

CHAPITRE 2 : Démarrage et Arrêt d'un système Linux

Mohammed SABER

Département Électronique, Informatique et Télécommunications
École Nationale des Sciences Appliquées "ENSA"
Université Mohammed Premier OUJDA

Année Universitaire : 2018-2019

Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Introduction

Questions

- Comment démarre une machine Linux ?
- Comment intervenir en cas de problème ?
- Comment ajouter/supprimer des services activés au démarrage ?
- Comment arrêter/redémarrer une machine Linux ?

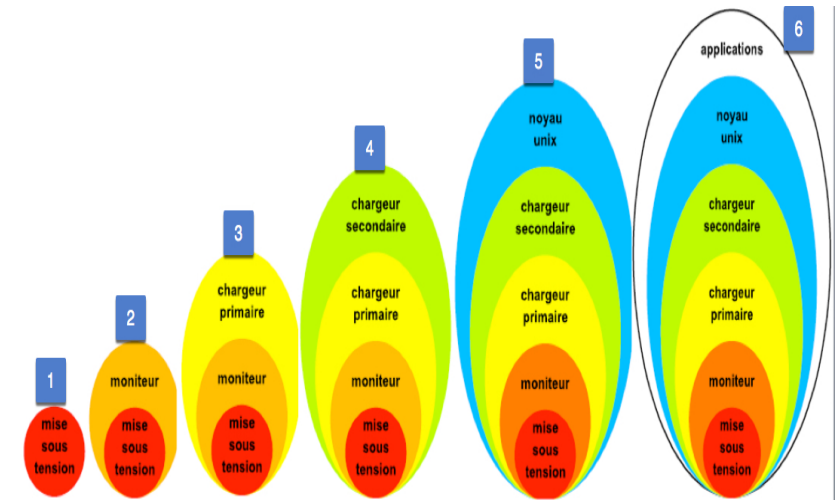
Système multi-tâches, multi-utilisateurs, multiplateformes

- Le démarrage d'un système Linux se divise en plusieurs étapes dont chacune est complexe.
- Pourquoi apprendre ce démarrage ?
 - D'abord pour le paramétrer,
 - Choisir de système d'exploitation de démarrage,
 - Démarrer différentes versions du noyau par exemple,
 - Choisir des services de démarrage,
 -

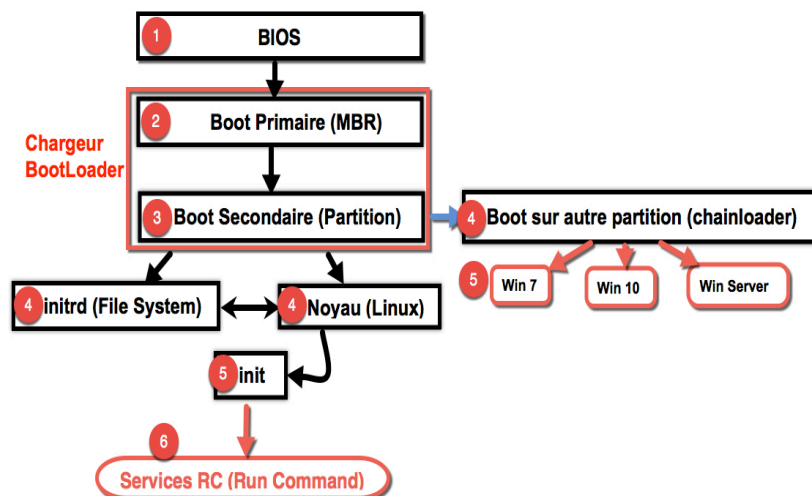
Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Les grandes étapes du démarrage



Les grandes étapes du démarrage



Les grandes étapes du démarrage : BIOS

- BIOS (Basic Input Output System) est le premier programme exécuté à l'allumage de l'ordinateur.
- Sans lui, la machine ne pourrait pas démarrer car le microprocesseur ne sait utiliser que les instructions stockées en mémoire vive ; la RAM étant dans un état indéfini à la mise sous tension,
- Ce code stocké dans une mémoire ROM permanente, éventuellement modifiable PROM ou EEPROM « flashable » par l'utilisateur.
- Aura donc pour fonction de :
 - Tester les circuits et composants matériels de la machine.
 - Initialiser le système d'affichage.
 - Tester la mémoire.
 - Détecter les périphériques de stockage (disque dur, lecteur disquette, lecteur de CD-ROM...).
 - Déterminer l'unité de disque de démarrage.
 - Lancer le gestionnaire d'amorçage présent sur le premier secteur du périphérique de démarrage ; pour charger le système d'exploitation présent sur ce périphérique.

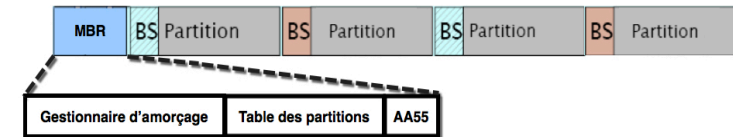
Les grandes étapes du démarrage : Gestionnaire d'amorçage

- Le « gestionnaire d'amorçage », « chargeur » ou « bootloader », est le plus petit programme lancé sur la machine après le BIOS.
- Sa tâche est de lire et de mettre en mémoire l'image du noyau d'un système d'exploitation.
- Celui-ci se trouve dans le premier bloc des données contenu sur le périphérique d'amorçage, plus connu sous le nom de « MBR » (Master Boot Record).
- Le MBR et le boot secteur constitue ce que l'on appelle le chargeur (loader).
- Sous Linux, on dispose essentiellement de 2 chargeurs (bootloader) possibles : LILO (Linux LOader) et GRUB (GRand Unified Bootloader).

Les grandes étapes du démarrage : Gestionnaire d'amorçage (Master Boot Record MBR (boot primaire))

Le format du premier bloc (secteur) de données de 512 octets est le suivant :

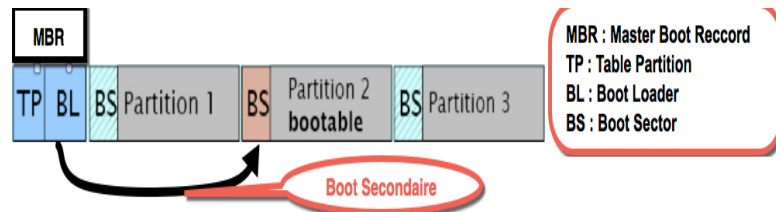
M B R	Partition primaire 1 /dev/hda1	Partition primaire 2 /dev/hda2	Partition étendue 3 /dev/hda3			
	Système Windows bootable	Data utilisateur	/dev/hda5 Linux Syst.	/dev/hda6 Data user	/dev/hda7 Swapp	/dev/hda8 Applications



- 446 octets pour le gestionnaire d'amorçage.
- 64 octets pour la table des partitions (16 octets pour définir une partition, soit 4 partitions au maximum sans compter les partitions logiques).
- 2 octets fixés à la valeur hexadécimale AA55 par convention ; sans cela, le MBR sera considéré comme erroné.
- Le MBR se contente la plupart du temps d'activer le boot secteur (secondaire).

Les grandes étapes du démarrage : Gestionnaire d'amorçage (Boot Secteur)

Le boot secteur (secondaire) a pour rôle d'activer un système d'exploitation.



Remarques

- Le MBR et le boot secteur constitue ce que l'on appelle le chargeur (loader).
- Sous Linux, on dispose essentiellement de 2 chargeurs (bootloader) possibles : LILO (Linux LOader) et GRUB (GRand Unified Bootloader).

Les grandes étapes du démarrage

Le noyau Linux (Kernel)

- Le noyau Linux prend possession de la mémoire et initialise les périphériques.
- Ensuite, il monte le système de fichiers racine et enfin active le premier programme : `/sbin/init` avec le **PID** égal à 1.

initrd

- Un fichier **initrd** contient un système de fichiers racine directement installé en mémoire par le chargeur.
- Son intérêt principal est de charger des modules du noyau nécessaires durant les premiers stades du démarrage, principalement des modules SCSI ou des modules de gestion de fichiers (Ext3, XFS, ...).

Les grandes étapes du démarrage

init

- Le programme **init** est le premier processus (PID=1), tous les autres processus sont ses descendants.
- Le rôle essentiel d'**init** est activé la première génération de processus ce qui débouche sur l'activation des **RC**.

Les RC (Run Command)

- Sont les scripts de démarrage des applications (services).
- L'administrateur Linux doit impérativement les connaître et savoir les gérer.

Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Démarrage d'un système par le chargeur GRUB

- GRUB (GRand Unified Bootloader) est un chargeur de système d'exploitation. Ce logiciel sous licence GPL a été écrit par E.S. Boleyn.
- GRUB est un multi-chargeur. Il peut activer des noyaux Linux associés à la même partition ou présents dans différentes partitions, ce qui permet de démarrer différentes distributions comme Windows, Unix SCO ou Unix BSD.

```
GNU GRUB version 2.00
Linux /boot/vmlinuz-3.8.0-19-generic
Linux /boot/vmlinuz-3.8.0-19-generic (single)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line. ESC to return
previous menu.
```

Les composantes du chargeur GRUB

Boot Secteur (Stage2)

Est le chargeur GRUB. Il se présente sous forme d'un fichier normalement installé dans le répertoire `/boot/grub`. Il affiche le menu de démarrage correspondant, sinon il affiche une invite de commande.

Boot Loader (Stage1)

- A pour but d'activer Boot Secteur (stage2). Si l'on utilise GRUB comme chargeur principal, on installe Stage1 dans le MBR.
- Si l'on utilise GRUB comme chargeur secondaire, on installe stage1 en tête de la partition de démarrage du système.
- Il faut dans ce cas disposer d'un chargeur principal. **Stage1** peut directement charger **stage2** ou charger un module **stage_1_5** qui lui-même charge stage2.

Fichier de configuration

Un fichier de configuration permet à GRUB d'offrir à l'utilisateur le choix du système à démarrer. Ce fichier, par défaut `/boot/grub/grub.cfg` (version 2) (et `/boot/grub/menu.lst` (version 1)) ou est chargé automatiquement.

Les principales commandes du chargeur GRUB

Partition : set root (root)

- **set root (root)** active une partition en tant que partition de démarrage.
- Une partition disque est représentée sous la forme (sd<disque>,<partition>) « les disques et les partitions sont numérotés à partir de zéro ».
- **root (sd0,1)** active la deuxième partition du premier disque.
- « root » et « rootnoverify ».

Noyau : linux (Kernel)

- Charge le noyau d'un système Linux ou BSD.
- On indique en argument le chemin du noyau et les arguments que l'on passe à ce noyau.
- Il faut préalablement monter la partition où réside le noyau par la commande **root**.
- L'argument **root** est quasi obligatoire.
- Linux /boot/vmlinuz-3.0.18.14 ro root=/dev/sda2 ou (Kernel /boot/vmlinuz-2.6.18.14 ro root=/dev/hda2)

Les principales commandes du chargeur GRUB

FileSystem : initrd

Le noyau Linux peut monter temporairement un système de fichiers racine à partir d'une image compressée stockée dans un fichier **initrd**.

boot

- Démarre un système d'exploitation.
- Elle provoque le démarrage du noyau préalablement chargé par la commande **linux (kernel)**.

chainloader

- La commande **chainloader** charge un fichier ou un ou plusieurs blocs jouant le rôle de chargeur secondaire.
- La commande précédente charge le chargeur présent dans le premier bloc de la partition de démarrage, activée par la commande **set root**.
- Cette commande peut être utilisée pour activer Windows (chainloader /bootsect.dos).

Les principales commandes du chargeur GRUB

Section menuentry (title)

- La commande **menuentry (Title)** débute une section décrivant un OS à l'intérieur d'un fichier de configuration.
- En argument, on indique la chaîne qui apparaît dans le menu.

OS de démarrage par défaut : default

La commande **default** précise, dans un fichier de configuration, l'OS à charger. Par défaut c'est le premier. La numérotation démarre à zéro.

Timeout

La commande **timeout** précise, dans un fichier de configuration, le délai en secondes, avant le démarrage automatique.

Démarrage interactif avec GRUB

Le chargeur GRUB, s'il ne trouve pas de menu, affiche un prompt et attend une commande. Si un menu est affiché, la touche C bascule en mode commande.

Démarrage d'un système Linux

- 1 `grub> set root`
- 2 `grub> linux`
- 3 `grub> initrd`
- 4 `grub> boot`

Démarrage d'un système Windows

- 1 `grub> set root`
- 2 `grub> chainloader +1`
- 3 `grub> boot`

Les fichiers de configuration

- Un fichier de configuration GRUB a : une section générale et une section pour chaque système d'exploitation que peut activer le chargeur.
- Chaque section décrivant un OS débute par la commande **menuentry** suivi de l'étiquette destinée à apparaître dans le menu.
- Le chemin par défaut du fichier de configuration de GRUB est `/boot/grub/grub.cfg` ou bien `/boot/grub/menu.lst`.

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Fedora Core (2.6.10-1.741_FC3)
    root (hd0,4)
    kernel /vmlinuz-2.6.10-1.741_FC3 ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.10-1.741_FC3.img
title Fedora Core (2.6.9-1.681_FC3)
    root (hd0,4)
    kernel /vmlinuz-2.6.9-1.681_FC3 ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.9-1.681_FC3.img
title Windows
    rootnoverify (hd0,0)
    chainloader +1
```

/boot/grub/menu.lst

Les fichiers de configuration

```
grub.cfg
set menu_color normal=cyan/blue
set menu_color_highlight=white/blue
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
menuentry "Ubuntu GNU/Linux, with Linux 2.6.31-14-generic" { 0
    insmod ext2
    set root=(hd0,6)
    search --no-floppy --fs-uuid --set ff848303-508a-46d6-b275-0c263f7b8938
    linux /boot/vmlinuz-2.6.31-14-generic root=UUID=ff848303-508a-46d6-b275-0c263f7b8938 ro splash vga=795 quiet splash
    initrd /boot/initrd.img-2.6.31-14-generic
}
menuentry "Ubuntu GNU/Linux, with Linux 2.6.31-14-generic (recovery mode)" 1
    insmod ext2
    set root=(hd0,6)
    search --no-floppy --fs-uuid --set ff848303-508a-46d6-b275-0c263f7b8938
    linux /boot/vmlinuz-2.6.31-14-generic root=UUID=ff848303-508a-46d6-b275-0c263f7b8938 ro single splash vga=795
    initrd /boot/initrd.img-2.6.31-14-generic
}
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_memtest86+ ###
menuentry "Memory test (memtest86+)" { 2
    linux16 /boot/memtest86+.bin
}
menuentry "Memory test (memtest86+, serial console 115200)" { 3
    linux16 /boot/memtest86+.bin console=ttyS0,115200n3
}
### END /etc/grub.d/20_memtest86+ ###

### BEGIN /etc/grub.d/30_os-prober ###
menuentry "Microsoft Windows XP Professional (on /dev/sda1)" { 4
    insmod ntfs
    set root=(hd0,1)
    search --no-floppy --fs-uuid --set 48142b45142b34fa
    drivemap -s (hd0) $(root)
    chainloader +1
}
```

/boot/grub/grub.cfg

Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Principes de fonctionnement

- Le système Linux offre plusieurs niveaux de fonctionnement.
- À chacun d'eux correspond un certain nombre de services pour les utilisateurs.
- Une commande (service) peut être associée à plusieurs niveaux. Cela permet de définir des services qui sont activés pour plusieurs niveaux. Pour le reste, on trouve deux cas de figure :
 - Un niveau englobe un autre. Il ajoute des services à ceux déjà offerts par le niveau précédent. Le niveau 4 offrira, par exemple, tous les services du niveau 3 plus un service particulier.
 - Un niveau offre des services totalement différents de ceux offerts par les autres niveaux. Le changement de niveau se traduit alors par une forme complètement nouvelle de fonctionnement.
- Au démarrage du système, lorsque le noyau crée le processus **init**, il ne spécifie pas de niveau. C'est le niveau défini par défaut dans **initab** qui choisi.
- Le rôle de la commande **init**, qu'a comme argument le niveau de fonctionnement choisi, est d'activer tous les processus associés à ce niveau.
- La commande **init** trouve la définition des commandes à exécuter pour un niveau donné dans la fichier `/etc/inittab`.

Les niveaux

- Niveau de fonctionnement = Ensemble des services.
- Il existe N niveaux (dépend la distribution) définies par les chiffres de 0 à 6.

```

root@fedora:~
File Edit View Terminal Tabs Help
[root@fedora ~]# cat /etc/inittab

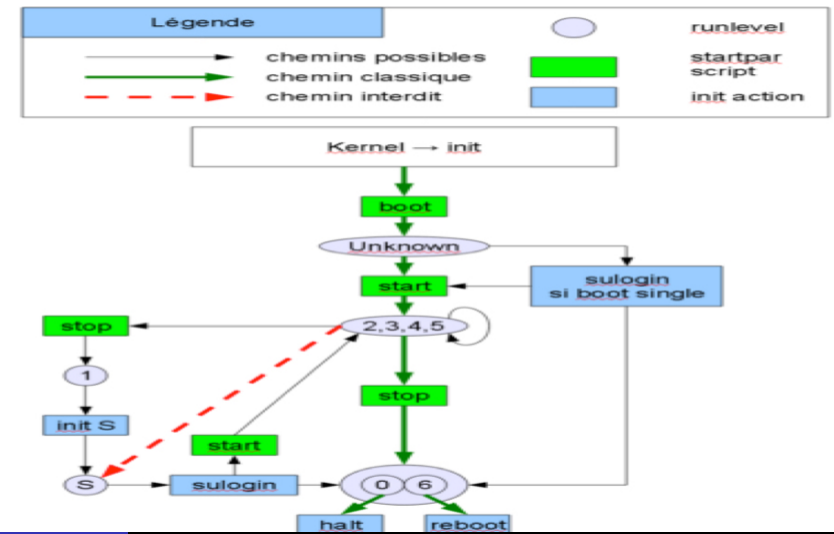
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)

# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

```

Debian

Les niveaux



Le fichier /etc/inittab

- Le fichier détermine quelles sont les commandes que le processus init doit exécuter pour un niveau donné. Le processus init l'examine séquentiellement. C'est un fichier de composé de lignes de texte.
- Une ligne de commande est composée de quatre champs séparés par le caractère « : »

ID :[niveau] :action :commande[#commentaire]

- **ID « identifiant »** : ce premier champ doit être unique et doit contenir de 1 à 4 caractères alphanumériques.
- **[niveau] Liste des niveaux d'exécution concernés** : ce champ contient le numéro (ou lettre) de champ runlevel concerné par la ligne. Si aucun niveau n'est spécifié, la ligne s'appliquera à tous les runlevels.
- **Commande** : chemin de la commande à lancer avec ses paramètres pour les niveaux d'exécution définis précédemment.

Le fichier /etc/inittab

Action : méthode ou manière d'exécuter la commande spécifiée dans le champ suivant est contient les directives suivantes :

ID :[niveau] :action :commande[#commentaire]

- **initdefault** : indique le mode de démarrage une fois le système opérationnel. Si aucun n'existe, **init** demandera un **runlevel** sur la console « demande à l'utilisateur le niveau de fonctionnement ».
- **sysinit** : Exécuté une seule et unique fois lors du démarrage du système.
- **boot** : le processus est exécuté pendant le démarrage du système après **sysinit**.
- **bootwait** : Idem, mais **init** attend la fin de l'exécution de la commande avant de continuer à dérouler le fichier **inittab**.
- **off** : la ligne ignorée.
- **once** : La commande est exécutée à chaque changement de niveau pour les niveaux Spécifiés (le processus n'exécuté qu'une seule fois).
- **wait** : Idem, mais **init** attend la fin de l'exécution avant de continuer.
- **ctrlaltdel** : le processus est exécuté lorsque **init** reçoit le signal **SIGINT**. (appui simultané des touches CTRL+ALT+DEL).

Le fichier /etc/inittab

debian

Le niveau d'exécution par défaut

Les niveaux d'exécution possibles

Action à faire sur l'évènement CTRL-ALT-DEL

Pour les niveaux 2 et 3, activer plusieurs consoles en mode caractère

```
# The default runlevel.
id:2:initdefault:

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

# What to do in single-user mode.
--:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt, 1 is single-user, 2-5 are multi-user, 6 is reboot.
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
... ..
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

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

1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
... ..
```

Commandes pour contrôler le processus init

- Ce mécanisme permet de changer de niveau d'exécution à tout moment sans redémarrer le système.
- L'administrateur pourra donc modifier l'état du système en contrôlant le processus **init**.
- Les commandes sont simples et peu nombreuses :
 - La commande **runlevel** affiche le niveau de fonctionnement courant ainsi que le précédent (**N** en premier position indiquant que le système n'a pas changé de niveau depuis le démarrage du système).

```
#runlevel
#N3 où #23
```

- Les commandes **init** et **telinit** permettent de changer de niveau.

```
#init 2 où telinit 2
#runlevel
#32
```

Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 **Les scripts de démarrage**
- 6 Arrêt du système

Les scripts de démarrage

- La connaissance de la structure du fichier **/etc/inittab** et la programmation en shell est théoriquement suffisante pour qu'un administrateur analyse toutes les opérations exécutées à partir de ce fichier.
- La plupart des commandes exécutées à partir de **inittab** sont en effet des scripts.

Le script rc.sysinit

Le processus **init** commence d'abord par exécuter le script **/etc/rc.d/rc.sysinit** qui contient des commandes de contrôle et d'initialisation indispensables au fonctionnement de Linux. Les principales opérations sont :

- Initialisation de la variable **PATH** avec les répertoires **/bin**, **/usr/bin**, **/sbin** et **/usr/sbin**.
- Initialisation des disques **swap**.
- Détermination du nom de la machine, le **hostname**.
- Montage de tous les systèmes de fichiers, à l'exception des systèmes de fichiers NFS.
- Chargement des pilotes dynamiques, les modules.
-

Les scripts de démarrage

Les RC « run commands »

Le processus **init** exécute ensuite les scripts spécifiques au niveau de démarrage demandé.

Le script rc.local

Le dernier script exécuté par **init** est le script run commands `/etc/rc.d/rc.local`, lorsque tous les autres services ont été lancés.

/etc/rc.d/init.d

Ce répertoire contient les scripts shell permettant de lancer tous les services sur le système. Les scripts présents ici doivent obligatoirement supporter l'argument **start** pour lancer le service et l'argument **stop** pour l'arrêter. Une grande majorité de ses scripts supportent aussi les arguments **restart**, **reload** et **status**.

`#/etc/init.d/nfs start` où **restart** où **reload** où **status** où **stop**.

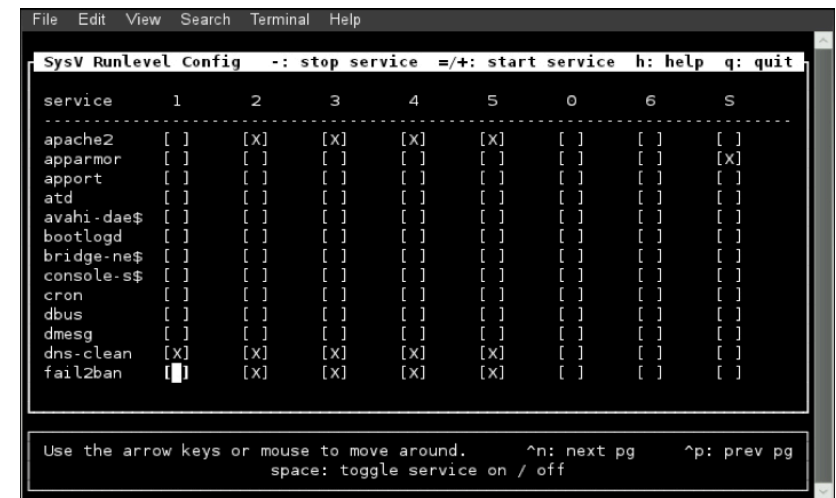
Ajouter et suppression de services au démarrage (chkconfig)

- Est une moyenne simple pour définir les niveaux pour lesquels un service doit être démarré ou arrêté.
- Elle permet de visualiser, d'ajouter ou de retirer un service pour un ou plusieurs niveaux de fonctionnement.
- Syntaxe :
 - Affiche les services et le mode arrêt ou démarrage selon les niveaux : `chkconfig --list service`
 - Ajout un service : `chkconfig --add service`
 - Retirer un service : `chkconfig --del service`
 - Positionner le mode arrêt ou démarrage selon les niveaux indiqués : `chkconfig --level niveaux service <on|off|reset>`

Ajouter et suppression de services au démarrage (chkconfig)

```
# chkconfig --list
Makefile                0:off 1:off 2:off 3:off 4:off 5:off 6:off
SuSEfirewall2_init      0:off 1:off 2:off 3:off 4:off 5:off 6:off
SuSEfirewall2_setup     0:off 1:off 2:off 3:on  4:on  5:on  6:off
VMd                      0:off 1:off 2:off 3:off 4:off 5:on  6:off
acpid                   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
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
bluetooth               0:off 1:off 2:off 3:off 4:off 5:off 6:off
boot.udev               0:off 1:off 2:on  3:on  4:off 5:on  6:off
bzipflagserver          0:off 1:off 2:off 3:off 4:off 5:off 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:off 3:on  4:off 5:on  6:off
dvb                     0:off 1:off 2:off 3:off 4:off 5:off 6:off
earlykbd                0:off 1:off 2:off 3:off 4:off 5:on  6:off
earlykdm                0:off 1:off 2:off 3:off 4:off 5:on  6:off
earlysyslog             0:off 1:off 2:off 3:off 4:off 5:on  6:off
esound                  0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Ajouter et suppression de services au démarrage (sysv-rc-conf)



Plan de chapitre

- 1 Introduction
- 2 Démarrage d'un système Linux
- 3 Démarrage d'un système par le chargeur GRUB
- 4 Niveaux de fonctionnement et Activation des processus
- 5 Les scripts de démarrage
- 6 Arrêt du système

Arrêt du système

Options shutdown

- `-h` pour arrêter le système
- `-r` pour redémarrer le système.
- `-c` pour annuler l'opération d'arrêt ou de redémarrage programmée.

Exemples shutdown

- `shutdown -h +10` : arrêt dans 10 minutes.
- `shutdown -r now` : redémarrage immédiat.
- `shutdown +5` : "arrêt pour sauvegarde" basculement en mode maintenance pour une sauvegarde.
- Plus d'information `man shutdown`.
- `Halt`, `reboot` et `poweroff` : (Voir `man`).

Arrêt du système

- L'arrêt d'un système GNU/Linux ne doit pas se faire en mettant hors tension la machine.
- Pour éviter toute mauvaise surprise comme la perte de données.
- Il faut exécuter un certain nombre de tâches avant de couper le courant, notamment :
 - Prévenir les utilisateurs connectés au système de l'arrêt/redémarrage de la machine pour qu'il puisse sauvegarde leur travail.
 - Arrêt tous les services.
 - Inscrire toutes les données contenues dans les tampon en mémoire sur le disque.

Commande shutdown

- Permet d'arrêter, de redémarrer et de faire passer en mode maintenance le système.
- Elle offre la possibilité de programmer cette opération à une date précise et d'en informer les utilisateurs.
- Syntaxe : `shutdown option "message"`

QUESTIONS ?