

Dernière mise-à-jour : 2020/01/30 03:27

101.2 - Démarrer le système (3/60)

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

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



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

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

BIOS, EFI et OpenFirmware

Systèmes à base du BIOS

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

- Tester les composants et les circuits,
- Faire appel au BIOS de la carte graphique pour initialiser le système d'affichage,

- Détecter les périphériques de stockage,
- Lancer le **Charger de Démarrage** du système d'exploitation en utilisant le **bootstrap loader**.

Charger de Démarrage

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

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

Systèmes à base de l'EFI

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

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

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



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



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

Autres Systèmes

Les systèmes utilisant des processeurs autre qu'un x86 ou x86-64 utilisent un logiciel tel [OpenFirmware](#).

Gestionnaires de Démarrage

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

- GRUB (Grand Unified Boot Loader)

Cependant il en existe d'autres :

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

LILO

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

La commande LILO

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

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

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

Codes Erreur de LILO

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

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

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





Important : LILO ne gère pas les systèmes UEFI.

Grub Legacy sous RHEL/CentOS 6



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

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

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

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

Il est aussi possible d'utiliser la commande :

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

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

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

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

ou :

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

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

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

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

```
A> fdisk /mbr [Entrée]
```

Configurer GRUB Legacy

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

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

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



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

Ce fichier comporte plusieurs sections :

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

La Section Globale

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

Une Section spécifique à un OS

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

Configurer l'Authentification

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

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

Le mot de passe doit être chiffré avec MD5 :

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

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

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

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

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



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

Modifier la Configuration de GRUB Legacy en Ligne de Commande

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

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

En mode édition notez l'utilisation des touches suivantes :

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

GRUB 2 sous RHEL/CentOS 7

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

Le lancement de GRUB 2 se fait en trois étapes :

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

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

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

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

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

```
[root@centos7 ~]# cat /boot/grub2/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub2-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

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

if [ -s $prefix/grubenv ]; then
  load_env
fi
```

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

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

export menuentry_id_option

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

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

function load_video {
    if [ x$feature_all_video_module = xy ]; then
```

```
    insmod all_video
else
    insmod efi_gop
    insmod efi_uga
    insmod ieee1275_fb
    insmod vbe
    insmod vga
    insmod video_bochs
    insmod video_cirrus
fi
}

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

### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --
class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-b35de665-5ec8-4226-
a533-58a1b567ac91' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
```

```
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-  
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309  
    else  
        search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309  
    fi  
    linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro  
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8  
    initrd16 /initramfs-3.10.0-229.4.2.el7.x86_64.img  
}  
menuentry 'CentOS Linux, with Linux 3.10.0-123.el7.x86_64' --class centos --class gnu-linux --class gnu --class  
os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-b35de665-5ec8-4226-  
a533-58a1b567ac91' {  
    load_video  
    set gfxpayload=keep  
    insmod gzio  
    insmod part_msdos  
    insmod xfs  
    set root='hd0,msdos1'  
    if [ x$feature_platform_search_hint = xy ]; then  
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-  
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309  
    else  
        search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309  
    fi  
    linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro vconsole.keymap=fr  
crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8  
    initrd16 /initramfs-3.10.0-123.el7.x86_64.img  
}  
menuentry 'CentOS Linux, with Linux 0-rescue-a2feb9eb09b1488da0f23b99a66350f8' --class centos --class gnu-linux -  
-class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-a2feb9eb09b1488da0f23b99a66350f8-  
advanced-b35de665-5ec8-4226-a533-58a1b567ac91' {  
    load_video  
    insmod gzio  
    insmod part_msdos
```

```
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
else
    search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
fi
linux16 /vmlinuz-0-rescue-a2feb9eb09b1488da0f23b99a66350f8 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet
initrd16 /initramfs-0-rescue-a2feb9eb09b1488da0f23b99a66350f8.img
}
if [ "x$default" = 'CentOS Linux, with Linux 3.10.0-123.el7.x86_64' ]; then default='Advanced options for CentOS Linux>CentOS Linux, with Linux 3.10.0-123.el7.x86_64'; fi;
### END /etc/grub.d/10_linux ###

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

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

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

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

Notez que ce fichier ne doit pas être modifié manuellement. En effet, il est généré par la commande **grub2-mkconfig** sous RHEL/CentOS 7. La commande grub2-mkconfig prend en argument l'emplacement du fichier destination, par exemple :

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

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

Le fichier /boot/grub2/device.map

```
[root@centos7 ~]# cat /boot/grub2/device.map
# this device map was generated by anaconda
(hd0)      /dev/sda
(hd1)      /dev/sda
```

Le fichier `/etc/default/grub`

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

```
[root@centos7 ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="vconsole.keymap=fr crashkernel=auto vconsole.font=latacyrheb-sun16 rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```



Important : Notez que toute modification de ce fichier nécessite l'exécution de la commande **grub2-mkconfig** sous RHEL/CentOS 7 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

Directive	Description
GRUB_BADRAM	Permet de spécifier de la mémoire défaillante

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@centos7 ~]# ls -l /etc/grub.d
total 68
-rwxr-xr-x. 1 root root 8702 Mar 26 09:27 00_header
-rwxr-xr-x. 1 root root 992 Oct 17 2014 00_tuned
-rwxr-xr-x. 1 root root 10114 Mar 26 09:27 10_linux
-rwxr-xr-x. 1 root root 10275 Mar 26 09:27 20_linux_xen
-rwxr-xr-x. 1 root root 2559 Mar 26 09:27 20_ppc_terminfo
-rwxr-xr-x. 1 root root 11169 Mar 26 09:27 30_os-prober
-rwxr-xr-x. 1 root root 214 Mar 26 09:27 40_custom
-rwxr-xr-x. 1 root root 216 Mar 26 09:27 41_custom
-rw-r--r--. 1 root root 483 Mar 26 09:27 README
```

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

Configurer l'Authentification

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

```
[root@centos7 ~]# touch /etc/grub.d/01_users
[root@centos7 ~]# chmod 755 /etc/grub.d/01_users
```

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

```
[root@centos7 ~]# grub2-mkpasswd-pbkdf2
Enter password: pass123
Reenter password: pass123
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.0298C1C613A451C46FBC95BB2AC7A41BCEC1C61512EF785BD81E3B65DFF9D57ED4ADF8906C3EF33C22C06FBD
D366E1C118FC41110BD646A4D49EF86EFD0573BF.E14A45900096D773BE99BEA9AB8D4FA81431458952798B997D4FC9E0850426F679897937
1B8EBD331DB33AE8FEAE25E6773156D42F21B884DBA405546782B3BD
[root@centos7 ~]# grub2-mkpasswd-pbkdf2
Enter password: pass456
Reenter password: pass456
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.161D4183DC832357403296ED05961FCF494AED9E20DC21C84EA89085EB9EF5AAE4C7D4A276AA5CC21F9C224B
2ECA010B915B4830E9648A7398EB4A91E7E3D252.8277512B849FF727FDD0716D1D4CDC6B92E53918F665282E02133AAD1046EB10273A2BC7
0D76558FFC34A0C0C8BE5132E4C4C02C7C9C1A567BD5365D77350FCF
```

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.sha512.10000.0298C1C613A451C46FBC95BB2AC7A41BCEC1C61512EF785BD81E3B65DFF9D57ED4ADF8906C3EF33C22C
06FBDD366E1C118FC41110BD646A4D49EF86EFD0573BF.E14A45900096D773BE99BEA9AB8D4FA81431458952798B997D4FC9E0850426
F6798979371B8EBD331DB33AE8FEAE25E6773156D42F21B884DBA405546782B3BD
password_pbkdf2 trainee
grub.pbkdf2.sha512.10000.161D4183DC832357403296ED05961FCF494AED9E20DC21C84EA89085EB9EF5AAE4C7D4A276AA5CC21F9
C224B2ECA010B915B4830E9648A7398EB4A91E7E3D252.8277512B849FF727FDD0716D1D4CDC6B92E53918F665282E02133AAD1046EB
10273A2BC70D76558FFC34A0C0C8BE5132E4C4C02C7C9C1A567BD5365D77350FCF
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
password trainee trainee
EOF
```

Ouvrez maintenant le fichier **/boot/grub2/grub.cfg** et copier le premier **menuentry** de la section **/etc/grub.d/10_linux** :

```
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --
class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-b35de665-5ec8-4226-
a533-58a1b567ac91' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
    else
        search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
    fi
    linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8
    initrd16 /initramfs-3.10.0-229.4.2.el7.x86_64.img
}
```

Collez maintenant ce **menuentry** dans le fichier **/etc/grub.d/40_custom** :

[/etc/grub.d/40_custom](#)

```
#!/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.
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu -
--class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-b35de665-5ec8-4226-
a533-58a1b567ac91' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
    else
        search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
    fi
    linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8
    initrd16 /initramfs-3.10.0-229.4.2.el7.x86_64.img
}
```

Modifier le debut du menuentry ainsi :

```
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core) pour TRAINEE' --class centos --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-
```

```
b35de665-5ec8-4226-a533-58alb567ac91' --users trainee {
```

Sous RHEL/CentOS, lancez la commande **grub2-mkconfig -o /boot/grub2/grub.cfg** :

```
[root@centos7 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-229.4.2.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-a2feb9eb09b1488da0f23b99a66350f8
Found initrd image: /boot/initramfs-0-rescue-a2feb9eb09b1488da0f23b99a66350f8.img
done
```

En examinant le fichier **/boot/grub2/grub.cfg** on doit pouvoir constater la présence de la section **/etc/grub.d/40_custom** :

```
...
### 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.
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core) pour TRAINEE' --class centos --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-
b35de665-5ec8-4226-a533-58alb567ac91' --users trainee {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
    else
```

```
search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
fi
linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8
initrd16 /initramfs-3.10.0-229.4.2.el7.x86_64.img
}
### END /etc/grub.d/40_custom ###
...
```



A faire : Redémarrez votre VM et choisissez l'entrée de GRUB 2 issue du fichier **/etc/grub.d/40_custom** puis constatez que GRUB 2 demande un nom d'utilisateur ainsi qu'un mot de passe quand vous voulez éditer une entrée de GRUB 2.

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,
- **Ctrl-X** : démarrer avec la configuration modifiée,
- **echap** : abandonner les modifications et retourner à l'interface menu de GRUB 2.



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

Chargeurs de Démarrages Alternatifs

Systemd-boot

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

U-boot

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

Le Projet Syslinux

SYSLINUX

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

EXTLINUX

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

ISOLINUX

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

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

PXELINUX

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

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

Isodhpfx

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

Initramfs sous RHEL/CentOS 6

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

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

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

Examiner l'image existante

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

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

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

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

Installez maintenant le paquet **tree** :

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

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

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

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

```
|  | sed  
|  | setfont  
|  | sh -> dash  
|  | sleep  
|  | umount  
|  | uname  
|  |  
|  | cmdline  
|--Plus--
```

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

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



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

Le script init

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

```
[root@centos6 initrd]# more init  
#!/bin/sh  
#  
# Licensed under the GPLv2  
#  
# Copyright 2008-2009, Red Hat, Inc.  
# Harald Hoyer <harald@redhat.com>
```

```
# Jeremy Katz <katzj@redhat.com>

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

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



A faire : Passez en revue le contenu du script.

La commande dracut

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

```
[root@centos6 initrd]# dracut --help
```

```
Usage: /sbin/dracut [OPTION]... <initramfs> <kernel-version>
```

```
Creates initial ramdisk images for preloading modules
```

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

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

Le fichier de configuration de dracut est **/etc/dracut.conf**. Editez ce fichier pour spécifier des modules noyau supplémentaires à inclure dans le fichier image générée :

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

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

# Dracut modules to omit
#omit_dracutmodules+="

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

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

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

# build initrd only to boot current hardware
```

```
#hostonly="yes"
#

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

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

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

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

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

Déplacez votre fichier usbinitramfs au répertoire **/boot** :

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

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

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



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

Initramfs sous RHEL/CentOS 7

Comme sous RHEL/CentOS 6, le fichier Initramfs *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@centos7 ~]# cp /boot/initramfs-3.10.0-229.4.2.el7.x86_64.img /tmp/custom.gz  
[root@centos7 ~]# gunzip /tmp/custom.gz
```

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

```
[root@centos7 ~]# cd /tmp  
[root@centos7 tmp]# mkdir initrd  
[root@centos7 tmp]# cd initrd  
[root@centos7 initrd]# cpio -cid -I ../custom  
71439 blocks
```

Installez maintenant le paquet **tree** :

```
[root@centos7 initrd]# yum install tree
```

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

```
[root@centos7 initrd]# tree | more
```

```
.
├── bin -> usr/bin
├── dev
│   ├── console
│   ├── kmsg
│   └── null
├── etc
│   ├── cmdline.d
│   ├── conf.d
│   │   └── systemd.conf
│   ├── dhclient.conf
│   ├── fstab.empty
│   ├── group
│   ├── hostname
│   ├── initrd-release
│   ├── ld.so.cache
│   ├── ld.so.conf
│   ├── ld.so.conf.d
│   │   ├── dyninst-x86_64.conf
│   │   ├── kernel-3.10.0-123.el7.x86_64.conf
│   │   ├── kernel-3.10.0-229.4.2.el7.x86_64.conf
│   │   ├── libiscsi-x86_64.conf
│   │   ├── mariadb-x86_64.conf
│   │   └── tracker-x86_64.conf
│   ├── libnl
│   │   └── classid
│   ├── locale.conf
│   ├── machine-id
│   └── modprobe.d
--More--
```

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

```
[root@centos7 initrd]# ls -l
```

```
total 8
lrwxrwxrwx. 1 root root 7 Jun 10 10:09 bin -> usr/bin
drwxr-xr-x. 2 root root 42 Jun 10 10:09 dev
drwxr-xr-x. 11 root root 4096 Jun 10 10:09 etc
lrwxrwxrwx. 1 root root 23 Jun 10 10:09 init -> usr/lib/systemd/systemd
lrwxrwxrwx. 1 root root 7 Jun 10 10:09 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Jun 10 10:09 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Jun 10 10:09 proc
drwxr-xr-x. 2 root root 6 Jun 10 10:09 root
drwxr-xr-x. 2 root root 6 Jun 10 10:09 run
lrwxrwxrwx. 1 root root 8 Jun 10 10:09 sbin -> usr/sbin
-rwxr-xr-x. 1 root root 3041 Jun 10 10:09 shutdown
drwxr-xr-x. 2 root root 6 Jun 10 10:09 sys
drwxr-xr-x. 2 root root 6 Jun 10 10:09 sysroot
drwxr-xr-x. 2 root root 6 Jun 10 10:09 tmp
drwxr-xr-x. 7 root root 61 Jun 10 10:09 usr
drwxr-xr-x. 3 root root 47 Jun 10 10:09 var
```

Le script init

Le script **init** est lancé lors du chargement de l'image. RHEL/CentOS 7 utilise le système de démarrage **systemd**. Pour cette raison le script init sous RHEL/CentOS 7 est un lien symbolique vers **/usr/lib/systemd/systemd** :

```
lrwxrwxrwx. 1 root root 23 Jun 10 10:09 init -> usr/lib/systemd/systemd
```

La commande dracut sous RHEL/CentOS

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

```
[root@centos7 initrd]# dracut --help
Usage: /sbin/dracut [OPTION]... [<initramfs> [<kernel-version>]]
```

Version: 033-241.el7_1.1

Creates initial ramdisk images for preloading modules

```
--kver [VERSION]      Set kernel version to [VERSION].
-f, --force           Overwrite existing initramfs file.
-a, --add [LIST]      Add a space-separated list of dracut modules.
--rebuild            Append arguments to those of existing image and rebuild
-m, --modules [LIST] Specify a space-separated list of dracut modules to
                    call when building the initramfs. Modules are located
                    in /usr/lib/dracut/modules.d.
-o, --omit [LIST]     Omit a space-separated list of dracut modules.
--force-add [LIST]    Force to add a space-separated list of dracut modules
                    to the default set of modules, when -H is specified.
-d, --drivers [LIST]  Specify a space-separated list of kernel modules to
                    exclusively include in the initramfs.
--add-drivers [LIST]  Specify a space-separated list of kernel
                    modules to add to the initramfs.
--omit-drivers [LIST] Specify a space-separated list of kernel
                    modules not to add to the initramfs.
--filesystems [LIST] Specify a space-separated list of kernel filesystem
                    modules to exclusively include in the generic
                    initramfs.
-k, --kmoddir [DIR]   Specify the directory, where to look for kernel
                    modules
--fwdir [DIR]         Specify additional directories, where to look for
                    firmwares, separated by :
--kernel-only        Only install kernel drivers and firmware files
--no-kernel          Do not install kernel drivers and firmware files
--print-cmdline       Print the kernel command line for the given disk layout
--early-microcode     Combine early microcode with ramdisk
--no-early-microcode  Do not combine early microcode with ramdisk
--kernel-cmdline [PARAMETERS] Specify default kernel command line parameters
--strip              Strip binaries in the initramfs
```

```
--nostrip      Do not strip binaries in the initramfs
--prelink      Prelink binaries in the initramfs
--noprelink    Do not prelink binaries in the initramfs
--hardlink     Hardlink files in the initramfs
--nohardlink   Do not hardlink files in the initramfs
--prefix [DIR] Prefix initramfs files with [DIR]
--noprefix     Do not prefix initramfs files
--mdadmconf    Include local /etc/mdadm.conf
--nomdadmconf  Do not include local /etc/mdadm.conf
--lvmconf      Include local /etc/lvm/lvm.conf
--nolvmconf    Do not include local /etc/lvm/lvm.conf
--fscks [LIST] Add a space-separated list of fsck helpers.
--nofscks      Inhibit installation of any fsck helpers.
--ro-mnt       Mount / and /usr read-only by default.
-h, --help     This message
--debug        Output debug information of the build process
--profile      Output profile information of the build process
-L, --stdlog [0-6] Specify logging level (to standard error)
                  0 - suppress any messages
                  1 - only fatal errors
                  2 - all errors
                  3 - warnings
                  4 - info
                  5 - debug info (here starts lots of output)
                  6 - trace info (and even more)
-v, --verbose  Increase verbosity level
-q, --quiet    Decrease verbosity level
-c, --conf [FILE] Specify configuration file to use.
                  Default: /etc/dracut.conf
--confdir [DIR] Specify configuration directory to use *.conf files
                  from. Default: /etc/dracut.conf.d
--tmpdir [DIR] Temporary directory to be used instead of default
                  /var/tmp.
-l, --local    Local mode. Use modules from the current working
```

directory instead of the system-wide installed in
/usr/lib/dracut/modules.d.

Useful when running dracut from a git checkout.

-H, --hostonly Host-Only mode: Install only what is needed for
booting the local host instead of a generic host.

-N, --no-hostonly Disables Host-Only mode

--hostonly-cmdline Store kernel command line arguments needed
in the initramfs

--no-hostonly-cmdline Do not store kernel command line arguments needed
in the initramfs

--persistent-policy [POLICY]
Use [POLICY] to address disks and partitions.
POLICY can be any directory name found in /dev/disk.
E.g. "by-uuid", "by-label"

--fstab Use /etc/fstab to determine the root device.

--add-fstab [FILE] Add file to the initramfs fstab

--mount "[DEV] [MP] [FSTYPE] [FSOPTS]"
Mount device [DEV] on mountpoint [MP] with filesystem
[FSTYPE] and options [FSOPTS] in the initramfs

--add-device "[DEV]" Bring up [DEV] in initramfs

-i, --include [SOURCE] [TARGET]
Include the files in the SOURCE directory into the
Target directory in the final initramfs.
If SOURCE is a file, it will be installed to TARGET
in the final initramfs.

-I, --install [LIST] Install the space separated list of files into the
initramfs.

--gzip Compress the generated initramfs using gzip.
This will be done by default, unless another
compression option or --no-compress is passed.

--bzip2 Compress the generated initramfs using bzip2.
Make sure your kernel has bzip2 decompression support
compiled in, otherwise you will not be able to boot.

--lzma Compress the generated initramfs using lzma.

```
Make sure your kernel has lzma support compiled in,
otherwise you will not be able to boot.
--xz      Compress the generated initramfs using xz.
          Make sure that your kernel has xz support compiled
          in, otherwise you will not be able to boot.
--lzo     Compress the generated initramfs using lzop.
          Make sure that your kernel has lzo support compiled
          in, otherwise you will not be able to boot.
--lz4     Compress the generated initramfs using lz4.
          Make sure that your kernel has lz4 support compiled
          in, otherwise you will not be able to boot.
--compress [COMPRESSION] Compress the generated initramfs with the
                        passed compression program. Make sure your kernel
                        knows how to decompress the generated initramfs,
                        otherwise you will not be able to boot.
--no-compress Do not compress the generated initramfs. This will
              override any other compression options.
--list-modules List all available dracut modules.
-M, --show-modules Print included module's name to standard output during
                  build.
--keep         Keep the temporary initramfs for debugging purposes
--printsize    Print out the module install size
--sshkey [SSHKEY] Add ssh key to initramfs (use with ssh-client module)
--logfile [FILE]  Logfile to use (overrides configuration setting)
```

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

For example:

```
# dracut --add-drivers "module1 module2" ...
```

Le fichier de configuration de dracut est **/etc/dracut.conf**. Editez ce fichier pour spécifier des modules noyau supplémentaires à inclure dans le fichier image générée :

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

# Sample dracut config file

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

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

# dracut modules to omit
#omit_dracutmodules+="

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

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

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

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

# install local /etc/mdadm.conf
#mdadmconf="no"

# install local /etc/lvm/lvm.conf
```

```
#lvmconf="no"

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

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

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

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

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

```
[root@centos7 initrd]# dracut -v usbinitramfs
Executing: /sbin/dracut -v usbinitramfs
*** Including module: bash ***
*** Including module: nss-softoken ***
*** Including module: il8n ***
*** Including module: network ***
*** Including module: ifcfg ***
*** Including module: drm ***
*** Including module: plymouth ***
*** Including module: kernel-modules ***
*** Including module: fcoe ***
*** Including module: fcoe-uefi ***
*** Including module: resume ***
*** Including module: rootfs-block ***
```

```
*** Including module: terminfo ***
*** Including module: udev-rules ***
Skipping udev rule: 91-permissions.rules
*** Including module: biosdevname ***
*** Including module: systemd ***
*** Including module: usrmount ***
*** Including module: base ***
*** Including module: fs-lib ***
*** Including module: shutdown ***
*** Including module: uefi-lib ***
*** Including modules done ***
*** Installing kernel module dependencies and firmware ***
*** Installing kernel module dependencies and firmware done ***
*** Resolving executable dependencies ***
*** Resolving executable dependencies done***
*** Hardlinking files ***
*** Hardlinking files done ***
*** Stripping files ***
*** Stripping files done ***
*** Generating early-microcode cpio image ***
*** Constructing GenuineIntel.bin *****
*** Store current command line parameters ***
*** Creating image file ***
*** Creating image file done ***
```

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

```
[root@centos7 initrd]# ls -l /tmp/initrd/usbinitramfs
-rw-----. 1 root root 16373488 Jun 10 10:23 /tmp/initrd/usbinitramfs
```

Déplacez votre fichier usbinitramfs au répertoire /boot :

```
[root@centos7 initrd]# mv usbinitramfs /boot
```

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

```
...
### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-229.4.2.el7.x86_64) 7 (Core)' --class centos --class gnu-linux --class gnu --
class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-b35de665-5ec8-4226-
a533-58a1b567ac91' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
    else
        search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
    fi
    linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet LANG=en_US.UTF-8
    initrd16 /initramfs-3.10.0-229.4.2.el7.x86_64.img
}
...
```

Créez maintenant le fichier **/etc/grub.d/09_usbredhat**, collez la section précédemment copiée à la place indiquée entre les lignes **cat « EOF** et **EOF** puis modifiez la dernière ligne **initrd16** :

```
#!/bin/sh -e
cat << EOF
menuentry 'CentOS Linux, with Linux 3.10.0-229.4.2.el7.x86_64 et usbinitramfs' --class centos --class gnu-linux -
-class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.4.2.el7.x86_64-advanced-
b35de665-5ec8-4226-a533-58a1b567ac91' {
    load_video
```

```
set gfxpayload=keep
insmod gzio
insmod part_msdos
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
else
    search --no-floppy --fs-uuid --set=root e8d3bd48-1386-411c-9675-41c3f8f1a309
fi
linux16 /vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-a533-58a1b567ac91 ro
vconsole.keymap=fr crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet
initrd16 /usbinitramfs
}
EOF
```

Rendez ce fichier exécutable :

```
[root@centos7 initrd]# chmod +x /etc/grub.d/09_usbredhat
```

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@centos7 initrd]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
```

```
Found initrd image: /boot/initramfs-3.10.0-229.4.2.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-a2feb9eb09b1488da0f23b99a66350f8
Found initrd image: /boot/initramfs-0-rescue-a2feb9eb09b1488da0f23b99a66350f8.img
done
```

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

```
...
### BEGIN /etc/grub.d/09_usbredhat ###
menuentry 'CentOS Linux, with Linux 3.10.0-229.4.2.el7.x86_64 et usbinitramfs' --class centos --class gnu-linux -
-class gnu --class os --unrestricted 'gnulinux-3.10.0-229.4.2.el7.x86_64-advanced-b35de665-5ec8-4226-
a533-58a1b567ac91' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    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 --hint='hd0,msdos1' e8d3bd48-1386-411c-9675-41c3f8f1a309
    else
        ...
```



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

Processus de Démarrage du Noyau Linux

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

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

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

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

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

Processus Init

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

- le montage de /proc et de /sys,
 - configuration des paramètres du noyau présents dans **/etc/sysctl.conf**,
 - l'activation de SELinux,
 - la mise à l'heure du système,
-

- la définition des consoles textes,
- la définition du nom de la machine,
- la détection des périphériques USB,
- la mise en place du support RAID et LVM,
- l'activation des quotas de disque,
- le montage 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

La commande `/bin/dmesg`

Cette commande retourne les messages du noyau (**Kernel Ring Buffer**) stockés dans le fichier `/var/log/dmesg` lors du dernier démarrage du système :

```
[root@centos7 ~]# dmesg | more
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.10.0-229.4.2.el7.x86_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.2
20140120 (Red Hat 4.8.2-16) (GCC) ) #1 SMP
Wed May 13 10:06:09 UTC 2015
[    0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.10.0-229.4.2.el7.x86_64 root=UUID=b35de665-5ec8-4226-
a533-58a1b567ac91 ro vconsole.keymap=fr crashk
ernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
```

```
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000005ffefffff] usable
[ 0.000000] BIOS-e820: [mem 0x000000005fff0000-0x000000005fffffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] SMBIOS 2.5 present.
[ 0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[ 0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[ 0.000000] No AGP bridge found
[ 0.000000] e820: last_pfn = 0x5ffff max_arch_pfn = 0x400000000
[ 0.000000] MTRR default type: uncachable
[ 0.000000] MTRR variable ranges disabled:
[ 0.000000] x86 PAT enabled: cpu 0, old 0x7040600070406, new 0x7010600070106
[ 0.000000] CPU MTRRs all blank - virtualized system.
[ 0.000000] found SMP MP-table at [mem 0x0009fff0-0x0009ffff] mapped at [ffff88000009fff0]
[ 0.000000] Base memory trampoline at [ffff880000099000] 99000 size 24576
[ 0.000000] init_memory_mapping: [mem 0x00000000-0x000fffff]
[ 0.000000] [mem 0x00000000-0x000fffff] page 4k
--More--
```

Options de la Commande

Les option de cette commande sont :

```
[root@centos7 ~]# dmesg --help
```

Usage:

dmesg [options]

Options:

-C, --clear	clear the kernel ring buffer
-c, --read-clear	read and clear all messages
-D, --console-off	disable printing messages to console

```
-d, --show-delta      show time delta between printed messages
-e, --reltime         show local time and time delta in readable format
-E, --console-on      enable printing messages to console
-F, --file <file>     use the file instead of the kernel log buffer
-f, --facility <list> restrict output to defined facilities
-H, --human           human readable output
-k, --kernel          display kernel messages
-L, --color           colorize messages
-l, --level <list>    restrict output to defined levels
-n, --console-level <level> set level of messages printed to console
-P, --nopager         do not pipe output into a pager
-r, --raw            print the raw message buffer
-S, --syslog          force to use syslog(2) rather than /dev/kmsg
-s, --buffer-size <size> buffer size to query the kernel ring buffer
-T, --ctime           show human readable timestamp (could be
                     inaccurate if you have used SUSPEND/RESUME)
-t, --notime         don't print messages timestamp
-u, --userspace       display userspace messages
-w, --follow          wait for new messages
-x, --decode          decode facility and level to readable string

-h, --help           display this help and exit
-V, --version         output version information and exit
```

Supported log facilities:

```
kern - kernel messages
user - random user-level messages
mail - mail system
daemon - system daemons
auth - security/authorization messages
syslog - messages generated internally by syslogd
lpr - line printer subsystem
news - network news subsystem
```

Supported log levels (priorities):

- emerg - system is unusable
- alert - action must be taken immediately
- crit - critical conditions
- err - error conditions
- warn - warning conditions
- notice - normal but significant condition
- info - informational
- debug - debug-level messages

For more details see `dmesg(q)`.

<html>

Copyright © 2004-2016 Hugh Norris.

Ce(tte) oeuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.

</html>
