Niveau : Admin Junior	Numéro de la Leçon	Dernière Modification
2/4	<progrecss 10/12 style=inline />	2020/01/30 03:28

Gestion du Démarrage et de l'Arrêt du Système

Détail du démarrage

BIOS

Au démarrage de la machine, le premier programme exécuté est 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 **gestionnaire d'amorçage** du système d'exploitation.

Gestionnaire d'amorçage

Le gestionnaire d'amorçage est en règle générale placé dans le MBR du disque. Le format du MBR est le suivant :

- 446 octets pour le gestionnaire d'amorçage,
- 64 octest pour la table de partitions, soit 16 octets par partition décrite,
- 2 octets ayant une valeur fixe en hexadécimale de **AA55**.

Il est à noter que le gestionnaire d'amorçage peut également être installé dans un PBR.

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
- ...

Grub

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

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

où **périphérique** est le nom du périphérique ou grub doit s'installer dans le MBR.

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

```
A> fdisk /mbr [Entrée]
```

</note>

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

```
opensuse:~ # cat /boot/grub/menu.lst
# Modified by YaST2. Last modification on Tue May 17 15:03:57 CEST 2011
# THIS FILE WILL BE PARTIALLY OVERWRITTEN by perl-Bootloader
# Configure custom boot parameters for updated kernels in /etc/sysconfig/bootloader

default 0
timeout 8
##YaST - generic_mbr
gfxmenu (hd0,0)/message
##YaST - activate
```

```
###Don't change this comment - YaST2 identifier: Original name: linux###
title Desktop -- openSUSE 11.4 - 2.6.37.6-0.5
    root (hd0,0)
    kernel /vmlinuz-2.6.37.6-0.5-desktop root=/dev/sda2 resume=/dev/sda3 splash=silent quiet showopts vga=0x314
    initrd /initrd-2.6.37.6-0.5-desktop

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- openSUSE 11.4 - 2.6.37.6-0.5
    root (hd0,0)
    kernel /vmlinuz-2.6.37.6-0.5-desktop root=/dev/sda2 showopts apm=off noresume nosmp maxcpus=0 edd=off
powersaved=off nohz=off highres=off processor.max_cstate=1 nomodeset x11failsafe vga=0x314
    initrd /initrd-2.6.37.6-0.5-desktop
```

Initramfs sous SLES/openSUSE

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 existante, il convient d'abord de la copier vers /tmp et de la décompresser :

```
opensuse:~ # cp /boot/initrd /tmp/custom.gz
opensuse:~ # gunzip /tmp/custom.gz
```

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

```
opensuse:~ # cd /tmp
```

```
opensuse:/tmp # mkdir initrd
opensuse:/tmp # cd initrd/
opensuse:/tmp/initrd # cpio -cid -I ../custom
44146 blocks
```

Installez maintenant le paquet **tree** :

```
opensuse:/tmp/initrd # zypper install tree
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  tree

1 new package to install.
Overall download size: 33.0 KiB. After the operation, additional 74.0 KiB will
be used.
Continue? [y/n/?] (y): y
Retrieving package tree-1.5.3-7.1.i586 (1/1), 33.0 KiB (74.0 KiB unpacked)
Retrieving: tree-1.5.3-7.1.i586.rpm [done (86.2 KiB/s)]
Installing: tree-1.5.3-7.1 [done]
```

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

```
opensuse:/tmp/initrd # tree | more
.
├── bin
│   ├── awk -> /etc/alternatives/awk
│   ├── bash
│   ├── cat
│   ├── chmod
│   ├── cp
│   ├── date
```

```
│    ├── grep
│    ├── ipconfig
│    ├── ipconfig.sh
│    ├── linuxrc
│    ├── ln
│    ├── logger
│    ├── ls
│    ├── mkdir
│    ├── mknod
│    ├── mount
│    ├── mv
│    ├── on_ac_power
│    ├── rm
│    ├── run-init
│    ├── sed
--More--
```

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

```
[root@centos initrd]# ls
opensuse:/tmp/initrd # ls
bin   bootsplash  dev  init  mkinitrd.config  root        sbin  tmp  var
boot  config      etc  lib   proc             run_all.sh  sys   usr
```

<note> Utilisez le manuel de la commande **cpio** pour comprendre les options utilisées. </note>

Le script init

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

```
opensuse:/tmp/initrd # more init
#!/bin/bash
```

```
##################################################
# add_module_param $kernelmodule $value
# remembers parameters for the given kernel module
# e.g. add_module_param rtl8193too debug=1
##################################################
add_module_param() {
    echo "options $1 $2" >> /etc/modprobe.d/options.conf
}

##################################################
# load_modules
# loads all kernelmodules that belong to the
# current init module
# this is also done automatically when the
# init module is done
# e.g. load_modules
##################################################
load_modules() {
    local module
    for module in $(eval echo $modules)
    do
--More--(32%)
```

<note> Passez en revue le contenu du script. </note>

La commande mkinitrd

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

```
opensuse:/boot # mkinitrd -h
```

```
MKINITRD
       Create initial ramdisk images that contain all kernel modules needed in
       the early boot process, before the root file system becomes  available.
       This  usually  includes  SCSI and/or RAID modules, a file system module
       for the root file system, or a  network  interface  driver  module  for
       dhcp.

       mkinitrd [options]



       -f feature list
              Features  to  be enabled when generating initrd.  Available fea-
              tures are: iscsi, md, multipath, lvm, lvm2, ifup

       -k kernel list
              List of kernel  images  for  which  initrd  files  are  created.
              Defaults to all kernels found in /boot.

       -i initrd list
              List  of file names for the initrd; position have match to "ker-
              nel list". Defaults to all kernels found in /boot.

       -l lib_dir
              mkinitrd directory. Defaults to /lib/mkinitrd.

       -b boot_dir
              Boot directory. Defaults to /boot.

       -M map System.map file to use.

       -A     Create a so called "monster initrd" which includes all  features
              and modules possible.

       -B     Do not update bootloader configuration.
```

```
       -v     Verbose mode.

       -R     Print release (version).

       -L     Disable logging.

       -h     This help screen.

       -m module list
              Modules  to  include  in  initrd. Defaults to the INITRD_MODULES
              variable in /etc/sysconfig/kernel

       -u DomU module list
              Modules to include in initrd. Defaults to  the  DOMU_INITRD_MOD-
              ULES variable in /etc/sysconfig/kernel.

       -d root_device
              Root  device.  Defaults  to  the device from which / is mounted.
              Overrides the rootdev enviroment variable if set.

       -j device
              Journal device

       -S     Don't include all libata drivers.

       -D interface
              Run dhcp on the specified interface.

       -I interface
              Configure the specified interface statically.

       -a acpi_dsdt
              Attach compiled ACPI DSDT (Differentiated System Description Ta-
              ble)  to initrd. This replaces the DSDT of the BIOS. Defaults to
```

```
              the ACPI_DSDT variable in /etc/sysconfig/kernel.

       -s size
              Add splash animation and bootscreen to initrd.

       -V script
              Vendor specific script to run in linuxrc (deprecated).
```

Le fichier de configuration de mkinitrd est **/etc/sysconfig/kernel**. Éditez la directive **INITRD_MODULES** de ce fichier pour spécifier des modules noyau supplémentaires à inclure dans le fichier image générée :

```
...
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="ahci ata_piix ata_generic thermal processor fan ehci-hcd ohci-hcd usb-storage scsi_mod sd_mod"

## Type:        string(yes)
...
```

Naviguez au répertoire **/boot** et exécutez ensuite la commande suivante :

```
# mkinitrd -v [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
opensuse:/boot # mkinitrd -v

Kernel image:   /boot/vmlinuz-2.6.37.6-0.5-desktop
Initrd image:   /boot/initrd-2.6.37.6-0.5-desktop
Root device:    /dev/disk/by-id/ata-VBOX_HARDDISK_VBb97f4bdd-e14419fa-part2 (/dev/sda2) (mounted on / as ext3)
Resume device:  /dev/disk/by-id/ata-VBOX_HARDDISK_VBb97f4bdd-e14419fa-part3 (/dev/sda3)
[BLOCK] /dev/sda -> ata_piix
[BLOCK] /dev/sda -> sd_mod
[MODULES]   02-start.sh: ahci ata_piix ata_generic thermal processor fan ehci-hcd ohci-hcd usb-storage scsi_mod
```

```
sd_mod
[MODULES]   02-start.sh:
[MODULES]   03-rtc.sh: rtc_cmos
[MODULES]   03-storage.sh:
[MODULES]   11-block.sh: ata_piix sd_mod
[MODULES]   11-usb.sh: usbcore
[MODULES]   11-usb.sh: ohci_hcd
[MODULES]   11-usb.sh: uhci-hcd
[MODULES]   11-usb.sh: ehci_hcd
[MODULES]   11-usb.sh: usbhid
[MODULES]   'modinfo -k "2.6.37.6-0.5-desktop" -F supported'  returned with an error.
Kernel Modules: ata_generic thermal_sys thermal processor fan
[MOUNT] Root:   /dev/disk/by-id/ata-VBOX_HARDDISK_VBb97f4bdd-e14419fa-part2
Features:       block usb resume.userspace resume.kernel
Bootsplash: openSUSE (800x600)
Shared libs:    /lib/udev/bluetooth.sh is a script
/lib/udev/bluetooth_serial is a script
/lib/udev/collect_lvm is a script
/lib/udev/findkeyboards is a script
/lib/udev/idedma.sh is a script
/lib/udev/iwlwifi-led.sh is a script
/lib/udev/keyboard-force-release.sh is a script
/lib/udev/kpartx_id is a script
/lib/udev/udev-add-printer is a script
/lib/udev/usb_modeswitch is a script
/lib/udev/write_cd_rules is a script
/lib/udev/write_net_rules is a script
/lib/mkinitrd/bin/ipconfig.sh is a script
/sbin/ifup is a script
/lib/mkinitrd/bin/ipconfig.sh is a script
/lib/mkinitrd/bin/linuxrc is a script
/usr/bin/on_ac_power is a script
lib/ld-2.11.3.so lib/libacl.so.1.1.0 lib/libattr.so.1.1.0 lib/libblkid.so.1.1.0 lib/libc-2.11.3.so
lib/libcap.so.2.19 lib/libcom_err.so.2.1 lib/libcrypto.so.1.0.0 lib/libdevmapper.so.1.02 lib/libdl-2.11.3.so
```

```
lib/libe2p.so.2.3 lib/libext2fs.so.2.4 lib/libgcrypt.so.11.6.0 lib/libgio-2.0.so.0.2800.0
lib/libglib-2.0.so.0.2800.0 lib/libgmodule-2.0.so.0.2800.0 lib/libgobject-2.0.so.0.2800.0 lib/libgpg-
error.so.0.7.0 lib/libgthread-2.0.so.0.2800.0 lib/libkeyutils-1.3.so lib/liblzo2.so.2.0.0 lib/libm-2.11.3.so
lib/libmount.so.1.1.0 lib/libncurses.so.5.7 lib/libpcre.so.0.0.1 lib/libpthread-2.11.3.so lib/libreadline.so.6.1
lib/libresolv-2.11.3.so lib/librt-2.11.3.so lib/libselinux.so.1 lib/libsepol.so.1 lib/libssl.so.1.0.0
lib/libudev.so.0.10.0 lib/libutil-2.11.3.so lib/libuuid.so.1.3.0 lib/libz.so.1.2.5 usr/lib/libatasmart.so.4.0.3
usr/lib/libcups.so.2 usr/lib/libdal-0.3.so.0.0.0 usr/lib/libdirect-1.4.so.5.0.0 usr/lib/libdirectfb-1.4.so.5.0.0
usr/lib/libfusion-1.4.so.5.0.0 usr/lib/libgdk_pixbuf-2.0.so.0.2200.1 usr/lib/libgnutls.so.26.14.12
usr/lib/libgpod.so.4.3.1 usr/lib/libgssapi_krb5.so.2.2 usr/lib/libimobiledevice.so.1.0.4
usr/lib/libk5crypto.so.3.1 usr/lib/libkrb5.so.3.3 usr/lib/libkrb5support.so.0.1 usr/lib/libmtp.so.8.3.6
usr/lib/libparted.so.0.0.1 usr/lib/libplist.so.1.1.3 usr/lib/libreiserfs-0.3.so.0.0.0
usr/lib/libsgutils2.so.2.0.0 usr/lib/libsplashy.so.1.0.0 usr/lib/libsplashycnf.so.1.0.0
usr/lib/libsqlite3.so.0.8.6 usr/lib/libtasn1.so.3.1.5 usr/lib/libusb-0.1.so.4.4.4 usr/lib/libusb-1.0.so.0.0.0
usr/lib/libusbmuxd.so.1.0.7 usr/lib/libxml2.so.2.7.8 lib/libnss_dns-2.11.3.so lib/libnss_dns.so.2
lib/libnss_files-2.11.3.so lib/libnss_files.so.2 lib/libgcc_s.so.1
44147 blocks
```

Notez la présence de votre nouvelle image :

```
opensuse:/boot # ls -l | grep initrd
lrwxrwxrwx 1 root root       27 Oct 19 14:13 initrd -> initrd-2.6.37.6-0.5-desktop
-rw-r--r-- 1 root root 10473839 Oct 19 14:13 initrd-2.6.37.6-0.5-desktop
```

<note> **Re-démarrez votre machine pour tester votre configuration.** </note>

Processus de Démarrage de Linux

Le processus de démarrage de 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** (*Interrupt Descriptor Table*) temporaire et **GDT** (*Global Descriptor Table*) 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

Niveaux d'exécution

Il existe 8 niveaux d'exécution ou **RUNLEVELS** sous SLES/openSUSE. 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 SLES/openSUSE, ils sont :

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

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

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

```
opensuse:~ # runlevel
N 5
```

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

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

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

Inittab

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

```
opensuse:~ # cat /etc/inittab
#
# /etc/inittab
#
# Copyright (c) 1996-2002 SuSE Linux AG, Nuernberg, Germany.  All rights reserved.
#
# Author: Florian La Roche, 1996
# Please send feedback to http://www.suse.de/feedback
#
# This is the main configuration file of /sbin/init, which
# is executed by the kernel on startup. It describes what
# scripts are used for the different run-levels.
#
# All scripts for runlevel changes are in /etc/init.d/.
#
# This file may be modified by SuSEconfig unless CHECK_INITTAB
# in /etc/sysconfig/suseconfig is set to "no"
#

# The default runlevel is defined here
```

```
id:5:initdefault:

# First script to be executed, if not booting in emergency (-b) mode
si::bootwait:/etc/init.d/boot

# /etc/init.d/rc takes care of runlevel handling
#
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do not use this for initdefault!)
#
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

# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# special keyboard request (Alt-UpArrow)
# look into the kbd-0.90 docs for this
kb::kbrequest:/bin/echo "Keyboard Request -- edit /etc/inittab to let this work."
```

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

# for ARGO UPS
sh:12345:powerfail:/sbin/shutdown -h now THE POWER IS FAILING

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field  MUST be the same as the last
# characters of the device (after "tty").
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
#
#S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102
#cons:12345:respawn:/sbin/smart_agetty -L 38400 console

#
#  Note: Do not use tty7 in runlevel 3, this virtual line
#  is occupied by the programm xdm.
#

#  This is for the package xdmsc, after installing and
#  and configuration you should remove the comment character
#  from the following line:
#7:3:respawn:+/etc/init.d/rx tty7
```

```
# modem getty.
# mo:235:respawn:/usr/sbin/mgetty -s 38400 modem

# fax getty (hylafax)
# mo:35:respawn:/usr/lib/fax/faxgetty /dev/modem

# vbox (voice box) getty
# I6:35:respawn:/usr/sbin/vboxgetty -d /dev/ttyI6
# I7:35:respawn:/usr/sbin/vboxgetty -d /dev/ttyI7

# end of /etc/inittab
```

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

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

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

Directive	Description
respawn	Le processus est relancé en cas d'arrêt de celui-ci
mingetty	Assure la gestion du terminal texte
once	Le processus n'est exécuté qu'une fois
wait	Le processus n'est exécuté qu'une fois. Init attend la fin du processus avant de passer à la ligne suivante
boot	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance
bootwait	Le processus est exécuté au démarrage de la machine. Le champ RUNLEVELS est sans importance. Init attend la fin du processus avant de passer à la ligne suivante
off	Revient à commenter la ligne
ondemand	La même chose que respawn mais la commande est exécuté dans un des 3 pseudo-niveaux d'exécution
initdefault	Définit le niveau d'exécution par défaut

Directive	Description
sysinit	La commande est exécutée au démarrage da 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 indique donc :

Ligne	Description
id:5:initdefault:	Le niveau d'exécution par défaut est 5
si::bootwait:/etc/init.d/boot	Le script /etc/rc.d/boot est lancé au démarrage de la machine
l0:0:wait:/etc/init.d/rc 0	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 0
l1:1:wait:/etc/init.d/rc 1	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 1
l2:2:wait:/etc/init.d/rc 2	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 2
l3:3:wait:/etc/init.d/rc 3	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 3
l4:4:wait:/etc/init.d/rc 4	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 4 (ligne en commentaire par défaut)
l5:5:wait:/etc/init.d/rc 5	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 5
l6:6:wait:/etc/init.d/rc 6	Le script /etc/init.d/rc est lancé pour le niveau d'exécution 6
ls:S:wait:/etc/init.d/rc S	Le script /etc/init.d/rc est lancé pour le niveau d'exécution S
~~:S:respawn:/sbin/sulogin	La commande /sbin/sulogin est lancée pour le niveau d'exécution S
kb::kbrequest:/bin/echo “Keyboard Request – edit /etc/inittab to let this work.”	La commande est exécutée suivant des séquences de touches saisies au clavier
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now	La commande **/sbin/shutdown -r -t 4 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 est lancé quand Init reçoit le signal SIGPWR
pn::powerfailnow:/etc/init.d/powerfail now	Le script /etc/init.d/powerfail est lancé avec l'option **now** quand Init reçoit un signal de coupure définitive d'électricité

Ligne	Description
po::powerokwait:/etc/init.d/powerfail stop	Le script /etc/init.d/powerfail est lancé avec l'option **stop** quand Init reçoit un signal de rétablissement du courant
sh:12345:powerfail:/sbin/shutdown -h now THE POWER IS FAILING	La commande /sbin/shutdown -h now est lancée quand Init reçoit un signal de coupure définitive d'électricité à partir d'un onduleur de type ARGO
1:2345:respawn:/sbin/mingetty –noclear tty1	Le terminal tty1 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F1]
2:2345:respawn:/sbin/mingetty tty2	Le terminal tty2 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F2]
3:2345:respawn:/sbin/mingetty tty3	Le terminal tty3 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F3]
4:2345:respawn:/sbin/mingetty tty4	Le terminal tty4 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F4]
5:2345:respawn:/sbin/mingetty tty5	Le terminal tty5 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F5]
6:2345:respawn:/sbin/mingetty tty6	Le terminal tty6 est initialisé par la commande /sbin/mingetty. Le terminal est disponible en appuyant sur les touches [CTRL] [ALT] [F6]

Scripts de Démarrage

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

```
opensuse:/etc # cd /etc/rc.d
opensuse:/etc/rc.d # ls
.depend.boot         boot.ipconfig      haveged           rc5.d
.depend.halt         boot.klog          inputattach       rc6.d
.depend.start        boot.ldconfig      jexec             rcS.d
.depend.stop         boot.loadmodules   joystick          reboot
SuSEfirewall2_init   boot.local         kbd               rpcbind
SuSEfirewall2_setup  boot.localfs       kexec             rpmconfigcheck
aaeventd             boot.localnet      lirc              rsyncd
acpid                boot.lvm           mdadmd            setserial
after.local          boot.md            microcode.ctl     single
```

```
alsasound            boot.multipath     multipathd        skeleton
atd                  boot.proc          network           skeleton.compat
auditd               boot.rootfsck      network-remotefs  smartd
autofs               boot.startpreload  nfs               smb
autoyast             boot.swap          nmb               smolt
avahi-daemon         boot.sysctl        nscd              smpppd
avahi-dnsconfd       boot.udev          ntp               splash
before.local         boot.udev_retry    openvpn           splash_early
bluez-coldplug       cifs               pm-profiler       sshd
boot                 cpufreq            postfix           stoppreload
boot.apparmor        cron               powerd            syslog
boot.cgroup          cups               powerfail         vboxadd
boot.cleanup         dbus               random            vboxadd-service
boot.clock           dnsmasq            raw               vboxadd-x11
boot.crypto          earlysyslog        rc                xdm
boot.crypto-early    earlyxdm           rc0.d             xfs
boot.cycle           fbset              rc1.d             xinetd
boot.d               gpm                rc2.d             ypbind
boot.device-mapper   halt               rc3.d
boot.dmraid          halt.local         rc4.d
```

<note important> Veuillez noter que sous openSUSE, le répertoire **/etc/rc.d** est un lien symbolique pointant vers **/etc/init.d**. </note>

Répertoire /etc/init.d

Le répertoire **init.d** contient, entre autre, :

- les scripts permettant de lancer les services du système,
- le script **rc** chargé de lancer les scripts identifiés par des liens symboliques dans les répertoires **rcX.d** selon le niveau d'exécution,
- le répertoire **boot.d** contenant des scripts lancé au démarrage par le script **boot**,
- le script **boot.local** qui joue le rôle d'autoexec.bat sous Windows™,
- le script **boot.setup** qui est exécuté lors d'un passage du niveau d'exécution **1** à n'importe quel autre niveau d'exécution. Ce script est notamment responsable de la définition du type de clavier ainsi que les consoles virtuels.

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 :

```
opensuse:/etc/rc.d # for rep in rc[345].d; do echo "dans $rep :"; ls $rep/S*; done
dans rc3.d :
rc3.d/S01SuSEfirewall2_init  rc3.d/S04splash_early
rc3.d/S01acpid               rc3.d/S07kbd
rc3.d/S01cpufreq             rc3.d/S08alsasound
rc3.d/S01dbus                rc3.d/S08avahi-daemon
rc3.d/S01earlysyslog         rc3.d/S08bluez-coldplug
rc3.d/S01fbset               rc3.d/S08network-remotefs
rc3.d/S01jexec               rc3.d/S08splash
rc3.d/S01random              rc3.d/S11cups
rc3.d/S01vboxadd             rc3.d/S11nscd
rc3.d/S01vboxadd-x11         rc3.d/S11postfix
rc3.d/S02network             rc3.d/S11smpppd
rc3.d/S02vboxadd-service     rc3.d/S12cron
rc3.d/S03syslog              rc3.d/S12smartd
rc3.d/S04auditd              rc3.d/S13SuSEfirewall2_setup
rc3.d/S04rpcbind             rc3.d/S13stoppreload
dans rc4.d :
rc4.d/S01jexec  rc4.d/S01vboxadd  rc4.d/S02vboxadd-service
dans rc5.d :
rc5.d/S01SuSEfirewall2_init  rc5.d/S07kbd
rc5.d/S01acpid               rc5.d/S08alsasound
rc5.d/S01cpufreq             rc5.d/S08avahi-daemon
rc5.d/S01dbus                rc5.d/S08bluez-coldplug
rc5.d/S01earlysyslog         rc5.d/S08network-remotefs
rc5.d/S01fbset               rc5.d/S08splash
rc5.d/S01jexec               rc5.d/S11cups
```

```
rc5.d/S01random              rc5.d/S11earlyxdm
rc5.d/S01vboxadd             rc5.d/S11nscd
rc5.d/S01vboxadd-x11         rc5.d/S11postfix
rc5.d/S02network             rc5.d/S11smpppd
rc5.d/S02vboxadd-service     rc5.d/S12cron
rc5.d/S03syslog              rc5.d/S12smartd
rc5.d/S04auditd              rc5.d/S12xdm
rc5.d/S04rpcbind             rc5.d/S13SuSEfirewall2_setup
rc5.d/S04splash_early        rc5.d/S13stoppreload
```

<note 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 exécuté 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. </note>

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

```
opensuse:/etc/rc.d # for rep in rc[345].d; do echo "dans $rep :"; ls $rep/K*; done
dans rc3.d :
rc3.d/K01SuSEfirewall2_setup  rc3.d/K02acpid
rc3.d/K01auditd               rc3.d/K02alsasound
rc3.d/K01bluez-coldplug       rc3.d/K02avahi-daemon
rc3.d/K01cpufreq              rc3.d/K02cups
rc3.d/K01cron                 rc3.d/K02fbset
rc3.d/K01jexec                rc3.d/K02kbd
rc3.d/K01nscd                 rc3.d/K02postfix
rc3.d/K01random               rc3.d/K02vboxadd
rc3.d/K01smartd               rc3.d/K03dbus
rc3.d/K01smpppd               rc3.d/K03network-remotefs
rc3.d/K01splash               rc3.d/K06rpcbind
rc3.d/K01splash_early         rc3.d/K07syslog
rc3.d/K01stoppreload          rc3.d/K08earlysyslog
rc3.d/K01vboxadd-service      rc3.d/K08network
rc3.d/K01vboxadd-x11          rc3.d/K09SuSEfirewall2_init
```

```
dans rc4.d :
rc4.d/K01jexec  rc4.d/K01vboxadd-service  rc4.d/K02vboxadd
dans rc5.d :
rc5.d/K01SuSEfirewall2_setup  rc5.d/K02acpid
rc5.d/K01auditd               rc5.d/K02alsasound
rc5.d/K01bluez-coldplug       rc5.d/K02avahi-daemon
rc5.d/K01cpufreq              rc5.d/K02cups
rc5.d/K01cron                 rc5.d/K02earlyxdm
rc5.d/K01jexec                rc5.d/K02fbset
rc5.d/K01nscd                 rc5.d/K02kbd
rc5.d/K01random               rc5.d/K02postfix
rc5.d/K01smartd               rc5.d/K02vboxadd
rc5.d/K01smpppd               rc5.d/K03dbus
rc5.d/K01splash               rc5.d/K03network-remotefs
rc5.d/K01splash_early         rc5.d/K06rpcbind
rc5.d/K01stoppreload          rc5.d/K07syslog
rc5.d/K01vboxadd-service      rc5.d/K08earlysyslog
rc5.d/K01vboxadd-x11          rc5.d/K08network
rc5.d/K01xdm                  rc5.d/K09SuSEfirewall2_init
```

<note 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**. </note>

La commande chkconfig

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

```
opensuse:/etc/rc.d # chkconfig --list
SuSEfirewall2_init        0:off  1:off  2:off  3:on   4:off  5:on   6:off
SuSEfirewall2_setup       0:off  1:off  2:off  3:on   4:off  5:on   6:off
aaeventd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

```
acpid                   0:off  1:off  2:on   3:on   4:off  5:on   6:off
after.local             0:off  1:off  2:off  3:off  4:off  5:off  6:off
alsasound               0:off  1:off  2:on   3:on   4:off  5:on   6:off
atd                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
auditd                  0:off  1:off  2:off  3:on   4:off  5:on   6:off
autofs                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
autoyast                0:off  1:off  2:off  3:off  4:off  5:off  6:off
avahi-daemon            0:off  1:off  2:off  3:on   4:off  5:on   6:off
avahi-dnsconfd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
before.local            0:off  1:off  2:off  3:off  4:off  5:off  6:off
bluez-coldplug          0:off  1:off  2:on   3:on   4:off  5:on   6:off
cifs                    0:off  1:off  2:off  3:off  4:off  5:off  6:off
cpufreq                 0:off  1:off  2:on   3:on   4:off  5:on   6:off
cron                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
cups                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dbus                    0:off  1:off  2:on   3:on   4:off  5:on   6:off
dnsmasq                 0:off  1:off  2:off  3:off  4:off  5:off  6:off
earlysyslog             0:off  1:off  2:on   3:on   4:off  5:on   6:off
earlyxdm                0:off  1:off  2:off  3:off  4:off  5:on   6:off
fbset                   0:off  1:on   2:on   3:on   4:off  5:on   6:off
gpm                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
haveged                 0:off  1:off  2:off  3:off  4:off  5:off  6:off
inputattach             0:off  1:off  2:off  3:off  4:off  5:off  6:off
jexec                   0:off  1:on   2:on   3:on   4:on   5:on   6:off
joystick                0:off  1:off  2:off  3:off  4:off  5:off  6:off
kbd                     0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
kexec                   0:off  1:off  2:off  3:off  4:off  5:off  6:off
lirc                    0:off  1:off  2:off  3:off  4:off  5:off  6:off
mdadmd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
microcode.ctl           0:off  1:off  2:off  3:off  4:off  5:off  6:off
multipathd              0:off  1:off  2:off  3:off  4:off  5:off  6:off
network                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
network-remotefs        0:off  1:off  2:off  3:on   4:off  5:on   6:off
nfs                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

```
nmb                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
nscd                    0:off  1:off  2:off  3:on   4:off  5:on   6:off
ntp                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
openvpn                 0:off  1:off  2:off  3:off  4:off  5:off  6:off
pm-profiler             0:off  1:off  2:off  3:off  4:off  5:off  6:off
postfix                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
powerd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
random                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
raw                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
rpcbind                 0:off  1:off  2:off  3:on   4:off  5:on   6:off
rpmconfigcheck          0:off  1:off  2:off  3:off  4:off  5:off  6:off
rsyncd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
setserial               0:off  1:off  2:off  3:off  4:off  5:off  6:off
skeleton.compat         0:off  1:off  2:off  3:off  4:off  5:off  6:off
smartd                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
smb                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
smolt                   0:off  1:off  2:off  3:off  4:off  5:off  6:off
smpppd                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
splash                  0:off  1:on   2:on   3:on   4:off  5:on   6:off  S:on
splash_early            0:off  1:off  2:on   3:on   4:off  5:on   6:off
sshd                    0:off  1:off  2:off  3:off  4:off  5:off  6:off
stoppreload             0:off  1:on   2:on   3:on   4:off  5:on   6:off
syslog                  0:off  1:off  2:on   3:on   4:off  5:on   6:off
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:on   4:off  5:on   6:off
xdm                     0:off  1:off  2:off  3:off  4:off  5:on   6:off
xfs                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
xinetd                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
ypbind                  0:off  1:off  2:off  3:off  4:off  5:off  6:off
xinetd based services:
        chargen:          off
        chargen-udp:      off
        cups-lpd:         off
```

```
        daytime:            off
        daytime-udp:        off
        discard:            off
        discard-udp:        off
        echo:               off
        echo-udp:           off
        netstat:            off
        rsync:              off
        sane-port:          off
        servers:            off
        services:           off
        swat:               off
        systat:             off
        time:               off
        time-udp:           off
        vnc:                off
```

Options de la commande

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

```
opensuse:/etc/rc.d # 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
        chkconfig --root=<root> ...  use <root> as the root file system
```

Les options les plus importantes sont :

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

Les services gérés par xinetd

Sous openSUSE, l'utilisation de xinetd permet :

- de centraliser la gestion de certains services réseaux sous le service xinetd,
- de démarrer les services ainsi gérés uniquement en cas de besoin.

Le programme xinetd est configuré via le fichier **/etc/xinetd.conf** :

```
opensuse:/etc/rc.d # cat /etc/xinetd.conf
#
# xinetd.conf
#
# Copyright (c) 1998-2001 SuSE GmbH Nuernberg, Germany.
# Copyright (c) 2002 SuSE Linux AG, Nuernberg, Germany.
#

defaults
{
    log_type        = FILE /var/log/xinetd.log
        log_on_success  = HOST EXIT DURATION
    log_on_failure    = HOST ATTEMPT
```

```
#        only_from       = localhost
        instances       = 30
    cps        = 50 10

#
# The specification of an interface is interesting, if we are on a firewall.
# For example, if you only want to provide services from an internal
# network interface, you may specify your internal interfaces IP-Address.
#
#   interface    = 127.0.0.1

}

includedir /etc/xinetd.d
```

Ce fichier ne définit pas les applications serveurs directement. Il indique plutôt le répertoire qui contient les fichiers de définitions des applications serveurs qui est **/etc/xinetd.d** :

```
opensuse:/etc/rc.d # ls /etc/xinetd.d
chargen      daytime      discard-udp  netstat    servers   systat    vnc
chargen-udp  daytime-udp  echo         rsync      services  time
cups-lpd     discard      echo-udp     sane-port  swat      time-udp
```

A l'examen de ce répertoire, vous noterez que celui-ci contient des fichiers nominatifs par application serveur.

Prenons le cas du script qui contrôle le service cups-lpd. Voici le contenu du fichier :

```
opensuse:/etc/rc.d # cat /etc/xinetd.d/cups-lpd
service printer
{
    disable     = yes
    flags       = NAMEINARGS
    socket_type = stream
    protocol    = tcp
```

```
    wait        = no
    user        = lp
    server      = /usr/lib/cups/daemon/cups-lpd
    server_args = cups-lpd -o document-format=application/octet-stream
}
```

Les directives principales de ce fichier sont :

Paramètre	Déscription
id	Identifie le service
socket_type	Nature du socket, soit **stream** pour TCP soit **dgram** pour UDP
protocol	Protocole utilisé soit **TCP** soit **UDP**
port	Le numéro de port ou à défaut le numéro indiqué pour le service dans le fichier /etc/services
wait	**no** : indique si xinetd active un serveur par client. **yes** : indique que xinetd active un seul serveur pour tous les client
user	Indique le compte sous lequel le serveur est exécuté
server	Indique le chemin d'accès de l'application serveur
server_args	Donne les arguments transmis à l'application serveur
disable	**no** : Le service est actif. **yes** : Le service est désactivité

Afin d'activer une application serveur, il suffit de modifier le paramètre **disable** dans le fichier concerné.

Arrêt du Système

La commande shutdown

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

- Il previent 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 :

```
opensuse:/etc/rc.d # shutdown --help
shutdown: invalid option -- '-'
Usage:    shutdown [-akrhPHfFnc] [-t sec] time [warning message]
          -a:      use /etc/shutdown.allow
          -k:      don't really shutdown, only warn.
          -r:      reboot after shutdown.
          -h:      halt after shutdown.
          -P:      halt action is to turn off power.
          -H:      halt action is to just halt.
          -f:      do a 'fast' reboot (skip fsck).
          -F:      Force fsck on reboot.
          -n:      do not go through "init" but go down real fast.
          -c:      cancel a running shutdown.
          -t secs: delay between warning and kill signal.
          ** the "time" argument is mandatory! (try "now") **
```

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 **time** 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

<note 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. </note>

L'utilisation de la commande **shutdown** peut être accordée à d'autres utilisateurs de root en utilisant le fichier **/etc/shutdown.allow**

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 :

```
opensuse:/etc/rc.d # reboot --help
usage: reboot [-n] [-w] [-d] [-f] [-h] [-i]
    -n: don't sync before halting the system
    -w: only write a wtmp reboot record and exit.
    -d: don't write a wtmp record.
    -f: force halt/reboot, don't call shutdown.
    -h: put harddisks in standby mode.
    -i: shut down all network interfaces.
```

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 :

```
opensuse:/etc/rc.d # halt --help
usage: halt [-n] [-w] [-d] [-f] [-h] [-i] [-p]
    -n: don't sync before halting the system
    -w: only write a wtmp reboot record and exit.
    -d: don't write a wtmp record.
    -f: force halt/reboot, don't call shutdown.
    -h: put harddisks in standby mode.
    -i: shut down all network interfaces.
    -p: power down the system (if possible, otherwise halt).
```

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 :

```
opensuse:/etc/rc.d # poweroff --help
usage: poweroff [-n] [-w] [-d] [-f] [-h] [-i]
```

```
    -n: don't sync before halting the system
    -w: only write a wtmp reboot record and exit.
    -d: don't write a wtmp record.
    -f: force halt/reboot, don't call shutdown.
    -h: put harddisks in standby mode.
    -i: shut down all network interfaces.
```

~~DISCUSSION:off~~

Donner votre Avis

{(rater>id=opensuse_11_l116|name=cette page|type=rate|trace=user|tracedetails=1)}