

Dernière mise-à-jour : 2020/01/30 03:28

LSF115 - Gestion du Démarrage et de l'Arrêt du Système

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

Systemes 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 2

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

Le lancement de GRUB 2 se fait en trois étapes :

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

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

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

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

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

```
SLES12SP1:~ # cat /boot/grub2/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub2-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
set btrfs_relative_path="y"
```

```
export btrfs_relative_path
if [ -f ${config_directory}/grubenv ]; then
    load_env -f ${config_directory}/grubenv
elif [ -s $prefix/grubenv ]; then
    load_env
fi

if [ "${env_block}" ]; then
    load_env -f "${env_block}"
fi

if [ "${next_entry}" ]; then
    set default="${next_entry}"
    set next_entry=
    save_env next_entry
    if [ "${env_block}" ]; then
        save_env -f "${env_block}" next_entry
    fi
    set boot_once=true
else
    set default="${saved_entry}"
fi

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

export menuentry_id_option

if [ "${prev_saved_entry}" ]; then
    set saved_entry="${prev_saved_entry}"
    save_env saved_entry
```

```
    set prev_saved_entry=
    save_env prev_saved_entry
    set boot_once=true
fi

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

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

if [ x$feature_default_font_path = xy ] ; then
    font=unicode
else
    insmod part_msdos
    insmod btrfs
    set root='hd0,msdos2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
```

```
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 65337196-2d6b-4c8b-b917-30c3867bf265
else
  search --no-floppy --fs-uuid --set=root 65337196-2d6b-4c8b-b917-30c3867bf265
fi
  font="/usr/share/grub2/unicode.pf2"
fi

if loadfont $font ; then
  set gfxmode=auto
  load_video
  insmod gfxterm
  set locale_dir=$prefix/locale
  set lang=en_GB
  insmod gettext
fi
terminal_output gfxterm
insmod part_msdos
insmod btrfs
set root='hd0,msdos2'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 65337196-2d6b-4c8b-b917-30c3867bf265
else
  search --no-floppy --fs-uuid --set=root 65337196-2d6b-4c8b-b917-30c3867bf265
fi
insmod gfxmenu
loadfont ($root)/boot/grub2/themes/SLE/ascii.pf2
loadfont ($root)/boot/grub2/themes/SLE/DejaVuSans10.pf2
loadfont ($root)/boot/grub2/themes/SLE/DejaVuSans12.pf2
loadfont ($root)/boot/grub2/themes/SLE/DejaVuSans-Bold14.pf2
insmod png
set theme=($root)/boot/grub2/themes/SLE/theme.txt
export theme
if [ x${boot_once} = xtrue ]; then
```



```
    set timeout=0
elif [ x$feature_timeout_style = xy ] ; then
    set timeout_style=menu
    set timeout=8
# Fallback normal timeout code in case the timeout_style feature is
# unavailable.
else
    set timeout=8
fi
if [ -n "$extra_cmdline" ]; then
    submenu "Bootable snapshot #$snapshot_num" {
        menuentry "If OK, run 'snapper rollback' and reboot." { true; }
    }
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/10_linux ###
menuentry 'SLES 12-SP1' --class sles --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-
simple-65337196-2d6b-4c8b-b917-30c3867bf265' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod btrfs
    set root='hd0,msdos2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 65337196-2d6b-4c8b-b917-30c3867bf265
    else
        search --no-floppy --fs-uuid --set=root 65337196-2d6b-4c8b-b917-30c3867bf265
    fi
    echo    'Loading Linux 3.12.49-11-default ...'
    linux    /boot/vmlinuz-3.12.49-11-default root=UUID=65337196-2d6b-4c8b-b917-30c3867bf265  ${extra_cmdline}
    resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high
```

```
    echo    'Loading initial ramdisk ...'
    initrd  /boot/initrd-3.12.49-11-default
}
submenu 'Advanced options for SLES 12-SP1' --hotkey=1 $menuentry_id_option 'gnulinux-advanced-65337196-2d6b-4c8b-
b917-30c3867bf265' {
    menuentry 'SLES 12-SP1, with Linux 3.12.49-11-default' --hotkey=2 --class sles --class gnu-linux --class gnu
--class os $menuentry_id_option 'gnulinux-3.12.49-11-default-advanced-65337196-2d6b-4c8b-b917-30c3867bf265' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod btrfs
        set root='hd0,msdos2'
        if [ x$feature_platform_search_hint = xy ]; then
            search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 65337196-2d6b-4c8b-b917-30c3867bf265
        else
            search --no-floppy --fs-uuid --set=root 65337196-2d6b-4c8b-b917-30c3867bf265
        fi
        echo    'Loading Linux 3.12.49-11-default ...'
        linux  /boot/vmlinuz-3.12.49-11-default root=UUID=65337196-2d6b-4c8b-b917-30c3867bf265
        ${extra_cmdline} resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd-3.12.49-11-default
    }
}

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

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

### BEGIN /etc/grub.d/20_memtest86+ ###
### END /etc/grub.d/20_memtest86+ ###
```

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

### 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 ###

### BEGIN /etc/grub.d/80_suse_btrfs_snapshot ###
if [ -f "/.snapshots/grub-snapshot.cfg" ]; then
    source "/.snapshots/grub-snapshot.cfg"
fi
### END /etc/grub.d/80_suse_btrfs_snapshot ###

### BEGIN /etc/grub.d/90_persistent ###
### END /etc/grub.d/90_persistent ###
```

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 **grub2-mkconfig** sous SLES/OpenSUSE. La commande `grub2-mkconfig` prend en argument l'emplacement du fichier destination, par exemple :

- `grub2-mkconfig -o /boot/grub2/grub.cfg`, ou
- `grub2-mkconfig -o /boot/efi/EFI/<nom_distribution>/grub.cfg`

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

Le fichier `/boot/grub2/device.map`

```
SLES12SP1:~ # cat /boot/grub2/device.map
(hd0) /dev/sda
```

Le fichier `/etc/default/grub`

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

```
SLES12SP1:~ # cat /etc/default/grub
# Modified by YaST2. Last modification on Sun May  1 16:01:03 CEST 2016
# THIS FILE WILL BE PARTIALLY OVERWRITTEN by perl-Bootloader
# For the new kernel it try to figure out old parameters. In case we are not able to recognize it (e.g. change of
flavor or strange install order ) it it use as fallback installation parameters from /etc/sysconfig/bootloader

# If you change this file, run 'grub2-mkconfig -o /boot/grub2/grub.cfg' afterwards to update
# /boot/grub2/grub.cfg.
GRUB_DISTRIBUTOR=""
```

```
GRUB_DEFAULT=saved
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=8
GRUB_CMDLINE_LINUX_DEFAULT="resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high"
# kernel command line options for failsafe mode
GRUB_CMDLINE_LINUX_RECOVERY=single
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=gfxterm
# 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=auto
# 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"
# Skip 30_os-prober if you experienced very slow in probing them
# WARNING foregin OS menu entries will be lost if set true here
GRUB_DISABLE_OS_PROBER=true
# Set to 'y' for grub to be installed on an encrypted partition
GRUB_ENABLE_CRYPTODISK=n
SUSE_BTRFS_SNAPSHOT_BOOTING=true
GRUB_BACKGROUND=/boot/grub2/themes/SLE/background.png
GRUB_THEME=/boot/grub2/themes/SLE/theme.txt
GRUB_CMDLINE_XEN_DEFAULT="crashkernel=104M,high"
```



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

Dans ce fichier les directives sont :

| Directive | Description |
|-----------------------------|--|
| GRUB_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_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 |
| GRUB_BADRAM | Permet de spécifier de la mémoire défectueuse |

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 :

```
SLES12SP1:~ # ls -l /etc/grub.d
total 76
-rwxr-xr-x 1 root root 10367 Nov 17 2015 00_header
-rwxr-xr-x 1 root root 12145 Nov 17 2015 10_linux
-rwxr-xr-x 1 root root 13794 Nov 17 2015 20_linux_xen
-rwxr-xr-x 1 root root 1802 Nov 17 2015 20_memtest86+
-rwxr-xr-x 1 root root 11206 Nov 17 2015 30_os-prober
```

```
-rwxr-xr-x 1 root root 214 Nov 17 2015 40_custom
-rwxr-xr-x 1 root root 216 Nov 17 2015 41_custom
-rwxr-xr-x 1 root root 224 Nov 17 2015 80_suse_btrfs_snapshot
-rwxr-xr-x 1 root root 1259 Nov 17 2015 90_persistent
-rw-r--r-- 1 root root 483 Nov 17 2015 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.

Configurer l'Authentification

Pour configurer l'authentification sous GRUB 2, il faut créer le fichier **/etc/grub.d/01_users** :

```
SLES12SP1:~ # touch /etc/grub.d/01_users
SLES12SP1:~ # chmod 755 /etc/grub.d/01_users
```

Créez deux mots de passe hashés au format **PBKDF2** en utilisant la commande **grub2-mkpasswd-pbkdf2** :

```
SUSE12SP1:~ # grub2-mkpasswd-pbkdf2
Enter password: pass123
Reenter password: pass123
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.EBF0C57807165D30936A7242EAD66AC661F7785615B3D0F227523AA1960E1E8849BB04646A17A69ECD0A0C00
4ECBA0E83F0A36EE14EA5AD4B8849DD612DD65EF.FA7089F32055FB097E842C814888A43F1D5C83EA631D64B1F342B841D04376BFD266CD09
C455DB8C89E762422BD4EB65A36ACAEB039D9E5B96D524AFDD84F70B
```

Editez le fichier **/etc/grub.d/01_users** ainsi :

```
SUSE12SP1:~ # vi /etc/grub.d/01_users
```

```
SUSE12SP1:~ # cat /etc/grub.d/01_users
#!/bin/sh -e
cat <<EOF
set superusers="root"
password_pbkdf2 root
grub.pbkdf2.sha512.10000.EBF0C57807165D30936A7242EAD66AC661F7785615B3D0F227523AA1960E1E8849BB04646A17A69ECD0A0C00
4ECBA0E83F0A36EE14EA5AD4B8849DD612DD65EF.FA7089F32055FB097E842C814888A43F1D5C83EA631D64B1F342B841D04376BFD266CD09
C455DB8C89E762422BD4EB65A36ACAEB039D9E5B96D524AFDD84F70B
EOF
```

Notez qu'il est aussi possible d'utiliser des mots de passe non cryptés, par exemple :

```
SLES12SP1:~ # vi /etc/grub.d/01_users
SLES12SP1:~ # cat /etc/grub.d/01_users
#!/bin/sh -e
cat <<EOF
set superusers="root"
password root fenestros
EOF
```

Lancez la commande **grub2-mkconfig -o /boot/grub2/grub.cfg** :

```
SLES12SP1:~ # grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found theme: /boot/grub2/themes/SLE/theme.txt
Found linux image: /boot/vmlinuz-3.12.49-11-default
Found initrd image: /boot/initrd-3.12.49-11-default
done
```



A faire : Redémarrez votre VM. Constatez que GRUB 2 demande un nom d'utilisateur ainsi qu'un mot de passe. Attention, le clavier est en QWERTY (pass123 = pqss&é").

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 BRTFS.

ISOLINUX

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

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

PXELINUX

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

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

Isodhpx

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

Initramfs

Le fichier Initrd *INITial Ram Disk* est une archive au format cpio.

Examiner l'image existante

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

```
SUSE12SP1:/tmp # cd /tmp
SUSE12SP1:/tmp # cp /boot/initrd-3.12.74-60.64.40-default /tmp/
SUSE12SP1:/tmp # ls
.ICE-unix  .X11-unix  .font-unix
.Test-unix .XIM-unix  initrd-3.12.74-60.64.40-default
SUSE12SP1:/tmp # lsinitrd initrd-3.12.74-60.64.40-default | more
Image: initrd-3.12.74-60.64.40-default: 14M
=====
Version: dracut-037-98.2

Arguments: --logfile --force

dracut modules:
bash
warpclock
i18n
ifcfg
drm
plymouth
btrfs
```

```
kernel-modules
resume
rootfs-block
terminfo
udev-rules
haveged
systemd
usrmount
base
--More--
```

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

Le script **init**

Le script **init** est lancé lors du chargement de l'image. SLES 12 utilise le système de démarrage **systemd**. Pour cette raison le script **init** sous SLES 12 est un lien symbolique vers **/usr/lib/systemd/systemd** :

```
SUSE12SP1:/tmp # lsinitrd initrd-3.12.74-60.64.40-default | grep /usr/lib/systemd/systemd
lrwxrwxrwx  1 root      root           26 Oct 21 12:49 sbin/init -> ../usr/lib/systemd/systemd
```

Consultez le contenu d'un fichier dans **initrd**

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

```
SUSE12SP1:/tmp # lsinitrd initrd-3.12.74-60.64.40-default /etc/ld.so.conf
/usr/local/lib64
/usr/local/lib
include /etc/ld.so.conf.d/*.conf
# /lib64, /lib, /usr/lib64 and /usr/lib gets added
# automatically by ldconfig after parsing this file.
```



```
--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
--print-cmdline      Print the kernel command line for the given disk layout
--early-microcode    Combine early microcode with ramdisk
--no-early-microcode Do not combine early microcode with ramdisk
--kernel-cmdline [PARAMETERS] Specify default kernel command line parameters
--strip              Strip binaries in the initramfs
--nostrip            Do not strip binaries in the initramfs
--prelink             Prelink binaries in the initramfs
--noprelink          Do not prelink binaries in the initramfs
--hardlink           Hardlink files in the initramfs
--nohardlink         Do not hardlink files in the initramfs
--nowaitforswap      Do not wait for swap getting active on boot
--prefix [DIR]       Prefix initramfs files with [DIR]
--noprefix           Do not prefix initramfs files
--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
--fscks [LIST]       Add a space-separated list of fsck helpers.
--nofscks            Inhibit installation of any fsck helpers.
--ro-mnt             Mount / and /usr read-only by default.
-h, --help           This message
--debug             Output debug information of the build process
--profile            Output profile information of the build process
-L, --stdlog [0-6]   Specify logging level (to standard error)
                      0 - suppress any messages
```

```
1 - only fatal errors
2 - all errors
3 - warnings
4 - info
5 - debug info (here starts lots of output)
6 - trace info (and even more)
-v, --verbose      Increase verbosity level
-q, --quiet        Decrease verbosity level
-c, --conf [FILE] Specify configuration file to use.
                   Default: /etc/dracut.conf
--confdir [DIR]   Specify configuration directory to use *.conf files
                   from. Default: /etc/dracut.conf.d
--tmpdir [DIR]    Temporary directory to be used instead of default
                   /var/tmp.
-l, --local        Local mode. Use modules from the current working
                   directory instead of the system-wide installed in
                   /usr/lib/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.
-N, --no-hostonly Disables Host-Only mode
--hostonly-cmdline Store kernel command line arguments needed
                   in the initramfs
--no-hostonly-cmdline Do not store kernel command line arguments needed
                   in the initramfs
--persistent-policy [POLICY]
                   Use [POLICY] to address disks and partitions.
                   POLICY can be any directory name found in /dev/disk.
                   E.g. "by-uuid", "by-label"
--fstab            Use /etc/fstab to determine the root device.
--add-fstab [FILE] Add file to the initramfs fstab
--mount "[DEV] [MP] [FSTYPE] [FSOPTS]"
                   Mount device [DEV] on mountpoint [MP] with filesystem
                   [FSTYPE] and options [FSOPTS] in the initramfs
```

```
--add-device "[DEV]"  Bring up [DEV] in initramfs
-i, --include [SOURCE] [TARGET]
    Include the files in the SOURCE directory into the
    Target directory in the final initramfs.
    If SOURCE is a file, it will be installed to TARGET
    in the final initramfs.
-I, --install [LIST]  Install the space separated list of files into the
    initramfs.
--gzip                Compress the generated initramfs using gzip.
    This will be done by default, unless another
    compression option or --no-compress is passed.
--bzip2               Compress the generated initramfs using bzip2.
    Make sure your kernel has bzip2 decompression support
    compiled in, otherwise you will not be able to boot.
--lzma                Compress the generated initramfs using lzma.
    Make sure your kernel has lzma support compiled in,
    otherwise you will not be able to boot.
--xz                  Compress the generated initramfs using xz.
    Make sure that your kernel has xz support compiled
    in, otherwise you will not be able to boot.
--lzo                  Compress the generated initramfs using lzop.
    Make sure that your kernel has lzo support compiled
    in, otherwise you will not be able to boot.
--lz4                  Compress the generated initramfs using lz4.
    Make sure that your kernel has lz4 support compiled
    in, otherwise you will not be able to boot.
--compress [COMPRESSION] Compress the generated initramfs with the
    passed compression program. Make sure your kernel
    knows how to decompress the generated initramfs,
    otherwise you will not be able to boot.
--no-compress         Do not compress the generated initramfs. This will
    override any other compression options.
--list-modules         List all available dracut modules.
-M, --show-modules    Print included module's name to standard output during
```



```
        build.
--keep      Keep the temporary initramfs for debugging purposes
--printsize Print out the module install size
--sshkey [SSHKEY] Add ssh key to initramfs (use with ssh-client module)
--logfile [FILE] Logfile to use (overrides configuration setting)
--check-supported Check to ensure that modules are marked supported when
                  using a kernel that is configured to check the
                  support status of a module before loading.
```

If [LIST] has multiple arguments, then you have to put these in quotes.

For example:

```
# dracut --add-drivers "module1 module2" ...
    Add splash animation and bootscreen to initrd.
```

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

```
SUSE12SP1:/tmp # cat /etc/dracut.conf
# PUT YOUR CONFIG HERE OR IN separate files named *.conf
# in /etc/dracut.conf.d
# SEE man dracut.conf(5)

# Sample dracut config file

#logfile=/var/log/dracut.log
#fileloglvl=6

# Exact list of dracut modules to use.  Modules not listed here are not going
# to be included.  If you only want to add some optional modules use
# add_dracutmodules option instead.
#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="no"  
  
# install local /etc/lvm/lvm.conf  
#lvmconf="no"  
  
# A list of fsck tools to install. If it's not specified, module's hardcoded  
# default is used, currently: "umount mount /sbin/fsck* xfs_db xfs_check  
# xfs_repair e2fsck jfs_fsck reiserfsck btrfsck". The installation is  
# opportunistic, so non-existing tools are just ignored.  
#fscks=""  
  
# inhibit installation of any fsck tools  
#nofscks="yes"  
  
# mount / and /usr read-only by default  
#ro_mnt="no"  
  
# set the directory for temporary files
```

```
# default: /var/tmp
#tmpdir=/tmp
```

Editez ce fichier pour inclure les modules nécessaires afin d'utiliser un disque dur USB pour la racine de SLES :

```
SUSE12SP1:/tmp # vi /etc/dracut.conf
SUSE12SP1:/tmp # cat /etc/dracut.conf
# PUT YOUR CONFIG HERE OR IN separate files named *.conf
# in /etc/dracut.conf.d
# SEE man dracut.conf(5)

# Sample dracut config file

#logfile=/var/log/dracut.log
#fileloglvl=6

# Exact list of dracut modules to use.  Modules not listed here are not going
# to be included.  If you only want to add some optional modules use
# add_dracutmodules option instead.
#dracutmodules+="

# dracut modules to omit
#omit_dracutmodules+="

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

# additional kernel modules to the default
add_drivers+="ehci-hcd ohci-hcd usb-storage scsi_mod sd_mod"

# 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="no"

# install local /etc/lvm/lvm.conf
#lvmconf="no"

# A list of fsck tools to install. If it's not specified, module's hardcoded
# default is used, currently: "umount mount /sbin/fsck* xfs_db xfs_check
# xfs_repair e2fsck jfs_fsck reiserfsck btrfsck". The installation is
# opportunistic, so non-existing tools are just ignored.
#fscks=""

# inhibit installation of any fsck tools
#nofscks="yes"

# mount / and /usr read-only by default
#ro_mnt="no"

# set the directory for temporary files
# default: /var/tmp
#tmpdir=/tmp
```

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

```
SUSE12SP1:/tmp # dracut -v usbinitrd
Executing: /usr/bin/dracut -v usbinitrd
*** Including module: bash ***
*** Including module: warpclock ***
*** Including module: il8n ***
*** Including module: ifcfg ***
*** Including module: drm ***
```

```
*** Including module: plymouth ***
...
```

Notez la présence de votre nouvelle image **/tmp/usbinitrd** :

```
SUSE12SP1:/tmp # ls -l /tmp/usbinitrd
-rw----- 1 root root 13274056 Oct 24 15:51 /tmp/usbinitrd
```

Déplacez votre fichier usbinitrd au répertoire /boot :

```
SUSE12SP1:/tmp # mv usbinitrd /boot
```

Notez la présence d'un lien symbolique **initrd** dans le répertoire /boot :

```
SUSE12SP1:/tmp # ls -l /boot | grep initrd
lrwxrwxrwx 1 root root          31 Oct 21 10:37 initrd -> initrd-3.12.74-60.64.40-default
-rw----- 1 root root 14634092 Oct 21 12:49 initrd-3.12.74-60.64.40-default
-rw----- 1 root root 11001368 Oct 21 16:07 initrd-3.12.74-60.64.40-default-kdump
-rw----- 1 root root 13274056 Oct 24 15:51 usbinitrd
```

Ouvrez votre fichier **/boot/grub2/grub.cfg** et cherchez la section **10_linux**. Copiez la première section **menuentry** :

```
...
### BEGIN /etc/grub.d/10_linux ###
menuentry 'SLES 12-SP1' --class sles --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-
simple-97d0968c-e6bf-454e-b46d-9b32906d094a' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod btrfs
    set root='hd0,msdos2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
```

```
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 97d0968c-e6bf-454e-b46d-9b32906d094a
else
  search --no-floppy --fs-uuid --set=root 97d0968c-e6bf-454e-b46d-9b32906d094a
fi
echo 'Loading Linux 3.12.74-60.64.40-default ...'
linux /boot/vmlinuz-3.12.74-60.64.40-default root=UUID=97d0968c-e6bf-454e-b46d-9b32906d094a
${extra_cmdline} resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high
echo 'Loading initial ramdisk ...'
initrd /boot/initrd-3.12.74-60.64.40-default
}
...
```

Créez maintenant le fichier **/etc/grub.d/09_sles**, collez la section précédemment copiée à la place indiquée entre les lignes **cat** « **EOF** et **EOF** puis modifiez la dernière ligne **initrd** :

```
#!/bin/sh -e
cat << EOF
menuentry 'SLES 12-SP1' --class sles --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-
simple-97d0968c-e6bf-454e-b46d-9b32906d094a' {
  load_video
  set gfxpayload=keep
  insmod gzio
  insmod part_msdos
  insmod btrfs
  set root='hd0,msdos2'
  if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 97d0968c-e6bf-454e-b46d-9b32906d094a
  else
    search --no-floppy --fs-uuid --set=root 97d0968c-e6bf-454e-b46d-9b32906d094a
  fi
  echo 'Loading Linux 3.12.74-60.64.40-default ...'
  linux /boot/vmlinuz-3.12.74-60.64.40-default root=UUID=97d0968c-e6bf-454e-b46d-9b32906d094a
${extra_cmdline} resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high
```

```
    echo    'Loading initial ramdisk ...'  
    initrd  /boot/usbintrd  
}  
EOF
```

Rendez ce fichier exécutable :

```
SLES12SP1:/tmp # chmod +x /etc/grub.d/09_sles
```

Avant de continuer, supprimer le fichier **/etc/grub.d/01_users**.

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

```
SLES12SP1:/tmp # grub2-mkconfig -o /boot/grub2/grub.cfg  
Generating grub configuration file ...  
Found theme: /boot/grub2/themes/SLE/theme.txt  
Found linux image: /boot/vmlinuz-3.12.74-60.64.40-default  
Found initrd image: /boot/initrd-3.12.74-60.64.40-default  
done
```

Contrôlez le fichier `/boot/grub2/grub.cfg` :

```
...  
### BEGIN /etc/grub.d/09_sles ###  
menuentry 'SLES 12-SP1' --class sles --class gnu-linux --class gnu --class os 'gnulinux-simple-97d0968c-  
e6bf-454e-b46d-9b32906d094a' {  
    load_video  
    set gfxpayload=keep  
    insmod gzio  
    insmod part_msdos  
    insmod btrfs  
    set root='hd0,msdos2'  
    if [ x = xy ]; then  
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 --hint-efi=hd0,msdos2 --hint-
```

```
baremetal=ahci0,msdos2 --hint='hd0,msdos2' 97d0968c-e6bf-454e-b46d-9b32906d094a
else
  search --no-floppy --fs-uuid --set=root 97d0968c-e6bf-454e-b46d-9b32906d094a
fi
echo 'Loading Linux 3.12.74-60.64.40-default ...'
linux /boot/vmlinuz-3.12.74-60.64.40-default root=UUID=97d0968c-e6bf-454e-b46d-9b32906d094a
resume=/dev/sda1 splash=silent quiet showopts crashkernel=104M,high
echo 'Loading initial ramdisk ...'
initrd /boot/usbinitrd
}
### END /etc/grub.d/09_sles ###
...
```



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

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
- /etc/init
- /bin/init
- /bin/sh

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
-

Systemd

SLES 12, comme beaucoup d'autres distributions, ont adopté **Systemd**. Ce dernier prend une approche différente au démarrage de Linux. En effet, **SysVinit** et **Upstart** sont 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 Cible est en quelque sorte une **grande étape** dans le démarrage du système tandis qu'une Unité peut être :

- un automount - (.automount),
- une périphérique - *Device* - (.device),
- un montage d'un périphérique - *Mount* - (.mount),
- un chemin - *Path* - (.path)
- un socket - *Socket* - (.socket),
- un service - *Service* - (.service),
- une instantanée - *Snapshot* - (.snapshot),
- une cible - *Target* - (.target).



Important : Dans le contexte d'une Unité, le type **cible** 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.

La Commande systemctl

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

```
SLES12SP1:~ # systemctl list-units
```

| UNIT | LOAD | ACTIVE | SUB | DESCRIPTION |
|--|--------|--------|---------|-----------------------------------|
| proc-sys-fs-binfmt_misc.automount File System Automount | loaded | active | waiting | Arbitrary Executable File Formats |
| sys-devices-pci00...rget2:0:0-2:0:0:0-block-sr0.device | loaded | active | plugged | VBOX_CD-ROM |
| sys-devices-pci0000:00-0000:00:03.0-net-eth0.device Controller (PRO/1000 MT Deskto | loaded | active | plugged | 82540EM Gigabit Ethernet |
| sys-devices-pci0000:00-0000:00:05.0-sound-card0.device | loaded | active | plugged | 82801AA AC'97 Audio Controller |
| sys-devices-pci00...:0:0-0:0:0:0-block-sda-sda1.device | loaded | active | plugged | VBOX_HARDDISK 1 |
| sys-devices-pci00...:0:0-0:0:0:0-block-sda-sda2.device | loaded | active | plugged | VBOX_HARDDISK 2 |
| sys-devices-pci00...rget0:0:0-0:0:0:0-block-sda.device | loaded | active | plugged | VBOX_HARDDISK |
| sys-devices-platform-serial8250-tty-ttyS0.device /sys/devices/platform/serial8250/tty/ttyS0 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS1.device /sys/devices/platform/serial8250/tty/ttyS1 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS10.device /sys/devices/platform/serial8250/tty/ttyS10 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS11.device /sys/devices/platform/serial8250/tty/ttyS11 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS12.device /sys/devices/platform/serial8250/tty/ttyS12 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS13.device /sys/devices/platform/serial8250/tty/ttyS13 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS14.device /sys/devices/platform/serial8250/tty/ttyS14 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS15.device /sys/devices/platform/serial8250/tty/ttyS15 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS16.device /sys/devices/platform/serial8250/tty/ttyS16 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS17.device /sys/devices/platform/serial8250/tty/ttyS17 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS18.device /sys/devices/platform/serial8250/tty/ttyS18 | loaded | active | plugged | |
| sys-devices-platform-serial8250-tty-ttyS19.device /sys/devices/platform/serial8250/tty/ttyS19 | loaded | active | plugged | |

```
sys-devices-platform-serial8250-tty-ttyS2.device      loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS2
sys-devices-platform-serial8250-tty-ttyS20.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS20
sys-devices-platform-serial8250-tty-ttyS21.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS21
sys-devices-platform-serial8250-tty-ttyS22.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS22
sys-devices-platform-serial8250-tty-ttyS23.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS23
sys-devices-platform-serial8250-tty-ttyS24.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS24
sys-devices-platform-serial8250-tty-ttyS25.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS25
sys-devices-platform-serial8250-tty-ttyS26.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS26
sys-devices-platform-serial8250-tty-ttyS27.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS27
sys-devices-platform-serial8250-tty-ttyS28.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS28
sys-devices-platform-serial8250-tty-ttyS29.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS29
sys-devices-platform-serial8250-tty-ttyS3.device      loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS3
sys-devices-platform-serial8250-tty-ttyS30.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS30
sys-devices-platform-serial8250-tty-ttyS31.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS31
sys-devices-platform-serial8250-tty-ttyS4.device     loaded active plugged
/sys/devices/platform/serial8250/tty/ttyS4
lines 1-35
```

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

```
SLES12SP1:~ # systemctl list-units --all | grep inactive | more
proc-fs-nfsd.mount                                loaded    inactive dead
NFSD configurati
on filesystem
proc-sys-fs-binfmt_misc.mount                    loaded    inactive dead
Arbitrary Execut
able File Formats File System
sys-fs-fuse-connections.mount                    loaded    inactive dead
FUSE Control Fil
e System
sys-kernel-config.mount                           loaded    inactive dead
Configuration Fi
le System
var-lib-nfs-rpc_pipefs.mount                      loaded    inactive dead
RPC Pipe File Sy
stem
var-lib.mount                                     not-found inactive dead
var-lib.mount                                     not-found inactive dead
var-lock.mount                                    loaded    inactive dead
Lock Directory
var-run.mount                                     loaded    inactive dead
Runtime Directory
var.mount                                          not-found inactive dead
var.mount                                          not-found inactive dead
systemd-ask-password-console.path                 loaded    inactive dead
Dispatch Passwor
d Requests to Console Directory Watch
acpid.service                                     not-found inactive dead
acpid.service                                     not-found inactive dead
alsa-restore.service                              loaded    inactive dead
Restore Sound Ca
rd State
alsa-state.service                                loaded    inactive dead
Manage Sound Car
```

```

d State (restore and store)
alsa-store.service              loaded    inactive dead
Store Sound Card
  State
amavis.service                  not-found inactive dead
amavis.service
auditd.service                  loaded    inactive dead
Security Auditing Service
auth-rpcgss-module.service      loaded    inactive dead
Kernel Module supporting RPCSEC_GSS
btrfsmaintenance-refresh.service loaded    inactive dead
Update cron periods from /etc/sysconfig/btrfsmaintenance
cyrus.service                   not-found inactive dead
cyrus.service
display-manager.service         loaded    inactive dead
X Display Manager
dracut-shutdown.service         loaded    inactive dead
Restore /run/initrd filesystem
--More--

```

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

```

SLES12SP1:~ # systemctl list-unit-files | more
UNIT FILE                                STATE
proc-sys-fs-binfmt_misc.automount       static
org.freedesktop.hostname1.busname       static
org.freedesktop.locale1.busname         static
org.freedesktop.login1.busname          static
org.freedesktop.machine1.busname        static
org.freedesktop.timedate1.busname        static

```

```
dev-hugepages.mount          static
dev-mqueue.mount             static
proc-fs-nfsd.mount           static
proc-sys-fs-binfmt_misc.mount static
sys-fs-fuse-connections.mount static
sys-kernel-config.mount      static
sys-kernel-debug.mount       static
tmp.mount                     static
var-lib-nfs-rpc_pipefs.mount static
var-lock.mount               static
var-run.mount                 static
systemd-ask-password-console.path static
systemd-ask-password-plymouth.path static
systemd-ask-password-wall.path static
session-1.scope              static
accounts-daemon.service      disabled
after-local.service          static
alsa-restore.service         static
alsa-state.service           static
alsa-store.service           static
alsasound.service           static
atd.service                  disabled
auditd.service              disabled
auth-rpcgss-module.service   static
autofs.service               disabled
autovt@.service              disabled
autoyast-initscripts.service disabled
blk-availability.service     disabled
--More--
```

Options de la Commande **systemctl**

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

```
SLES12SP1:~ # systemctl --help
systemctl [OPTIONS...] {COMMAND} ...
```

Query or send control commands to the systemd manager.

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



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

Unit Commands:

```
list-units [PATTERN...]      List loaded units
list-sockets [PATTERN...]    List loaded sockets ordered by address
list-timers [PATTERN...]     List loaded timers ordered by next elapse
start NAME...                Start (activate) one or more units
stop NAME...                 Stop (deactivate) one or more units
reload NAME...               Reload one or more units
restart NAME...              Start or restart one or more units
try-restart NAME...          Restart one or more units if active
reload-or-restart NAME...    Reload one or more units if possible,
                             otherwise start or restart
reload-or-try-restart NAME... Reload one or more units if possible,
                             otherwise restart if active
isolate NAME                 Start one unit and stop all others
kill NAME...                 Send signal to processes of a unit
is-active PATTERN...         Check whether units are active
is-failed PATTERN...         Check whether units are failed
status [PATTERN...|PID...]   Show runtime status of one or more units
show [PATTERN...|JOB...]     Show properties of one or more
                             units/jobs or the manager
cat PATTERN...               Show files and drop-ins of one or more units
```

```
set-property NAME ASSIGNMENT... Sets one or more properties of a unit
help PATTERN...|PID... Show manual for one or more units
reset-failed [PATTERN...] Reset failed state for all, one, or more
units
list-dependencies [NAME] Recursively show units which are required
or wanted by this unit or by which this
unit is required or wanted
```

Unit File Commands:

```
list-unit-files [PATTERN...] List installed unit files
enable NAME... Enable one or more unit files
disable NAME... Disable one or more unit files
reenable NAME... Reenable one or more unit files
preset NAME... Enable/disable one or more unit files
based on preset configuration
is-enabled NAME... Check whether unit files are enabled

mask NAME... Mask one or more units
unmask NAME... Unmask one or more units
link PATH... Link one or more units files into
the search path
get-default Get the name of the default target
set-default NAME Set the default target
```

Job Commands:

```
list-jobs [PATTERN...] List jobs
cancel [JOB...] Cancel all, one, or more jobs
```

Snapshot Commands:

```
snapshot [NAME] Create a snapshot
delete NAME... Remove one or more snapshots
```

Environment Commands:

```
show-environment Dump environment
```

```
set-environment NAME=VALUE... Set one or more environment variables
unset-environment NAME... Unset one or more environment variables
import-environment NAME... Import all, one or more environment variables
```

Manager Lifecycle Commands:

```
daemon-reload Reload systemd manager configuration
daemon-reexec Reexecute systemd manager
```

System Commands:

```
default Enter system default mode
rescue Enter system rescue mode
emergency Enter system emergency mode
halt Shut down and halt the system
poweroff Shut down and power-off the system
reboot [ARG] Shut down and reboot the system
kexec Shut down and reboot the system with kexec
exit Request user instance exit
switch-root ROOT [INIT] Change to a different root file system
suspend Suspend the system
hibernate Hibernate the system
hybrid-sleep Hibernate and suspend the system
```

```
lines 88-122/122 (END)
```

Fichiers de Configuration

Les Cibles et les Unités sont configurées par des fichiers se trouvant dans le répertoire **/etc/systemd/system** :

```
SLES12SP1:~ # ls -l /etc/systemd/system
total 40
lrwxrwxrwx 1 root root 45 Oct 21 12:45 SuSEfirewall2_setup.service ->
/usr/lib/systemd/system/SuSEfirewall2.service
drwxr-xr-x 1 root root 24 Oct 21 10:08 apparmor.service.d
drwxr-xr-x 1 root root 34 Oct 21 10:50 bluetooth.target.wants
```

```
lrwxrwxrwx 1 root root 41 Oct 21 10:50 dbus-org.bluez.service -> /usr/lib/systemd/system/bluetooth.service
lrwxrwxrwx 1 root root 45 Oct 21 10:40 dbus-org.opensuse.Network.AUT04.service ->
/usr/lib/systemd/system/wickedd-auto4.service
lrwxrwxrwx 1 root root 45 Oct 21 10:40 dbus-org.opensuse.Network.DHCP4.service ->
/usr/lib/systemd/system/wickedd-dhcp4.service
lrwxrwxrwx 1 root root 45 Oct 21 10:40 dbus-org.opensuse.Network.DHCP6.service ->
/usr/lib/systemd/system/wickedd-dhcp6.service
lrwxrwxrwx 1 root root 45 Oct 21 10:40 dbus-org.opensuse.Network.Nanny.service ->
/usr/lib/systemd/system/wickedd-nanny.service
lrwxrwxrwx 1 root root 41 Oct 21 12:45 default.target -> /usr/lib/systemd/system/multi-user.target
drwxr-xr-x 1 root root 228 Oct 21 12:39 default.target.wants
drwxr-xr-x 1 root root 36 Oct 21 10:33 getty.target.wants
drwxr-xr-x 1 root root 46 Oct 21 10:53 graphical.target.wants
drwxr-xr-x 1 root root 538 Oct 21 12:50 multi-user.target.wants
drwxr-xr-x 1 root root 28 Oct 21 10:40 network-online.target.wants
lrwxrwxrwx 1 root root 38 Oct 21 10:40 network.service -> /usr/lib/systemd/system/wicked.service
drwxr-xr-x 1 root root 26 Oct 21 10:34 remote-fs.target.wants
drwxr-xr-x 1 root root 50 Oct 21 12:15 sockets.target.wants
drwxr-xr-x 1 root root 68 Oct 21 10:38 sysinit.target.wants
lrwxrwxrwx 1 root root 39 Oct 21 12:13 syslog.service -> /usr/lib/systemd/system/rsyslog.service
drwxr-xr-x 1 root root 60 Oct 21 10:33 system-update.target.wants
lrwxrwxrwx 1 root root 47 Oct 21 10:53 xdm.service -> /usr/lib/systemd/system/display-manager.service
```

ainsi que par des fichiers se trouvant dans le répertoire **/lib/systemd/system** et **/usr/lib/systemd/system** :

```
SLES12SP1:~ # ls -l /usr/lib/systemd/system | more
total 1240
-rw-r--r-- 1 root root 403 Apr 20 2017 -.slice
-rw-r--r-- 1 root root 423 Apr 20 2017 SuSEfirewall2.service
-rw-r--r-- 1 root root 267 Apr 20 2017 SuSEfirewall2_init.service
-rw-r--r-- 1 root root 783 Feb 26 2016 YaST2-Firstboot.service
-rw-r--r-- 1 root root 1739 Feb 26 2016 YaST2-Second-Stage.service
-rw-r--r-- 1 root root 417 Sep 8 2015 accounts-daemon.service
-rw-r--r-- 1 root root 518 Apr 20 2017 after-local.service
```

```
-rw-r--r-- 1 root root 471 Aug 21 2015 alsa-restore.service
-rw-r--r-- 1 root root 482 Aug 21 2015 alsa-state.service
-rw-r--r-- 1 root root 415 Aug 21 2015 alsa-store.service
lrwxrwxrwx 1 root root 20 Oct 21 10:28 alsasound.service -> alsa-restore.service
-rw-r--r-- 1 root root 160 Sep 12 2016 atd.service
-rw-r--r-- 1 root root 621 Sep 22 2014 auditd.service
-rw-r--r-- 1 root root 686 Apr 20 2017 auth-rpcgss-module.service
-rw-r--r-- 1 root root 464 Apr 21 2017 autofs.service
lrwxrwxrwx 1 root root 14 Oct 21 10:33 autovt@.service -> getty@.service
-rw-r--r-- 1 root root 433 Mar 21 2017 autoyast-initscripts.service
-rw-r--r-- 1 root root 524 Apr 20 2017 basic.target
drwxr-xr-x 1 root root 76 Apr 20 2017 basic.target.wants
-r--r--r-- 1 root root 360 Mar 9 2017 blk-availability.service
-rw-r--r-- 1 root root 338 Sep 23 2014 bluetooth.service
-rw-r--r-- 1 root root 379 Apr 20 2017 bluetooth.target
-rw-r--r-- 1 root root 355 Dec 14 2016 bmc-snmp-proxy.service
-rw-r--r-- 1 root root 386 Aug 21 2015 brltty.service
-rw-r--r-- 1 root root 231 Aug 1 2016 btrfsmaintenance-refresh.service
-rw-r--r-- 1 root root 358 Apr 20 2017 busnames.target
drwxr-xr-x 1 root root 318 Oct 21 10:33 busnames.target.wants
lrwxrwxrwx 1 root root 9 Oct 21 10:33 cgroup.service -> /dev/null
lrwxrwxrwx 1 root root 30 Oct 21 10:33 cleanup.service -> systemd-tmpfiles-setup.service
lrwxrwxrwx 1 root root 9 Oct 21 10:33 clock.service -> /dev/null
-rw-r--r-- 1 root root 154 Sep 23 2014 configure-printer@.service
-rw-r--r-- 1 root root 770 Apr 20 2017 console-getty.service
-rw-r--r-- 1 root root 769 Apr 20 2017 console-shell.service
-rw-r--r-- 1 root root 751 Apr 20 2017 container-getty@.service
--More--
```

Par exemple, sous SLES 12, le service **sshd** est configuré par le fichier **/usr/lib/systemd/system/sshd.service** :

```
SLES12SP1:~ # cat /usr/lib/systemd/system/sshd.service
[Unit]
Description=OpenSSH Daemon
```

```
After=network.target

[Service]
EnvironmentFile=-/etc/sysconfig/ssh
ExecStartPre=/usr/sbin/sshd-gen-keys-start
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=always

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

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

- **ExecStart=/usr/sbin/sshd -D \$SSHD_OPTS,**
 - Cette ligne définit l'exécutable à lancer,
- **After=network.target,**
 - Cette ligne indique les services qui devraient être démarrés avant le démarrage de sshd,
- **WantedBy=multi-user.target,**
 - Cette ligne indique la Cible dans laquelle le service doit être démarré,
- **Restart=always,**
 - Cette ligne indique quand le service doit être re-démarré.

Systeme de Démarrage

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** :

- runlevel0.target,
 - runlevel1.target,
 - runlevel2.target,
 - runlevel3.target,
-

- runlevel4.target,
- runlevel5.target,
- runlevel6.target.

Ceci étant il y a principalement deux Cibles finales :

- **multi-user.target** qui est l'équivalent du niveau d'exécution 3,
- **graphical.target** qui est l'équivalent du niveau d'exécution 5.

Chaque Cible est décrite par un fichier de configuration :

```
SLES12SP1:~ # cat /usr/lib/systemd/system/graphical.target
# 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
After=multi-user.target
Conflicts=rescue.target
Wants=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 auparavant,
- **After=multi-user.target**,
 - Cette ligne indique le **multi-user.target** doit d'abord être lancé,
- **Conflicts=rescue.target**,

- Cette ligne indique la Cible en conflit avec le **graphical.target**,
- **Wants=display-manager.service**,
- Cette ligne indique quel service doit être démarré.

Dernièrement la Cible par défaut peut être modifiée en éditant le lien symbolique **/etc/systemd/system/default.target** :

```
SLES12SP1:~ # ls -l /etc/systemd/system/default.target
lrwxrwxrwx 1 root root 41 Oct 21 12:45 /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target
```

La Commande **systemd-analyze**

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

```
SLES12SP1:~ # systemd-analyze
Startup finished in 2.150s (kernel) + 3.388s (initrd) + 24.601s (userspace) = 30.140s
```

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 :

```
SLES12SP1:~ # systemd-analyze blame
15.276s wicked.service
2.381s SuSEfirewall2.service
2.105s kdump.service
1.836s dev-sda2.device
1.796s SuSEfirewall2_init.service
1.739s postfix.service
1.433s apparmor.service
1.014s plymouth-read-write.service
856ms kdump-rebuild-initrd.service
517ms btrfsmaintenance-refresh.service
444ms rsyslog.service
327ms alsa-restore.service
```



```
320ms boot-grub2-i386\x2dpc.mount
318ms systemd-user-sessions.service
315ms nscd.service
290ms rc-local.service
278ms opt.mount
252ms var-log.mount
244ms systemd-udev-trigger.service
244ms \x2esnapshots.mount
235ms var-tmp.mount
232ms boot-grub2-x86_64\x2defi.mount
224ms home.mount
220ms var-spool.mount
219ms tmp.mount
214ms var-opt.mount
212ms srv.mount
197ms var-lib-pgsql.mount
192ms usr-local.mount
186ms var-lib-mysql.mount
186ms var-lib-libvirt-images.mount
176ms var-crash.mount
170ms var-lib-mailman.mount
169ms sys-kernel-debug.mount
166ms var-lib-named.mount
164ms systemd-udev-root-symlink.service
159ms systemd-modules-load.service
159ms systemd-journald.service
147ms sshd.service
143ms plymouth-quit-wait.service
128ms plymouth-quit.service
125ms var-lib-mariadb.mount
124ms systemd-vconsole-setup.service
118ms dev-disk-by\x2duuid-b397a430\x2d7344\x2d4a60\x2d8c3b\x2d59c1c83c4740.swap
117ms dev-mqueue.mount
116ms dev-hugepages.mount
```

```
104ms wickedd-dhcp4.service
103ms wickedd-dhcp6.service
100ms plymouth-start.service
 99ms systemd-remount-fs.service
 96ms systemd-logind.service
 90ms wickedd-auto4.service
 83ms kmod-static-nodes.service
 71ms systemd-tmpfiles-setup.service
 70ms systemd-update-utmp.service
 64ms user@1000.service
 57ms systemd-tmpfiles-setup-dev.service
 51ms systemd-sysctl.service
 42ms iscsi.service
 42ms wickedd-nanny.service
 39ms systemd-udev.service
 39ms systemd-journal-flush.service
 38ms systemd-random-seed.service
 37ms wickedd.service
 17ms systemd-update-utmp-runlevel.service
```

lines 31-65/65 (END)

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

```
SLES12SP1:~ # systemd-analyze critical-chain sshd.service
The time after the unit is active or started is printed after the "@" character.
The time the unit takes to start is printed after the "+" character.

sshd.service +147ms
└─network.target @22.179s
   └─wicked.service @6.902s +15.276s
      └─wickedd-nanny.service @6.853s +42ms
         └─wickedd.service @6.813s +37ms
            └─wickedd-dhcp4.service @6.703s +104ms
               └─SuSEfirewall2_init.service @4.899s +1.796s
```

```
└─basic.target @4.576s
  └─timers.target @4.575s
    └─systemd-tmpfiles-clean.timer @4.574s
      └─sysinit.target @4.567s
        └─apparmor.service @3.132s +1.433s
          └─systemd-tmpfiles-setup.service @3.058s +71ms
            └─local-fs.target @3.001s
              └─boot-grub2-i386\x2dpc.mount @2.566s +320ms
                └─dev-disk-by\x2duuid-97d0968c\x2de6bf\x2d454e\x2db46d\x2d9b32906d094a.device @2.367s
```

Options de la Commande

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

```
SLES12SP1:~ # systemd-analyze --help
systemd-analyze [OPTIONS...] {COMMAND} ...
```

Process systemd profiling information.

| | | |
|----|--|--|
| -h | --help | Show this help |
| | --version | Show package version |
| | --no-pager | Do not pipe output into a pager |
| | --system | Connect to system manager |
| | --user | Connect to user manager |
| -H | --host=[USER@]HOST | Operate on remote host |
| -M | --machine=CONTAINER | Operate on local container |
| | --order | When generating a dependency graph, show only order |
| | --require | When generating a dependency graph, show only requirement |
| | --from-pattern=GLOB, --to-pattern=GLOB | When generating a dependency graph, filter only origins or destinations, respectively |
| | --fuzz=TIMESPAN | When printing the tree of the critical chain, print also services, which finished TIMESPAN earlier, than the |

```
latest in the branch. The unit of TIMESPAN is seconds
unless specified with a different unit, i.e. 50ms
```

Commands:

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

Gestion des Services

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

```
SLES12SP1:~ # systemctl status sshd.service
sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Tue 2017-10-24 18:15:51 CEST; 14min ago
   Process: 1330 ExecStartPre=/usr/sbin/sshd-gen-keys-start (code=exited, status=0/SUCCESS)
  Main PID: 1346 (sshd)
   CGroup: /system.slice/sshd.service
           └─1346 /usr/sbin/sshd -D

Oct 24 18:15:51 15 sshd-gen-keys-start[1330]: Checking for missing server keys in /etc/ssh
Oct 24 18:15:51 15 systemd[1]: Started OpenSSH Daemon.
Oct 24 18:15:52 15 sshd[1346]: Server listening on 0.0.0.0 port 22.
Oct 24 18:15:52 15 sshd[1346]: Server listening on :: port 22.
Oct 24 18:16:07 15 sshd[1827]: Accepted keyboard-interactive/pam for trainee from 10.0.2.2 port 36696 ssh2
Oct 24 18:16:07 15 sshd[1827]: pam_unix(sshd:session): session opened for user trainee by (uid=0)
```

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

```
SLES12SP1:~ # systemctl stop sshd.service
SLES12SP1:~ # systemctl status sshd.service
sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: inactive (dead) since Tue 2017-10-24 18:30:48 CEST; 7s ago
   Process: 1346 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, status=0/SUCCESS)
   Process: 1330 ExecStartPre=/usr/sbin/sshd-gen-keys-start (code=exited, status=0/SUCCESS)
   Main PID: 1346 (code=exited, status=0/SUCCESS)

Oct 24 18:15:51 15 sshd-gen-keys-start[1330]: Checking for missing server keys in /etc/ssh
Oct 24 18:15:51 15 systemd[1]: Started OpenSSH Daemon.
Oct 24 18:15:52 15 sshd[1346]: Server listening on 0.0.0.0 port 22.
Oct 24 18:15:52 15 sshd[1346]: Server listening on :: port 22.
Oct 24 18:16:07 15 sshd[1827]: Accepted keyboard-interactive/pam for trainee from 10.0.2.2 port 36696 ssh2
Oct 24 18:16:07 15 sshd[1827]: pam_unix(sshd:session): session opened for user trainee by (uid=0)
Oct 24 18:30:48 SLES12SP1.fenestros.loc systemd[1]: Stopping OpenSSH Daemon...
Oct 24 18:30:48 SLES12SP1.fenestros.loc systemd[1]: Stopped OpenSSH Daemon.
```

Pour démarrer un service, utilisez la commande suivante :

```
SLES12SP1:~ # systemctl start sshd.service
SLES12SP1:~ # systemctl status sshd.service
sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Tue 2017-10-24 18:33:41 CEST; 3s ago
   Process: 2197 ExecStartPre=/usr/sbin/sshd-gen-keys-start (code=exited, status=0/SUCCESS)
   Main PID: 2200 (sshd)
   CGroup: /system.slice/sshd.service
           └─2200 /usr/sbin/sshd -D

Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Starting OpenSSH Daemon...
Oct 24 18:33:41 SLES12SP1.fenestros.loc sshd-gen-keys-start[2197]: Checking for missing server keys in /etc/ssh
Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Started OpenSSH Daemon.
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on 0.0.0.0 port 22.
```

```
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on :: port 22.
```

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

```
SLES12SP1:~ # systemctl disable sshd.service
rm '/etc/systemd/system/multi-user.target.wants/sshd.service'
SLES12SP1:~ # systemctl status sshd.service
sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled)
   Active: active (running) since Tue 2017-10-24 18:33:41 CEST; 39s ago
 Main PID: 2200 (sshd)
    CGroup: /system.slice/sshd.service
           └─2200 /usr/sbin/sshd -D

Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Starting OpenSSH Daemon...
Oct 24 18:33:41 SLES12SP1.fenestros.loc sshd-gen-keys-start[2197]: Checking for missing server keys in /etc/ssh
Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Started OpenSSH Daemon.
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on 0.0.0.0 port 22.
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on :: port 22.
```

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

```
SLES12SP1:~ # systemctl enable sshd.service
ln -s '/usr/lib/systemd/system/sshd.service' '/etc/systemd/system/multi-user.target.wants/sshd.service'
SLES12SP1:~ # systemctl status sshd.service
sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Tue 2017-10-24 18:33:41 CEST; 1min 6s ago
 Main PID: 2200 (sshd)
    CGroup: /system.slice/sshd.service
           └─2200 /usr/sbin/sshd -D

Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Starting OpenSSH Daemon...
Oct 24 18:33:41 SLES12SP1.fenestros.loc sshd-gen-keys-start[2197]: Checking for missing server keys in /etc/ssh
```

```
Oct 24 18:33:41 SLES12SP1.fenestros.loc systemd[1]: Started OpenSSH Daemon.  
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on 0.0.0.0 port 22.  
Oct 24 18:33:42 SLES12SP1.fenestros.loc sshd[2200]: Server listening on :: port 22.
```

Arrêt du Système

La commande shutdown

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

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

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

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

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # shutdown --help  
shutdown [OPTIONS...] [TIME] [WALL...]
```

Shut down the system.

```
--help      Show this help  
-H --halt   Halt the machine
```

```
-P --poweroff  Power-off the machine
-r --reboot    Reboot the machine
-h            Equivalent to --poweroff, overridden by --halt
-k            Don't halt/power-off/reboot, just send warnings
  --no-wall   Don't send wall message before halt/power-off/reboot
-c            Cancel a pending shutdown
```

Parmi les options les plus importantes, on note :

| Option | Description |
|--------|--|
| -h | Arrêter le système |
| -r | Re-démarrer le système |
| -c | Annuler l'opération shutdown en cours |
| -f | Re-démarrer rapidement sans vérifier les systèmes de fichiers |
| -F | Forcer la vérification des systèmes de fichiers lors du prochain démarrage |

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

| Valeur | Description |
|--------|---|
| hh:mm | L'heure à laquelle l'opération aura lieu |
| +m | Nombre de minutes avant que l'opération aura lieu |
| now | L'opération est immédiate |



Important : Si l'opération est programmée pour dans moins de 5 minutes, les connexions supplémentaires sont interdites, y comprises les tentatives de connexion de root. Notez aussi que l'utilisation de la commande **shutdown** peut être accordée à d'autres utilisateurs de root en utilisant le fichier **/etc/shutdown.allow**.

Dans votre VM, ouvrez deux terminaux. Dans le premier passez en tant que root. Planifiez ensuite un redémarrage de la VM dans 30 minutes :

```
SLES12SP1:~ # shutdown -r +30
Shutdown scheduled for Tue 2017-10-24 19:06:34 CEST, use 'shutdown -c' to cancel.
```


Dans le deuxième terminal est passez en tant que route. Annulez ensuite le shutdown :

```
SLES12SP1:~ # shutdown -c  
  
Broadcast message from root@SLES12SP1.fenestros.loc (Tue 2017-10-24 18:36:58 CEST):  
  
The system shutdown has been cancelled at Tue 2017-10-24 18:37:58 CEST!
```

La commande reboot

Cette commande redémarre le système. Quand le système fonctionne normalement, l'exécution de reboot appelle la commande **shutdown -r**.

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # reboot --help  
reboot [OPTIONS...] [ARG]  
  
Reboot the system.  
  
    --help      Show this help  
    --halt      Halt the machine  
-p --poweroff  Switch off the machine  
    --reboot    Reboot the machine  
-f --force     Force immediate halt/power-off/reboot  
-w --wtmp-only Don't halt/power-off/reboot, just write wtmp record  
-d --no-wtmp   Don't write wtmp record  
    --no-wall   Don't send wall message before halt/power-off/reboot
```

La commande halt

Cette commande arrête le système. Quand le système fonctionne normalement, l'exécution de halt appelle la commande **shutdown -h**.

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # halt --help
halt [OPTIONS...]

Halt the system.

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

La commande poweroff

Cette commande arrête le système et coupe l'alimentation électrique. Elle est l'équivalente de la commande **halt -p**. Quand le système fonctionne normalement, l'exécution de **poweroff** appelle la commande **shutdown -hP**.

Options de la commande

Les options de cette commande sont :

```
SLES12SP1:~ # poweroff --help
poweroff [OPTIONS...]
```

Power off the system.

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

<html>

Copyright © 2004-2017 I2TCH LIMITED

</html>
