

Dernière mise-à-jour : 2020/08/07 13:49

LDF210 - Gestion du Démarrage et de l'Arrêt du Système

Contenu du Module

- **LDF210 - Gestion du Démarrage et de l'Arrêt du Système**
 - Contenu du Module
 - Présentation
 - BIOS, EFI et OpenFirmware
 - Systèmes à base du BIOS
 - Charger de Démarrage
 - Systèmes à base de l'EFI
 - Autres Systèmes
 - Gestionnaire d'amorçage
 - LILO
 - La Commande LILO
 - Codes Erreur de LILO
 - GRUB 2 sous Debian 8
 - Le fichier /boot/grub2/device.map
 - Le fichier /etc/default/grub
 - Les fichiers du répertoire /etc/grub.d
 - Configurer l'Authentification
 - Modifier la Configuration de GRUB 2 en Ligne de Commande
 - Gestionnaires de Démarrages Alternatifs
 - Systemd-boot
 - U-boot
 - Le Projet Syslinux
 - SYSLINUX
 - EXTLINUX
 - ISOLINUX

- PXELINUX
 - Isodhpx
- Initramfs sous Debian 8
 - Examiner l'image existante
 - Le script init
 - La Commande mkinitramfs
- Processus de Démarrage du Noyau Linux
- Processus Init
- 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
 - Linux Standard Base
 - Codes Retour Standardisés
 - Scripts
 - Répertoires rcx.d
 - rc.local
 - La Commande chkconfig
 - La Commande update-rc.d sous Debian 6
- Démarrer Debian 8 avec systemd
 - La Commande systemctl
 - Fichiers de Configuration
 - Système de Démarrage
 - La Commande systemd-analyze
 - Gestion des Services
- Arrêt Système du Système
 - La Commande shutdown
 - La Commande reboot
 - La Commande halt
 - La Commande poweroff

Présentation

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

- Le **firmware** ou **micrologiciel** démarre en effectuant un test rapide du matériel, appelé un **Power-On Self Test** ou **POST**, puis recherche le **Charger de Démarrage** (*Bootloader*) à exécuter à partir d'un support bootable,
- Le Charger de Démarrage est exécuté et il détermine quel noyau Linux à charger,
- Le noyau se charge en mémoire et commence à exécuter en arrière plan les programmes nécessaires au fonctionnement du système.



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

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**.



Important : Il est à noter que la première partie du Charger de Démarrage peut également être installé dans un PBR.

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
-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 une 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 système UEFI.

GRUB 2 sous Debian 8

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@debian8:~# 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_uqa
        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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    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=-1
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 0fc7a234-259a-416b-b674-51eedfb1823b
else
    search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
fi
insmod png
if background_image /usr/share/images/desktop-base/lines-grub.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 $menuentry_id_option
'gnulinux-simple-0fc7a234-259a-416b-b674-51eedfb1823b' {
    load_video
    insmod gzio
    if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    echo    'Loading Linux 3.16.0-4-686-pae ...'
    linux    /boot/vmlinuz-3.16.0-4-686-pae root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro
    initrd=/install/gtk/initrd.gz quiet
    echo    'Loading initial ramdisk ...'
    initrd    /boot/initrd.img-3.16.0-4-686-pae
}
```

```
submenu 'Advanced options for Debian GNU/Linux' $menuentry_id_option 'gnulinux-advanced-0fc7a234-259a-416b-b674-51eedfb1823b' {
    menuentry 'Debian GNU/Linux, with Linux 3.16.0-4-686-pae' --class debian --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-3.16.0-4-686-pae-advanced-0fc7a234-259a-416b-b674-51eedfb1823b' {
        load_video
        insmod gzio
        if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
        else
            search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
        fi
        echo      'Loading Linux 3.16.0-4-686-pae ...'
        linux      /boot/vmlinuz-3.16.0-4-686-pae root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro
        initrd=/install/gtk/initrd.gz quiet
        echo      'Loading initial ramdisk ...'
        initrd      /boot/initrd.img-3.16.0-4-686-pae
    }
    menuentry 'Debian GNU/Linux, with Linux 3.16.0-4-686-pae (recovery mode)' --class debian --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-3.16.0-4-686-pae-recovery-0fc7a234-259a-416b-b674-51eedfb1823b' {
        load_video
        insmod gzio
        if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
        fi
    }
}
```

```
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    echo    'Loading Linux 3.16.0-4-686-pae ...'
    linux   /boot/vmlinuz-3.16.0-4-686-pae root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro single
initrd=/install/gtk/initrd.gz
    echo    'Loading initial ramdisk ...'
    initrd   /boot/initrd.img-3.16.0-4-686-pae
}
menuentry 'Debian GNU/Linux, with Linux 3.16.0-4-586' --class debian --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-3.16.0-4-586-advanced-0fc7a234-259a-416b-b674-51eedfb1823b' {
    load_video
    insmod gzio
    if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    echo    'Loading Linux 3.16.0-4-586 ...'
    linux   /boot/vmlinuz-3.16.0-4-586 root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro
initrd=/install/gtk/initrd.gz quiet
    echo    'Loading initial ramdisk ...'
    initrd   /boot/initrd.img-3.16.0-4-586
}
menuentry 'Debian GNU/Linux, with Linux 3.16.0-4-586 (recovery mode)' --class debian --class gnu-linux --
class gnu --class os $menuentry_id_option 'gnulinux-3.16.0-4-586-recovery-0fc7a234-259a-416b-b674-51eedfb1823b' {
    load_video
    insmod gzio
    if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
else
    search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
fi
echo 'Loading Linux 3.16.0-4-586 ...'
linux /boot/vmlinuz-3.16.0-4-586 root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro single
initrd=/install/gtk/initrd.gz
echo 'Loading initial ramdisk ...'
initrd /boot/initrd.img-3.16.0-4-586
}
}
```

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+

```
menuentry "Memory test (memtest86+)" {
    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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
}
```

```
linux16    /boot/memtest86+.bin
}
menuentry "Memory test (memtest86+, serial console 115200)" {
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    linux16    /boot/memtest86+.bin console=ttyS0,115200n8
}
menuentry "Memory test (memtest86+, experimental multiboot)" {
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    multiboot    /boot/memtest86+_multiboot.bin
}
menuentry "Memory test (memtest86+, serial console 115200, experimental multiboot)" {
    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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
```

```
    search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
fi
multiboot    /boot/memtest86+_multiboot.bin console=ttyS0,115200n8
}
### END /etc/grub.d/20_memtest86+ ###

### 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 **update-grub** ou **grub-mkconfig**, plusieurs fichiers sont lus :

Le fichier /boot/grub2/device.map

```
root@debian8:~# cat /boot/grub/device.map
cat: /boot/grub/device.map: No such file or directory
root@debian8:~# which grub-mkdevicemap
/usr/sbin/grub-mkdevicemap
root@debian8:~# grub-mkdevicemap
root@debian8:~# cat /boot/grub/device.map
(hd0)    /dev/disk/by-id/ata-VBOX_HARDDISK_VB823e2981-6120fc97
```

Le fichier /etc/default/grub

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

```
root@debian8:~# 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="initrd=/install/gtk/initrd.gz"

# 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** / **update-grub** sous Debian 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 :

```
root@debian8:~# ls -l /etc/grub.d
total 76
-rwxr-xr-x 1 root root 9424 Mar 23 2015 00_header
-rwxr-xr-x 1 root root 6058 Mar 23 2015 05_debian_theme
-rwxr-xr-x 1 root root 12261 Mar 23 2015 10_linux
-rwxr-xr-x 1 root root 11082 Mar 23 2015 20_linux_xen
-rwxr-xr-x 1 root root 1570 Sep 10 2014 20_memtest86+
-rwxr-xr-x 1 root root 11692 Mar 23 2015 30_os-prober
-rwxr-xr-x 1 root root 1416 Mar 23 2015 30_uefi-firmware
-rwxr-xr-x 1 root root 214 Mar 23 2015 40_custom
-rwxr-xr-x 1 root root 216 Mar 23 2015 41_custom
-rw-r--r-- 1 root root 483 Mar 23 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'Authentication

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

```
root@debian8:~# touch /etc/grub.d/01_users
root@debian8:~# chmod 755 /etc/grub.d/01_users
```

Créez deux mots de passe hashés au format **PBKDF2** en utilisant la commande la commande **`grub-mkpasswd-pbkdf2`** sous Debian :

```
root@debian8:~# grub-mkpasswd-pbkdf2
Enter password: pass123
Reenter password: pass123
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.270AE83E768097679495D221C76D2052E6493B7787E374A7BF6EFFC93E375427866E34C75E668778FAF795B1
9E38FAA5C864A86F1372ECBC1E559E546AF11D71.64C6E790267D3B9B84CF4DFA062A64ECAB4C8831AA80D5FA210C6886C4B1ED96155D66B0
A90F4C6FC32CE59D8FE67E0A465E829E39E3B1BAE180C31C547BBABE
```

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

[`/etc/grub.d/01_users`](#)

```
#!/bin/sh -e
cat <<EOF
set superusers="root"
password_pbkdf2 root
grub.pbkdf2.sha512.10000.270AE83E768097679495D221C76D2052E6493B7787E374A7BF6EFFC93E375427866E34C75E668778FAF
```

```
795B19E38FAA5C864A86F1372ECBC1E559E546AF11D71.64C6E790267D3B9B84CF4DFA062A64ECAB4C8831AA80D5FA210C6886C4B1ED
96155D66B0A90F4C6FC32CE59D8FE67E0A465E829E39E3B1BAE180C31C547BBABE
EOF
```

Il est aussi possible d'utiliser des mots de passe non cryptés. Modifiez donc le fichier **/etc/grub.d/01_users** ainsi :

[/etc/grub.d/01_users](#)

```
#!/bin/sh -e
cat <<EOF
set superusers="root"
password root fenestros
EOF
```

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


```
root@debian8:~# 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-3.16.0-4-686-pae
Found initrd image: /boot/initrd.img-3.16.0-4-686-pae
Found linux image: /boot/vmlinuz-3.16.0-4-586
Found initrd image: /boot/initrd.img-3.16.0-4-586
Found memtest86+ image: /boot/memtest86+.bin
Found memtest86+ multiboot image: /boot/memtest86+_multiboot.bin
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.

Gestionnaires de Démarrages Alternatifs

Systemd-boot

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

U-boot

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

Le Projet Syslinux

SYSLINUX

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

EXTLINUX

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

ISOLINUX

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

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

PXELINUX

Un Charger 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 Charger 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 Charger 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**.

Initramps sous Debian 8

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

Examiner l'image existante

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

```
root@debian8:~# cp /boot/initrd.img-3.16.0-4-686-pae /tmp/custom.gz
root@debian8:~# gunzip /tmp/custom.gz
```

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

```
root@debian8:~# cd /tmp
root@debian8:/tmp# mkdir initrd
root@debian8:/tmp# cd initrd
root@debian8:/tmp/initrd# cpio -idvB < ../custom
.
sbin
sbin/fsck
sbin/logsave
sbin/dmsetup
sbin/e2fsck
sbin/blkid
sbin/mount.fuse
sbin/plymouthd
sbin/fsck.ext4
sbin/mount.ntfs
```



```
sbin/udevadm
sbin/modprobe
sbin/mount.ntfs-3g
sbin/sulogin
...
conf
conf/arch.conf
conf/conf.d
conf/conf.d/resume
conf/initramfs.conf
10250 blocks
```

Installez maintenant le paquet **tree** :

```
root@debian8:/tmp/initrd# apt-get install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.9 kB of archives.
After this operation, 137 kB of additional disk space will be used.
Get:1 http://ftp.fr.debian.org/debian/ jessie/main tree i386 1.7.0-3 [47.9 kB]
Fetched 47.9 kB in 0s (263 kB/s)
Selecting previously unselected package tree.
(Reading database ... 167512 files and directories currently installed.)
Preparing to unpack .../archives/tree_1.7.0-3_i386.deb ...
Unpacking tree (1.7.0-3) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up tree (1.7.0-3) ...
```

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

```
root@debian8:/tmp/initrd# tree | more
```

```
.
├── bin
│   ├── [
│   ├── [[
│   ├── ar
│   ├── ash
│   ├── awk
│   ├── basename
│   ├── bunzip2
│   ├── busybox
│   ├── bzip2
│   ├── cal
│   ├── cat
│   ├── chgrp
│   ├── chmod
│   ├── chown
│   ├── chroot
│   ├── chvt
│   ├── clear
│   ├── cmp
│   ├── cp
│   ├── cpio
│   ├── cttyhack
│   ├── cut
│   ├── date
│   ├── dc
│   ├── dd
│   ├── deallocvt
│   ├── df
│   └── diff
--More--
```

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

```
root@debian8:/tmp/initrd# ls -l
total 44
drwxr-xr-x 2 root root 4096 Oct 29 16:17 bin
drwxr-xr-x 3 root root 4096 Oct 29 16:17 conf
drwxr-xr-x 7 root root 4096 Oct 29 16:17 etc
-rwxr-xr-x 1 root root 7057 Oct 29 16:17 init
drwxr-xr-x 8 root root 4096 Oct 29 16:17 lib
drwxr-xr-x 2 root root 4096 Oct 29 16:17 run
drwxr-xr-x 2 root root 4096 Oct 29 16:17 sbin
drwxr-xr-x 7 root root 4096 Oct 29 16:17 scripts
drwxr-xr-x 4 root root 4096 Oct 29 16:17 usr
drwxr-xr-x 3 root root 4096 Oct 29 16:17 var
```

Le script **init**

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

```
-rwxr-xr-x 1 root root 7057 Oct 29 16:17 init
```



A faire : Passez en revue le contenu du script.

La commande **mkinitramfs**

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

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 :

```
# 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@debian8:/tmp/initrd# mkinitramfs -o usbinitramfs-`uname -r`.img
```

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

```
root@debian8:/tmp/initrd# ls -l /tmp/initrd/usbinitramfs-3.16.0-4-686-pae.img
-rw-r--r-- 1 root root 21454339 Oct 29 16:35 /tmp/initrd/usbinitramfs-3.16.0-4-686-pae.img
```

Déplacez votre fichier usbinitramfs au répertoire /boot :

```
root@debian8:/tmp/initrd# mv usbinitramfs-3.16.0-4-686-pae.img /boot
```

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

```
#!/bin/sh -e
cat << EOF
menuentry 'Debian GNU/Linux et usbinitramfs' --class debian --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-simple-0fc7a234-259a-416b-b674-51eedfb1823b' {
    load_video
    insmod gzio
    if [ x$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 0fc7a234-259a-416b-b674-51eedfb1823b
    else
        search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
    fi
    echo    'Loading Linux 3.16.0-4-686-pae ...'
    linux   /boot/vmlinuz-3.16.0-4-686-pae root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro
    initrd=/install/gtk/initrd.gz quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/usbinitramfs-3.16.0-4-686-pae.img
}
EOF
```

Rendez ce fichier exécutable :

```
root@debian8:/tmp/initrd# chmod +x /etc/grub.d/09_usbdebian
```

Avant de continuer, supprimer le fichier **/etc/grub.d/01_users** et éditez le contenu du fichier **/etc/grub.d/40_custom** ainsi :

```
#!/bin/sh
exec tail -n +3 $0
# 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.
```

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

```
root@debian8:/tmp/initrd# update-grub
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-3.16.0-4-686-pae
Found initrd image: /boot/initrd.img-3.16.0-4-686-pae
Found linux image: /boot/vmlinuz-3.16.0-4-586
Found initrd image: /boot/initrd.img-3.16.0-4-586
Found memtest86+ image: /boot/memtest86+.bin
Found memtest86+ multiboot image: /boot/memtest86+_multiboot.bin
done
```



Important : Notez l'utilisation de la commande **update-grub** en lieu et place de la commande **grub-mkconfig -o /boot/grub/grub.cfg**.

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

```
...
### BEGIN /etc/grub.d/09_usbdebian ###
menuentry 'Debian GNU/Linux et usbinitramfs' --class debian --class gnu-linux --class gnu --class os 'gnulinux-
simple-0fc7a234-259a-416b-b674-51eedfb1823b' {
    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 0fc7a234-259a-416b-b674-51eedfb1823b
else
    search --no-floppy --fs-uuid --set=root 0fc7a234-259a-416b-b674-51eedfb1823b
fi
echo 'Loading Linux 3.16.0-4-686-pae ...'
linux /boot/vmlinuz-3.16.0-4-686-pae root=UUID=0fc7a234-259a-416b-b674-51eedfb1823b ro
initrd=/install/gtk/initrd.gz quiet
echo 'Loading initial ramdisk ...'
initrd /boot/usbinitramfs-3.16.0-4-686-pae.img
}
### END /etc/grub.d/09_usbdebian ###

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



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

Etape	Description
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

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 quelquechose 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

Directive	Description
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
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

Ligne	Description
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	

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**.

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

Code Retour	Description
100-149	Réserves pour la distribution
150-199	Réservés pour l'application
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 .

Mot Clef	Description
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 .
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.

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@debian6:/etc/init.d# for rep in /etc/rc[2345].d; do echo "dans $rep : "; ls $rep/S*; done
dans /etc/rc2.d :
/etc/rc2.d/S01vboxadd          /etc/rc2.d/S18kerneloops
/etc/rc2.d/S02vboxadd-service /etc/rc2.d/S18loadcpufreq
/etc/rc2.d/S14portmap          /etc/rc2.d/S19avahi-daemon
/etc/rc2.d/S15nfs-common       /etc/rc2.d/S19bluetooth
/etc/rc2.d/S17binfmt-support   /etc/rc2.d/S19cpufreutils
```

```
/etc/rc2.d/S17fancontrol      /etc/rc2.d/S19network-manager
/etc/rc2.d/S17rsyslog         /etc/rc2.d/S20cups
/etc/rc2.d/S17sudo            /etc/rc2.d/S20gdm3
/etc/rc2.d/S18acpid           /etc/rc2.d/S20saned
/etc/rc2.d/S18anacron         /etc/rc2.d/S21bootlogs
/etc/rc2.d/S18atd             /etc/rc2.d/S22rc.local
/etc/rc2.d/S18cron            /etc/rc2.d/S22rmnologin
/etc/rc2.d/S18dbus            /etc/rc2.d/S22stop-bootlogd
/etc/rc2.d/S18exim4
dans /etc/rc3.d :
/etc/rc3.d/S01vboxadd         /etc/rc3.d/S18kerneloops
/etc/rc3.d/S02vboxadd-service /etc/rc3.d/S18loadcpufreq
/etc/rc3.d/S14portmap         /etc/rc3.d/S19avahi-daemon
/etc/rc3.d/S15nfs-common      /etc/rc3.d/S19bluetooth
/etc/rc3.d/S17binfmt-support  /etc/rc3.d/S19cpufrequtils
/etc/rc3.d/S17fancontrol      /etc/rc3.d/S19network-manager
/etc/rc3.d/S17rsyslog         /etc/rc3.d/S20cups
/etc/rc3.d/S17sudo            /etc/rc3.d/S20gdm3
/etc/rc3.d/S18acpid           /etc/rc3.d/S20saned
/etc/rc3.d/S18anacron         /etc/rc3.d/S21bootlogs
/etc/rc3.d/S18atd             /etc/rc3.d/S22rc.local
/etc/rc3.d/S18cron            /etc/rc3.d/S22rmnologin
/etc/rc3.d/S18dbus            /etc/rc3.d/S22stop-bootlogd
/etc/rc3.d/S18exim4
dans /etc/rc4.d :
/etc/rc4.d/S01vboxadd         /etc/rc4.d/S18kerneloops
/etc/rc4.d/S02vboxadd-service /etc/rc4.d/S18loadcpufreq
/etc/rc4.d/S14portmap         /etc/rc4.d/S19avahi-daemon
/etc/rc4.d/S15nfs-common      /etc/rc4.d/S19bluetooth
/etc/rc4.d/S17binfmt-support  /etc/rc4.d/S19cpufrequtils
/etc/rc4.d/S17fancontrol      /etc/rc4.d/S19network-manager
/etc/rc4.d/S17rsyslog         /etc/rc4.d/S20cups
/etc/rc4.d/S17sudo            /etc/rc4.d/S20gdm3
/etc/rc4.d/S18acpid           /etc/rc4.d/S20saned
```

```
/etc/rc4.d/S18anacron      /etc/rc4.d/S21bootlogs
/etc/rc4.d/S18atd         /etc/rc4.d/S22rc.local
/etc/rc4.d/S18cron        /etc/rc4.d/S22rmnologin
/etc/rc4.d/S18dbus        /etc/rc4.d/S22stop-bootlogd
/etc/rc4.d/S18exim4
dans /etc/rc5.d :
/etc/rc5.d/S01vboxadd      /etc/rc5.d/S18kerneloops
/etc/rc5.d/S02vboxadd-service /etc/rc5.d/S18loadcpufreq
/etc/rc5.d/S14portmap      /etc/rc5.d/S19avahi-daemon
/etc/rc5.d/S15nfs-common   /etc/rc5.d/S19bluetooth
/etc/rc5.d/S17binfmt-support /etc/rc5.d/S19cpufrequtils
/etc/rc5.d/S17fancontrol   /etc/rc5.d/S19network-manager
/etc/rc5.d/S17rsyslog      /etc/rc5.d/S20cups
/etc/rc5.d/S17sudo         /etc/rc5.d/S20gdm3
/etc/rc5.d/S18acpid        /etc/rc5.d/S20saned
/etc/rc5.d/S18anacron      /etc/rc5.d/S21bootlogs
/etc/rc5.d/S18atd         /etc/rc5.d/S22rc.local
/etc/rc5.d/S18cron        /etc/rc5.d/S22rmnologin
/etc/rc5.d/S18dbus        /etc/rc5.d/S22stop-bootlogd
/etc/rc5.d/S18exim4
```



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/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/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@debian6:/etc/init.d# for rep in /etc/rc[016].d; do echo "dans $rep :"; ls $rep/K*; done
```

```

dans /etc/rc0.d :
/etc/rc0.d/K01alsa-utils      /etc/rc0.d/K02avahi-daemon
/etc/rc0.d/K01anacron         /etc/rc0.d/K02vboxadd
/etc/rc0.d/K01atd             /etc/rc0.d/K03sendsigs
/etc/rc0.d/K01bluetooth       /etc/rc0.d/K04rsyslog
/etc/rc0.d/K01exim4           /etc/rc0.d/K05umountnfs.sh
/etc/rc0.d/K01fuse            /etc/rc0.d/K06nfs-common
/etc/rc0.d/K01gdm3            /etc/rc0.d/K06portmap
/etc/rc0.d/K01kerneloops      /etc/rc0.d/K07hwclock.sh
/etc/rc0.d/K01network-manager /etc/rc0.d/K07networking
/etc/rc0.d/K01saned           /etc/rc0.d/K08ifupdown
/etc/rc0.d/K01unattended-upgrades /etc/rc0.d/K09umountfs
/etc/rc0.d/K01urandom         /etc/rc0.d/K10umountroot
/etc/rc0.d/K01vboxadd-service /etc/rc0.d/K11halt
dans /etc/rc1.d :
/etc/rc1.d/K01alsa-utils      /etc/rc1.d/K01network-manager
/etc/rc1.d/K01anacron         /etc/rc1.d/K01saned
/etc/rc1.d/K01atd            /etc/rc1.d/K01vboxadd-service
/etc/rc1.d/K01bluetooth      /etc/rc1.d/K02avahi-daemon
/etc/rc1.d/K01cups           /etc/rc1.d/K02vboxadd
/etc/rc1.d/K01exim4          /etc/rc1.d/K04rsyslog
/etc/rc1.d/K01gdm3           /etc/rc1.d/K06nfs-common
/etc/rc1.d/K01kerneloops     /etc/rc1.d/K06portmap
dans /etc/rc6.d :
/etc/rc6.d/K01alsa-utils      /etc/rc6.d/K02avahi-daemon
/etc/rc6.d/K01anacron         /etc/rc6.d/K02vboxadd
/etc/rc6.d/K01atd             /etc/rc6.d/K03sendsigs
/etc/rc6.d/K01bluetooth       /etc/rc6.d/K04rsyslog
/etc/rc6.d/K01exim4           /etc/rc6.d/K05umountnfs.sh
/etc/rc6.d/K01fuse            /etc/rc6.d/K06nfs-common
/etc/rc6.d/K01gdm3            /etc/rc6.d/K06portmap
/etc/rc6.d/K01kerneloops      /etc/rc6.d/K07hwclock.sh
/etc/rc6.d/K01network-manager /etc/rc6.d/K07networking
/etc/rc6.d/K01saned           /etc/rc6.d/K08ifupdown
```

```
/etc/rc6.d/K01unattended-upgrades /etc/rc6.d/K09umountfs
/etc/rc6.d/K01urandom             /etc/rc6.d/K10umountroot
/etc/rc6.d/K01vboxadd-service     /etc/rc6.d/K11reboot
```



Important : Ici le principe est le même sauf que la lettre **K** indique au script **rc** que le script dans **/etc/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

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

Option	Description
- -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.

Démarrer Debian 8 avec systemd

Debian 8, comme beaucoup d'autres distributions, ont abandonné **Upstart** pour **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** :

```
root@debian8:~# 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-ata3-host2-target2:0:0-2:0:0:0-block-sr0.device	loaded	active	plugged	
VBOX_CD-ROM				
sys-devices-pci0000:00-0000:00:03.0-net-eth0.device	loaded	active	plugged	82540EM Gigabit Ethernet
Controller (PRO/1000 MT Desktop Adapte				
sys-devices-pci0000:00-0000:00:05.0-sound-card0.device	loaded	active	plugged	82801AA AC'97 Audio
Controller				
sys-devices-pci0000:00-0000:00:0d.0-ata1-host0-target0:0:0-0:0:0:0-block-sda-sda1.device	loaded	active	plugged	
VBOX_HARDDISK 1				
sys-devices-pci0000:00-0000:00:0d.0-ata1-host0-target0:0:0-0:0:0:0-block-sda-sda2.device	loaded	active	plugged	
VBOX_HARDDISK 2				
sys-devices-pci0000:00-0000:00:0d.0-ata1-host0-target0:0:0-0:0:0:0-block-sda-sda5.device	loaded	active	plugged	
VBOX_HARDDISK 5				
sys-devices-pci0000:00-0000:00:0d.0-ata1-host0-target0:0:0-0:0:0:0-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-ttyS2.device /sys/devices/platform/serial8250/tty/ttyS2	loaded active plugged	
sys-devices-platform-serial8250-tty-ttyS3.device /sys/devices/platform/serial8250/tty/ttyS3	loaded active plugged	
sys-module-fuse.device	loaded active plugged	/sys/module/fuse
sys-subsystem-net-devices-eth0.device Controller (PR0/1000 MT Desktop Adapte - .mount	loaded active plugged	82540EM Gigabit Ethernet
dev-hugepages.mount	loaded active mounted	/
dev-mqueue.mount	loaded active mounted	Huge Pages File System
System	loaded active mounted	POSIX Message Queue File
etc-machine\x2did.mount	loaded active mounted	/etc/machine-id
run-rpc_pipefs.mount	loaded active mounted	/run/rpc_pipefs
run-user-1000.mount	loaded active mounted	/run/user/1000
run-user-120.mount	loaded active mounted	/run/user/120
sys-fs-fuse-connections.mount	loaded active mounted	FUSE Control File System
sys-kernel-debug.mount	loaded active mounted	Debug File System
cups.path	loaded active waiting	CUPS Printer Service Spool
systemd-ask-password-console.path to Console Directory Watch	loaded active waiting	Dispatch Password Requests
systemd-ask-password-wall.path to Wall Directory Watch	loaded active waiting	Forward Password Requests
session-2.scope	loaded active running	Session 2 of user trainee
session-cl.scope	loaded active abandoned	Session cl of user lightdm
accounts-daemon.service	loaded active running	Accounts Service
atd.service	loaded active running	Deferred execution
scheduler		
lines 1-31		

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

```

root@debian8:~# systemctl list-units --all | grep inactive | more
proc-sys-fs-binfmt_misc.mount                                loaded    inactive
dead          Arbitrary Executable File Form
ats File System
run-vmblock\x2dfuse.mount                                    loaded    inactive
dead          VMware vmblock fuse mount
sys-kernel-config.mount                                      loaded    inactive
dead          Configuration File System
tmp.mount                                                  loaded    inactive
dead          Temporary Directory
● var-lock.mount                                             not-found  inactive
dead          var-lock.mount
● var-run.mount                                              not-found  inactive
dead          var-run.mount
systemd-ask-password-plymouth.path                          loaded    inactive
dead          Forward Password Requests to P
lymouth Directory Watch
alsa-restore.service                                        loaded    inactive
dead          Restore Sound Card State
alsa-state.service                                          loaded    inactive
dead          Manage Sound Card State (resto
re and store)
alsa-store.service                                           loaded    inactive
dead          Store Sound Card State
anacron.service                                              loaded    inactive
dead          Run anacron jobs
● bootmisc.service                                           masked     inactive
dead          bootmisc.service
● clamav-daemon.service                                       not-found  inactive
dead          clamav-daemon.service
● console-screen.service                                      not-found  inactive
dead          console-screen.service
debian-fixup.service                                         loaded    inactive
dead          Various fixups to make systemd

```

```

work better on Debian
  emergency.service                                loaded    inactive
dead      Emergency Shell
● festival.service                                not-found inactive
dead      festival.service
  getty-static.service                             loaded    inactive
dead      getty on tty2-tty6 if dbus and
  logind are not available
● greylist.service                                not-found inactive
dead      greylist.service
● keymap.service                                  not-found inactive
dead      keymap.service
● mountkernfs.service                             masked    inactive
dead      mountkernfs.service
● mysql.service                                   not-found inactive
dead      mysql.service
  open-vm-tools.service                             loaded    inactive
dead      Service for virtual machines h
osted on VMware
  plymouth-quit-wait.service                       loaded    inactive
dead      Wait for Plymouth Boot Screen
to Quit
--More--

```

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

```

root@debian8:~# systemctl list-unit-files | more
UNIT FILE                                STATE
proc-sys-fs-binfmt_misc.automount      static
dev-hugepages.mount                    static
dev-mqueue.mount                        static
proc-sys-fs-binfmt_misc.mount           static
run-vmblock\x2dfuse.mount               enabled
sys-fs-fuse-connections.mount           static

```


sys-kernel-config.mount	static
sys-kernel-debug.mount	static
tmp.mount	disabled
cups.path	enabled
systemd-ask-password-console.path	static
systemd-ask-password-plymouth.path	static
systemd-ask-password-wall.path	static
session-2.scope	static
session-cl.scope	static
accounts-daemon.service	enabled
alsa-restore.service	static
alsa-state.service	static
alsa-store.service	static
alsa-utils.service	masked
anacron-resume.service	enabled
anacron.service	enabled
atd.service	enabled
auditd.service	enabled
autovt@.service	disabled
avahi-daemon.service	enabled
bluetooth.service	enabled
bootlogd.service	masked
bootlogs.service	masked
bootmisc.service	masked
--More--	

Options de la Commande systemctl

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

```
root@debian8:~# 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
-r --recursive      Show unit list of host and local containers
--reverse           Show reverse dependencies with 'list-dependencies'
--job-mode=MODE     Specify how to deal with already queued jobs, when
                  queueing a new job
--show-types        When showing sockets, explicitly show their type
-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
```

lines 1-31

Fichiers de Configuration

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

```
root@debian8:~# ls -l /etc/systemd/system
total 52
drwxr-xr-x 2 root root 4096 Oct 23 16:34 bluetooth.target.wants
lrwxrwxrwx 1 root root 37 Oct 23 16:34 dbus-org.bluez.service -> /lib/systemd/system/bluetooth.service
lrwxrwxrwx 1 root root 40 Oct 23 16:34 dbus-org.freedesktop.Avahi.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root 40 Oct 23 16:34 dbus-org.freedesktop.ModemManager1.service -> /lib/systemd/system/ModemManager.service
lrwxrwxrwx 1 root root 53 Oct 23 16:34 dbus-org.freedesktop.nm-dispatcher.service -> /lib/systemd/system/NetworkManager-dispatcher.service
lrwxrwxrwx 1 root root 35 Oct 23 16:34 display-manager.service -> /lib/systemd/system/lightdm.service
drwxr-xr-x 2 root root 4096 Oct 23 16:34 getty.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 graphical.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 halt.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 hibernate.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 hybrid-sleep.target.wants
drwxr-xr-x 2 root root 4096 Oct 28 06:37 multi-user.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 paths.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 poweroff.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 printer.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 reboot.target.wants
drwxr-xr-x 2 root root 4096 Oct 23 16:34 sockets.target.wants
lrwxrwxrwx 1 root root 31 Oct 23 16:34 sshd.service -> /lib/systemd/system/ssh.service
drwxr-xr-x 2 root root 4096 Oct 23 16:34 suspend.target.wants
lrwxrwxrwx 1 root root 35 Oct 23 16:34 syslog.service -> /lib/systemd/system/rsyslog.service
```

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

```
root@debian8:~# ls -l /lib/systemd/system | more
```

```

total 904
-rw-r--r-- 1 root root 411 Oct 1 2014 accounts-daemon.service
-rw-r--r-- 1 root root 476 Jul 8 2014 alsa-restore.service
-rw-r--r-- 1 root root 526 Jul 8 2014 alsa-state.service
-rw-r--r-- 1 root root 373 Jul 8 2014 alsa-store.service
lrwxrwxrwx 1 root root 9 Oct 23 16:34 alsa-utils.service -> /dev/null
-rw-r--r-- 1 root root 283 Dec 28 2014 anacron-resume.service
-rw-r--r-- 1 root root 183 Dec 28 2014 anacron.service
-rw-r--r-- 1 root root 169 Sep 30 2014 atd.service
-rw-r--r-- 1 root root 688 Dec 9 2014 auditd.service
lrwxrwxrwx 1 root root 14 Aug 31 00:04 autovt@.service -> getty@.service
-rw-r--r-- 1 root root 1044 Apr 14 2015 avahi-daemon.service
-rw-r--r-- 1 root root 874 Apr 14 2015 avahi-daemon.socket
-rw-r--r-- 1 root root 524 Aug 31 00:04 basic.target
drwxr-xr-x 2 root root 4096 Oct 23 16:34 basic.target.wants
-rw-r--r-- 1 root root 338 Jan 3 2015 bluetooth.service
-rw-r--r-- 1 root root 379 Aug 31 00:04 bluetooth.target
lrwxrwxrwx 1 root root 9 Aug 31 00:04 bootlogd.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 bootlogs.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 bootmisc.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 checkfs.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 checkroot-bootclean.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 checkroot.service -> /dev/null
-rw-r--r-- 1 root root 298 Dec 9 2014 colord.service
-rw-r--r-- 1 root root 150 Oct 23 2014 configure-printer@.service
-rw-r--r-- 1 root root 737 Aug 31 00:04 console-getty.service
-rw-r--r-- 1 root root 741 Aug 31 00:04 console-shell.service
-rw-r--r-- 1 root root 783 Aug 31 00:04 container-getty@.service
-rw-r--r-- 1 root root 251 May 14 09:18 cron.service
lrwxrwxrwx 1 root root 9 Aug 31 00:04 cryptdisks-early.service -> /dev/null
lrwxrwxrwx 1 root root 9 Aug 31 00:04 cryptdisks.service -> /dev/null
--More--

```

Par exemple, sous Debian 8, le service sshd est configuré par le fichier **/etc/systemd/system/ssh.service** :

```
root@debian8:~# cat /etc/systemd/system/sshd.service
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

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

Dans ce 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 auditd.service,**
 - 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=on-failure,**
 - Cette ligne indique quand le service doit être re-démarré.

Système 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 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 :

```
root@debian8:~# cat /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 **/lib/systemd/system/default.target** :

```
root@debian8:~# ls -l /lib/systemd/system/default.target
lrwxrwxrwx 1 root root 16 Aug 31 00:04 /lib/systemd/system/default.target -> graphical.target
```

La Commande systemd-analyze

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

```
root@debian8:~# systemd-analyze
Startup finished in 6.296s (kernel) + 17.035s (userspace) = 23.332s
```

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@debian8:~# systemd-analyze blame
6.942s NetworkManager.service
6.781s ModemManager.service
6.672s exim4.service
5.483s accounts-daemon.service
3.697s alsa-restore.service
3.697s pppd-dns.service
3.587s systemd-logind.service
3.554s vboxadd-x11.service
3.545s vboxadd.service
```

```
3.543s rc-local.service
3.539s minissdpd.service
3.538s irqbalance.service
3.527s gdomap.service
3.527s speech-dispatcher.service
3.461s avahi-daemon.service
2.895s networking.service
2.726s kbd.service
1.652s polkitd.service
1.459s keyboard-setup.service
1.434s lightdm.service
1.055s rsyslog.service
955ms systemd-tmpfiles-setup.service
864ms systemd-modules-load.service
830ms vboxadd-service.service
778ms systemd-journal-flush.service
697ms nfs-common.service
644ms systemd-tmpfiles-setup-dev.service
600ms console-setup.service
555ms rpcbind.service
494ms systemd-setup-dgram-qlen.service
456ms sys-kernel-debug.mount
```

lines 1-31

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

```
root@debian8:~# 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.

ssh.service @10.074s
└─basic.target @9.951s
   └─paths.target @9.950s
      └─cups.path @9.950s
```



```
└─sysinit.target @9.940s
  └─nfs-common.service @9.242s +697ms
    └─rpcbind.target @9.242s
      └─rpcbind.service @8.684s +555ms
        └─network-online.target @8.683s
          └─network.target @8.683s
            └─networking.service @5.788s +2.895s
              └─systemd-random-seed.service @5.458s +328ms
                └─systemd-remount-fs.service @5.298s +157ms
                  └─keyboard-setup.service @3.837s +1.459s
                    └─systemd-udev.service @3.702s +126ms
                      └─systemd-tmpfiles-setup-dev.service @3.057s +644ms
                        └─kmod-static-nodes.service @2.858s +193ms
                          └─system.slice @2.805s
                            └─.slice @2.804s
```

Options de la Commande

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

```
root@debian8:~# 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

Sous Debian 8, sshd.service est un lien vers ssh.service :

```
root@debian8:~# ls -l /etc/systemd/system/sshd.service
lrwxrwxrwx 1 root root 31 Oct 23 16:34 /etc/systemd/system/sshd.service -> /lib/systemd/system/ssh.service
```

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

```
root@debian8:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Sun 2015-11-01 09:35:03 CET; 10min ago
   Main PID: 2024 (sshd)
   CGroup: /system.slice/ssh.service
```

```
└─2024 /usr/sbin/sshd -D
```

```
Nov 01 09:35:03 debian8 sshd[2024]: Server listening on 0.0.0.0 port 22.  
Nov 01 09:35:03 debian8 sshd[2024]: Server listening on :: port 22.
```

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

```
root@debian8:~# systemctl stop ssh.service  
root@debian8:~# systemctl status ssh.service  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)  
   Active: inactive (dead) since Sun 2015-11-01 09:46:13 CET; 2s ago  
   Main PID: 2024 (code=exited, status=0/SUCCESS)  
  
Nov 01 09:35:03 debian8 sshd[2024]: Server listening on 0.0.0.0 port 22.  
Nov 01 09:35:03 debian8 sshd[2024]: Server listening on :: port 22.
```

Pour démarrer un service, utilisez la commande suivante :

```
root@debian8:~# systemctl start ssh.service  
root@debian8:~# systemctl status ssh.service  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)  
   Active: active (running) since Sun 2015-11-01 09:47:10 CET; 6s ago  
   Main PID: 2180 (sshd)  
   CGroup: /system.slice/ssh.service  
           └─2180 /usr/sbin/sshd -D  
  
Nov 01 09:47:10 debian8 sshd[2180]: Server listening on 0.0.0.0 port 22.  
Nov 01 09:47:10 debian8 sshd[2180]: Server listening on :: port 22.
```

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

```
root@debian8:~# systemctl disable ssh.service
```

```
Synchronizing state for ssh.service with sysvinit using update-rc.d...
Executing /usr/sbin/update-rc.d ssh defaults
Executing /usr/sbin/update-rc.d ssh disable
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB defaults (empty).
Removed symlink /etc/systemd/system/sshd.service.
root@debian8:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled)
   Active: active (running) since Sun 2015-11-01 09:47:10 CET; 1min 7s ago
 Main PID: 2180 (sshd)
    CGroup: /system.slice/ssh.service
            └─2180 /usr/sbin/sshd -D

Nov 01 09:47:10 debian8 sshd[2180]: Server listening on 0.0.0.0 port 22.
Nov 01 09:47:10 debian8 sshd[2180]: Server listening on :: port 22.
```

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

```
root@debian8:~# systemctl enable ssh.service
Synchronizing state for ssh.service with sysvinit using update-rc.d...
Executing /usr/sbin/update-rc.d ssh defaults
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB defaults (empty).
Executing /usr/sbin/update-rc.d ssh enable
Created symlink from /etc/systemd/system/sshd.service to /lib/systemd/system/ssh.service.
root@debian8:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since Sun 2015-11-01 09:47:10 CET; 1min 43s ago
 Main PID: 2180 (sshd)
    CGroup: /system.slice/ssh.service
            └─2180 /usr/sbin/sshd -D
```

```
Nov 01 09:47:10 debian8 sshd[2180]: Server listening on 0.0.0.0 port 22.  
Nov 01 09:47:10 debian8 sshd[2180]: 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 :

```
root@debian8:~# 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 :

```
root@debian8:~# shutdown -r +30
Shutdown scheduled for Sun 2015-11-01 10:21:04 CET, use 'shutdown -c' to cancel.
```

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

```
trainee@debian8:~$ su -  
Password:  
root@debian8:~# shutdown -c  
  
Broadcast message from root@debian8 (Sun 2015-11-01 09:53:34 CET):  
  
The system shutdown has been cancelled at Sun 2015-11-01 09:54:34 CET!  
  
root@debian8:~#
```

Retournez au premier terminal et constatez le message affiché :

```
root@debian8:~#  
Broadcast message from root@debian8 (Sun 2015-11-01 09:53:34 CET):  
  
The system shutdown has been cancelled at Sun 2015-11-01 09:54:34 CET!
```

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 :

```
root@debian8:~# 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 :

```
root@debian8:~# 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 :

```
root@debian8:~# 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 © 2020 Hugh Norris

</html>

From:

<https://www.ittraining.team/> - **www.ittraining.team**

Permanent link:

<https://www.ittraining.team/doku.php?id=elearning:workbooks:debian:6:junior:l116>

Last update: **2020/08/07 13:49**

