /etc/apache2-instance01/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 81

<IfModule ssl_module>
    Listen 443
```

```
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Editez le fichier **vi /etc/apache2-instance02/ports.conf** en modifiant le port d'écoute à 82 :

```
root@debian11:~# vi /etc/apache2-instance02/ports.conf
root@debian11:~# cat /etc/apache2-instance02/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 82

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Editez la directive **APACHE_PID_FILE** du fichier **/etc/apache2-instance01/envvars** :

```
root@debian11:~# vi /etc/apache2-instance01/envvars
root@debian11:~# cat /etc/apache2-instance01/envvars
# envvars - default environment variables for apache2ctl
```

```
# this won't be correct after changing uid
unset HOME

# for supporting multiple apache2 instances
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
    SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
    SUFFIX=
fi

# Since there is no sane way to get the parsed apache2 config in scripts, some
# settings are defined via environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
# temporary state file location. This might be changed to /run in Wheezy+1
export APACHE_PID_FILE=/var/run/apache2-instance01/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX

## The locale used by some modules like mod_dav
export LANG=C
## Uncomment the following line to use the system default locale instead:
#. /etc/default/locale

export LANG

## The command to get the status for 'apache2ctl status'.
## Some packages providing 'www-browser' need '--dump' instead of '-dump'.
#export APACHE_LYNX='www-browser -dump'

## If you need a higher file descriptor limit, uncomment and adjust the
```

```
## following line (default is 8192):  
#APACHE_ULIMIT_MAX_FILES='ulimit -n 65536'  
  
## If you would like to pass arguments to the web server, add them below  
## to the APACHE_ARGUMENTS environment.  
#export APACHE_ARGUMENTS=''  
  
## Enable the debug mode for maintainer scripts.  
## This will produce a verbose output on package installations of web server modules and web application  
## installations which interact with Apache  
#export APACHE2_MAINTSCRIPT_DEBUG=1
```

Editez la directive **APACHE_PID_FILE** du fichier **/etc/apache2-instance02/envvars** :

```
root@debian11:~# vi /etc/apache2-instance02/envvars  
root@debian11:~# cat /etc/apache2-instance02/envvars  
# envvars - default environment variables for apache2ctl  
  
# this won't be correct after changing uid  
unset HOME  
  
# for supporting multiple apache2 instances  
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then  
    SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"  
else  
    SUFFIX=  
fi  
  
# Since there is no sane way to get the parsed apache2 config in scripts, some  
# settings are defined via environment variables and then used in apache2ctl,  
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.  
export APACHE_RUN_USER=www-data  
export APACHE_RUN_GROUP=www-data  
# temporary state file location. This might be changed to /run in Wheezy+1
```

```
export APACHE_PID_FILE=/var/run/apache2-instance02/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX

## The locale used by some modules like mod_dav
export LANG=C
## Uncomment the following line to use the system default locale instead:
#. /etc/default/locale

export LANG

## The command to get the status for 'apache2ctl status'.
## Some packages providing 'www-browser' need '--dump' instead of '-dump'.
#export APACHE_LYNX='www-browser -dump'

## If you need a higher file descriptor limit, uncomment and adjust the
## following line (default is 8192):
#APACHE_ULIMIT_MAX_FILES='ulimit -n 65536'

## If you would like to pass arguments to the web server, add them below
## to the APACHE_ARGUMENTS environment.
#export APACHE_ARGUMENTS=''

## Enable the debug mode for maintainer scripts.
## This will produce a verbose output on package installations of web server modules and web application
## installations which interact with Apache
#export APACHE2_MAINTSCRIPT_DEBUG=1
```

Créez les répertoires **/var/log/apache2-instance01** et **/var/log/apache2-instance01** et modifiant le groupe associé :

```
root@debian11:~# mkdir /var/log/apache2-instance01
root@debian11:~# mkdir /var/log/apache2-instance02
```

```
root@debian11:~# chown root:adm /var/log/apache2-instance01
root@debian11:~# chown root:adm /var/log/apache2-instance02
```

Démarrez les deux services :

```
root@debian11:~# systemctl start apache2@instance01.service

root@debian11:~# systemctl status apache2@instance01.service
● apache2@instance01.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2@instance01.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 13:51:47 CEST; 4s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 56906 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 56910 (apache2)
    Tasks: 55 (limit: 4663)
   Memory: 8.8M
      CPU: 31ms
   CGroup: /system.slice/system-apache2.slice/apache2@instance01.service
           └─56910 /usr/sbin/apache2 -d /etc/apache2-instance01 -k start
           └─56911 /usr/sbin/apache2 -d /etc/apache2-instance01 -k start
           └─56912 /usr/sbin/apache2 -d /etc/apache2-instance01 -k start

Apr 30 13:51:47 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 13:51:47 debian11 systemd[1]: Started The Apache HTTP Server.

root@debian11:~# systemctl start apache2@instance02.service

root@debian11:~# systemctl status apache2@instance02.service
● apache2@instance02.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2@instance02.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 14:04:21 CEST; 10s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 57137 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 57148 (apache2)
```

```
Tasks: 55 (limit: 4663)
Memory: 8.8M
CPU: 37ms
CGroup: /system.slice/system-apache2.slice/apache2@instance02.service
├─57148 /usr/sbin/apache2 -d /etc/apache2-instance02 -k start
├─57149 /usr/sbin/apache2 -d /etc/apache2-instance02 -k start
└─57150 /usr/sbin/apache2 -d /etc/apache2-instance02 -k start
```

Apr 30 14:04:21 debian11 systemd[1]: Starting The Apache HTTP Server...

Apr 30 14:04:21 debian11 systemd[1]: Started The Apache HTTP Server.

Installez l'exécutable **lynx** :

```
root@debian11:~# apt -y install lynx
```

Testez ensuite les deux instances d'Apache :

```
root@debian11:~# lynx --dump http://localhost:81 | more
```

Debian Logo Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is fully documented in `/usr/share/doc/apache2/README.Debian.gz`. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the `[1]manual` if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

- * `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- * `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- * Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- * They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and

a2enconf, a2disconf . See their respective man pages for detailed information.

- * The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default configuration.

--More--

[q]

```
root@debian11:~# lynx --dump http://localhost:82 | more
```

```
Debian Logo Apache2 Debian Default Page
It works!
```

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [1]manual if the apache2-doc package was

installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

/etc/apache2/

```
|-- apache2.conf
|    |-- ports.conf
|-- mods-enabled
|    |-- *.load
|    |-- *.conf
|-- conf-enabled
|    |-- *.conf
|-- sites-enabled
|    |-- *.conf
```

- * apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- * ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- * Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- * They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf . See their respective man pages for detailed information.
- * The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not work with the default

configuration.

--More--

[q]

6.3 - Interdire la Modification du Statut d'un Service

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

```
root@debian11:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 12:30:17 CEST; 1h 39min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 56452 (apache2)
    Tasks: 55 (limit: 4663)
   Memory: 8.7M
      CPU: 379ms
   CGroup: /system.slice/apache2.service
           └─56452 /usr/sbin/apache2 -k start
           └─56453 /usr/sbin/apache2 -k start
           └─56454 /usr/sbin/apache2 -k start
```

```
Apr 30 12:30:17 debian11 systemd[1]: Starting The Apache HTTP Server...
```

```
Apr 30 12:30:17 debian11 systemd[1]: Started The Apache HTTP Server.
```

```
root@debian11:~# systemctl mask apache2.service
Created symlink /etc/systemd/system/apache2.service → /dev/null.
```

```
root@debian11:~# systemctl list-unit-files --type=service | grep apache2
apache2.service             masked          enabled
apache2@.service            disabled       enabled
apache2@instance01.service  disabled       enabled
```

```
apache2@instance02.service          disabled          enabled
```

```
root@debian11:~# systemctl disable apache2.service
```

```
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
```

```
Executing: /lib/systemd/systemd-sysv-install disable apache2
```

```
Unit /etc/systemd/system/apache2.service is masked, ignoring.
```

```
root@debian11:~# systemctl status apache2.service
```

```
● apache2.service - The Apache HTTP Server
```

```
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
```

```
   Active: active (running) since Sat 2022-04-30 14:17:14 CEST; 3s ago
```

```
   Docs: https://httpd.apache.org/docs/2.4/
```

```
   Process: 57431 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
```

```
 Main PID: 57435 (apache2)
```

```
   Tasks: 55 (limit: 4663)
```

```
  Memory: 8.8M
```

```
    CPU: 33ms
```

```
 CGroup: /system.slice/apache2.service
```

```
        └─57435 /usr/sbin/apache2 -k start
```

```
        └─57436 /usr/sbin/apache2 -k start
```

```
        └─57437 /usr/sbin/apache2 -k start
```

```
Apr 30 14:17:14 debian11 systemd[1]: Starting The Apache HTTP Server...
```

```
Apr 30 14:17:14 debian11 systemd[1]: Started The Apache HTTP Server.
```

```
root@debian11:~# systemctl stop apache2.service
```

```
root@debian11:~# systemctl status apache2.service
```

```
● apache2.service
```

```
   Loaded: masked (Reason: Unit apache2.service is masked.)
```

```
   Active: inactive (dead) since Sat 2022-04-30 14:17:45 CEST; 3s ago
```

```
 Main PID: 57435 (code=exited, status=0/SUCCESS)
```

```
    CPU: 39ms
```

```
Apr 30 14:17:14 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 14:17:14 debian11 systemd[1]: Started The Apache HTTP Server.
Apr 30 14:17:45 debian11 systemd[1]: Stopping apache2.service...
Apr 30 14:17:45 debian11 systemd[1]: apache2.service: Succeeded.
Apr 30 14:17:45 debian11 systemd[1]: Stopped apache2.service.
```

```
root@debian11:~# systemctl start apache2.service
Failed to start apache2.service: Unit apache2.service is masked.
```

```
root@debian11:~# systemctl unmask apache2.service
Removed /etc/systemd/system/apache2.service.
```

```
root@debian11:~# systemctl status apache2.service
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sat 2022-04-30 14:17:45 CEST; 1min 13s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 57435 (code=exited, status=0/SUCCESS)
      CPU: 39ms
```

```
Apr 30 14:17:14 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 14:17:14 debian11 systemd[1]: Started The Apache HTTP Server.
Apr 30 14:17:45 debian11 systemd[1]: Stopping apache2.service...
Apr 30 14:17:45 debian11 systemd[1]: apache2.service: Succeeded.
Apr 30 14:17:45 debian11 systemd[1]: Stopped apache2.service.
```

```
root@debian11:~# systemctl enable apache2.service
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

```
root@debian11:~# systemctl status apache2.service
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sat 2022-04-30 14:17:45 CEST; 1min 20s ago
```

```
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 57435 (code=exited, status=0/SUCCESS)
CPU: 39ms
```

```
Apr 30 14:17:14 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 14:17:14 debian11 systemd[1]: Started The Apache HTTP Server.
Apr 30 14:17:45 debian11 systemd[1]: Stopping apache2.service...
Apr 30 14:17:45 debian11 systemd[1]: apache2.service: Succeeded.
Apr 30 14:17:45 debian11 systemd[1]: Stopped apache2.service.
```

```
root@debian11:~# systemctl start apache2.service
```

```
root@debian11:~# systemctl status apache2.service
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-04-30 14:19:12 CEST; 3s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 57685 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 57689 (apache2)
    Tasks: 55 (limit: 4663)
   Memory: 8.7M
      CPU: 30ms
   CGroup: /system.slice/apache2.service
           └─57689 /usr/sbin/apache2 -k start
             └─57690 /usr/sbin/apache2 -k start
               └─57691 /usr/sbin/apache2 -k start
```

```
Apr 30 14:19:11 debian11 systemd[1]: Starting The Apache HTTP Server...
Apr 30 14:19:12 debian11 systemd[1]: Started The Apache HTTP Server.
```

