

Ecole Nationale des Sciences Appliquées Khouribga

UNIX GNU/Linux Processus de BOOT

Med AMNAI
2018–2019

Plan

- ❑ Processus de démarrage et d'arrêt
- ❑ X Windows

Processus de BOOT

Processus de démarrage et d'arrêt

1- Introduction

2- Programme INIT

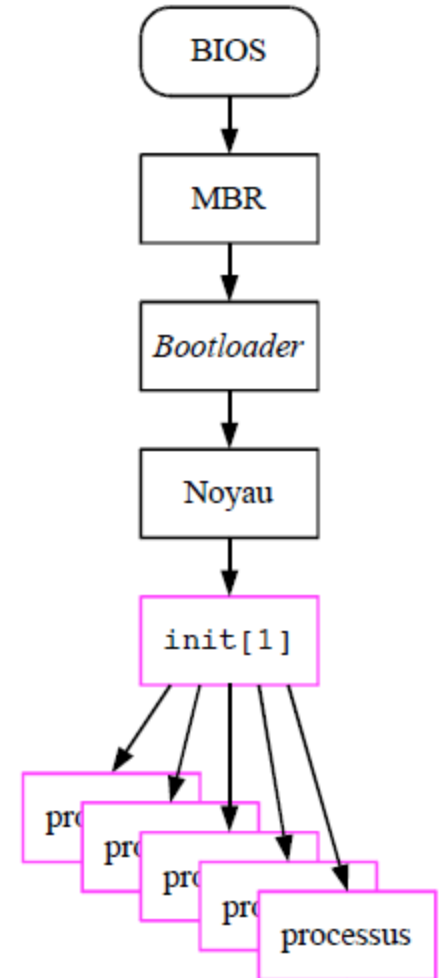
3- Fichier de configuration **/etc/inittab**

4- Les niveaux de fonctionnement (**runlevels**)

5- Scripts de démarrage

6- Contrôle du processus **init**

7- Ajout et suppression des services au démarrage



1-Introduction

- La procédure d'initialisation d'**Unix** utilise le concept de niveaux d'exécution "**runlevels**" pour définir les services devant être lancés au démarrage.
- Une fois son chargement terminé, le noyau Linux lance le **processus Init (/sbin/init)** qui **prend en charge l'initialisation du système** ;
- le processus (**init**) étant le premier processus lancé, il possède le **PID 1 (Process ID)**.
- L'affichage correspondant au début de cette phase lors du démarrage est reconnaissable sous RedHat par le message "**Press I to enter interactive startup**".

```
        Welcome to Fedora Core
        Press 'I' to enter interactive startup.
Starting udev:                               [ OK ]
Initializing hardware... storage network audio done [ OK ]
Setting clock (localtime): Thu Jan 26 11:29:25 EST 2006 [ OK ]
```

1- Introduction : Programme **init**

- Au démarrage, le Kernel (**noyau**) **charge les pilotes de périphériques** pour lesquels il a été configuré.
- Puis il **lance /sbin/init (-> upstart)** (**PID 1**), **Seul et unique** processus lancé par le Kernel,
- **Init** a pour tâche :
 - lancer chacun des processus;
 - les sessions de login;
 - Il doit aussi collecter les zombies (processus orphelins);
 - gérer l'arrêt du système;
 - prend en charge la suite des opérations;
 - **Init** lit le fichier de configuration **/etc/inittab (id:5:initdefault:)**
 - Par la suite le Kernel n'interviendra plus que pour répondre aux appels système.

1-Introduction : **inittab**

- ❑ Le processus **Init** a pour mission **d'initialiser l'environnement logiciel du système et de lancer les services.**
- ❑ A son lancement le programme **Init** lit le fichier de configuration [/etc/inittab](#) qui détermine le [comportement de **Init**](#).
- ❑ **Les tâches d'initialisation comprennent (/etc/rc.d/rc.sysinit) :**
 - La mise à l'heure du système par rapport à l'heure matérielle
 - La définition des consoles
 - La définition du nom de la machine
 - L'activation des quotas
 - L'activation du swap
 - La vérification des systèmes de fichiers et le montage de ceux-ci
 - Le lancement du démon **Syslog** de journalisation des messages
 - Le chargement des modules du noyau
 - L'initialisation des variables d'environnement (**PATH, HOME,..**).

Les services démarrés dépendront du niveau d'exécution spécifié au processus Init
(id:5:initdefault:)

2- Fichier de configuration **/etc/inittab**

Pour définir le comportement du système et les services à démarrer en fonction du niveau d'exécution, **Init** se réfère au fichier **/etc/inittab**.

Le fichier **/etc/inittab** est un fichier texte composé de lignes de commentaires (commençant par **#**) et d'entrées constituées de **4** champs délimités par des deux-points, elles sont de la **forme** :

```
id:runlevels:action:process [arguments]
```

- ❑ **id** : Identifiant unique qui comprend de 1 à 4 caractères alphanumériques qui identifie l'entrée dans le fichier **/etc/inittab**.
- ❑ **runlevels** : Définit la liste des **niveaux d'exécution** pour lesquelles cette ligne s'applique. Si cette **liste est vide**, cela équivaut à **indiquer tous les niveaux**.
- ❑ **action** : Méthode ou **manière d'exécuter** la commande spécifiée dans le champ suivant (process).
- ❑ **process** : Chemin de la commande à lancer avec ses paramètres pour les niveaux d'exécution définis précédemment.

2- Fichier de configuration `/etc/inittab`

Les différentes directives pour le troisième champ "action" sont :

- **Respawn** Le processus sera **relancé s'il se termine**. Utilisé principalement avec **mingetty** pour assurer la *gestion des terminaux texte* :

```
1:2345:respawn:/sbin/mingetty 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
```

- **Once** Le processus n'est exécuté qu'une seule fois.

```
x:5:once:/etc/X11/prefdm nodaemon
```

- **Wait** Identique à la directive précédente mais ici, **Init attend** que le processus soit terminé avant de passer à la ligne suivante.

Ce type d'entrée sert principalement à lancer les scripts d'initialisation pour chaque niveau. Exemples : Indique qu'il faut lancer le script **`/etc/rc.d/rc n`**.

```
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
```

2- Fichier de configuration /etc/inittab

Les différentes directives pour le troisième champ "action" sont : (suite)

- **Initdefault (1)** Définit le niveau d'exécution **par défaut au démarrage du système**, le champ runlevels est ignoré. Si cette directive est absente, **Init** le demandera sur la console. Le champ commande est ignoré.
- **Sysinit (2)** Le processus doit s'effectuer **au démarrage du système**, avant toute entrée d'action **boot ou bootwait**. Le champ runlevels est ignoré.

Exemple :

```
si::sysinit:/etc/rc.d/rc.sysinit
```

- **Boot (3)** Le processus est exécuté **au démarrage du système**, le champ runlevels est ignoré.
- **Bootwait** Identique à la directive précédente mais ici, **Init attend que le processus** soit terminé avant de passer à la ligne suivante.
- **Off** Ne fait rien.
- **Ondemand** Identique à **respawn** mais cette directive utilise les pseudos **runlevels a, b et c**. Cela permet de demander à **Init** d'entreprendre une action sans changer de niveau d'exécution.
- **Powerfail** La commande est exécutée lorsque **Init reçoit le signal SIGPWR** signifiant que **l'alimentation** est sur le point d'être **interrompue**. Ce signal est envoyé par un dispositif de gestion d'énergie tel qu'un **UPS**.

2- Fichier de configuration /etc/inittab

Les différentes directives pour le troisième champ "action" sont : (suite)

- **Powerwait** Identique à la directive précédente mais ici, **Init attend que le processus soit terminé** avant de passer à la ligne suivante.
- **Powerokwait** Le processus est exécuté si **Init est informé du rétablissement de l'alimentation**.
- **powerfailnow** Le processus est exécuté si **Init est informé que l'accumulateur de l'UPS externe est presque vide**.
- **Ctrlaltdel** La commande est lancée lorsque **Init reçoit le signal SIGINT** (généré par la combinaison de touches **CTRL-ALT-DEL**). Elle sert généralement à arrêter et/ou redémarrer le système. Exemple

```
ca::ctrlaltdel:/sbin/shutdown -t3 -h now
```

- **Kbrequest** Permet de **lancer un processus suivant certaines séquences de touches saisies au clavier**.

2- Fichier de configuration **/etc/inittab**

L'analyse du fichier de configuration **/etc/inittab** nous apprend que :

- Le niveau d'exécution par défaut est le niveau **5**. Mettre cette valeur à **3** si on ne veut pas lancer systématiquement le système **X Window** à chaque démarrage; il faudra alors saisir la commande **startx** une fois connecté au système pour lancer l'interface graphique.
- La commande **/etc/rc.d/rc.sysinit** est lancée en premier lieu.
- La commande **/etc/rc.d/rc** est lancée pour chaque niveau d'exécution avec son numéro en argument.
- La commande **/sbin/shutdown** est exécutée avec différents arguments lors de l'appui sur les touches **CTRL-ALT-DEL** et lors d'un événement concernant l'alimentation.
- Six terminaux virtuels "**tty**" sont initialisés avec la commande **/sbin/mingetty** pour les niveaux d'exécution 2, 3, 4 et 5. De plus, ceux-ci seront relancés s'ils se terminent.
- Pour le niveau d'exécution 5, l'outil **/etc/X11/prefdm** est exécuté ; cette commande lance le programme de connexion à la console graphique.

3- Les niveaux de fonctionnement (runlevels)

Il existe huit **runlevels** définis sous Linux dont quatre réservés (0, 1, 6, S ou s) :

NIVEAU	DESCRIPTION
0	Halt - Arrêt de la machine.
1	Single user- Mode mono-utilisateur ou maintenance.
2	Multi-utilisateurs sans le support NFS.
3	Multi-utilisateurs.
4	Libre.
5	X11 (Multi-utilisateurs avec graphique).
6	Reboot - redémarrage de la machine.
S ou s	Mode mono-utilisateur dans lequel seule la partition racine est montée.

- L'administrateur pourra sélectionner les services associés à chaque **runlevel**.
- Lors de la phase de démarrage, **Init** doit savoir à quel niveau placer le système. Pour cela, il recherche dans le fichier **/etc/inittab** la ligne qui configure ce niveau par défaut :

id:3:initdefault:

Signifie que le système démarrera en Mode **multi-utilisateurs** complet.

4-Scripts de démarrage :/etc/rc.d

Un certain nombre de commandes exécutées par **Init** se trouvent dans */etc/rc.d* :

```
[amnai@localhost ~]$ ls -l /etc/rc.d
total 60
drwxr-xr-x. 2 root root 4096 24 sept. 19:34 init.d
-rwxr-xr-x. 1 root root 2625 27 sept. 2010 rc
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc0.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc1.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc2.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc3.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc4.d
drwxr-xr-x. 2 root root 4096 28 sept. 15:07 rc5.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc6.d
-rwxr-xr-x. 1 root root 220 27 sept. 2010 rc.local
-rwxr-xr-x. 1 root root 19021 27 sept. 2010 rc.sysinit
[amnai@localhost ~]$
```

4-Scripts de démarrage : **/etc/rc.d/rc.sysinit**

- ❑ Le programme **Init** exécute le script **/etc/rc.d/rc.sysinit** avant tous les autres.
- ❑ S'occupe de tout ce qui doit être fait sur le système au moment de son initialisation.
- ❑ Ce script permet entre autre :
 - la vérification des systèmes de fichiers,
 - le montage du répertoire de swap,
 - définition du nom de la machine
 - l'initialisation du système de traçage (log),
 - l'initialisation de l'horloge,
 - configurer la variable d'environnement PATH
 -

4-Scripts de démarrage : `/etc/rc.d/init.d`

Le répertoire `init.d` contient les scripts Shell permettant de lancer tous les services sur le système.

- Maintenir tous les scripts dans une seule location facilite grandement leur gestion.
- Chaque script supporte l'argument **start (S*)** pour lancer le service et **stop (K*)** pour l'arrêter.
- Une grande majorité de ces scripts supportent aussi les arguments **restart**, **reload** et **status**.
- Chaque script est utilisé pour démarrer ou arrêter un service particulier.

4-Scripts de démarrage :/etc/rc.d/rc

- ❑ **rc**, la commande `/etc/rc.d/rc` est lancée pour chaque **niveau d'exécution** avec son **numéro en argument**.
- ❑ Elle est appelée par **Init avec un paramètre** correspondant au **niveau d'exécution** dans lequel on entre.
- ❑ Ce paramètre est récupéré par le script **rc** pour **former un nom de répertoire**.
- ❑ Par exemple, l'appel `/etc/rc.d/rc 2` fera que **rc** ira examiner le répertoire `/etc/rc.d/rc2.d`.
- ❑ Le script **rc** joue un rôle **d'aiguillage** et **d'exécution** d'autres scripts.

exemple :

```
l2:2:wait:/etc/rc.d/rc 2
```

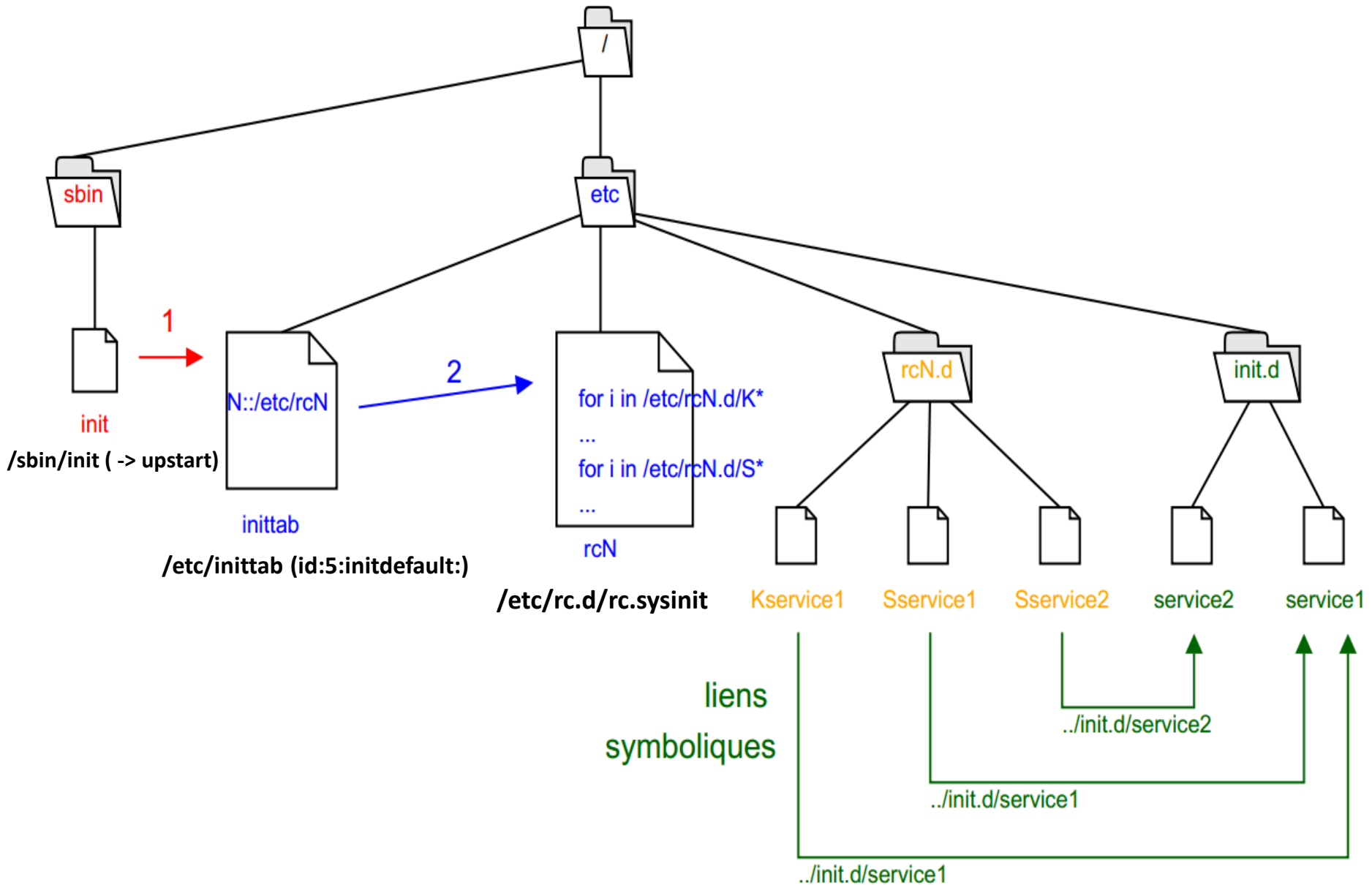
Signifie que **Init** doit lancer le **script** `/etc/rc.d/rc` en lui passant **2** en paramètre à chaque fois qu'on entre dans le niveau 2 et qu'il doit attendre la terminaison de ce script avant de poursuivre.

4-Scripts de démarrage :/etc/rc.d/rc0 ..rc6

- ❑ Pour éviter de dupliquer des scripts et donc de compliquer les manipulations lors de la mise à jour de ceux-ci,
- ❑ les répertoires **rc0.d à rc6.d** ne **contiennent** que **des liens symboliques**, avec des noms très précis, qui font référence **aux scripts réels** qui se trouvent **dans le répertoire /etc/rc.d/init.d**.

```
12:2:wait:/etc/rc.d/rc 2
```

4-Scripts de démarrage : `/etc/rc.d/rc0` .. `rc6`



4-Scripts de démarrage : `/etc/rc.d/rc.local`

- ❑ comme son nom l'indique, il est "**local**" à votre machine : c'est dans ce fichier que vous **ajouterez les commandes d'initialisation propres à votre système.**
- ❑ **`rc.local`** est appelé en dernier, c'est-à-dire après tous les autres scripts.
- ❑ Ce script peut être utilisé par l'administrateur pour exécuter certaines tâches une seule fois à la fin du niveau d'exécution correspondant.

Exemple : après chaque démarrage du serveur on voudra démarrer la base de données.

4-ARRÊT DU SYSTÈME

Pour éviter toute mauvaise surprise comme la perte de données. Il faut exécuter un certain nombre de tâches avant de couper le courant :

- ❑ **Prévenir les utilisateurs** connectés au système de l'arrêt imminent de la machine.
- ❑ **Arrêter tous les services.**
- ❑ **Vider les buffers** sur le disque et démonter les systèmes de fichiers.
Sans cela, les données se trouvant dans le cache et non encore écrites sur les unités de stockage seront perdues.

5-ARRÊT DU SYSTÈME : **shutdown**

- ❑ La commande **shutdown** permet **d'arrêter**, de **redémarrer** et de passer le système en **mode maintenance**.
- ❑ Elle offre la possibilité de **programmer** cette opération à une date précise et d'en informer les utilisateurs.
- ❑ Si l'arrêt du système est prévu dans moins de cinq minutes, la commande **shutdown** **empêchera** tout utilisateur, autre que **root** de **se connecter**.
- ❑ Sa syntaxe est la suivante :

```
/sbin/shutdown [-t sec] [-arkhncfF] heure [messages]
```

L'heure peut être spécifiée de plusieurs manières :

- **hh:mm** heure à laquelle l'opération est programmé
- **[+]m** nombre de **minutes** avant que l'opération soit effectuée
- **now** l'opération doit être **immédiate (alias de +0)**

5-ARRÊT DU SYSTÈME : **shutdown**

Les options à retenir sont :

- ☐ **-h (halt)** arrêter le système.
- ☐ **-r (reboot)** redémarrer le système.
- ☐ **-c (cancel)** annuler l'opération d'arrêt ou de redémarrage programmée.
- ☐ **-f** effectuer un **redémarrage rapide** sans vérification des systèmes de fichiers.
- ☐ **-F** **forcer** la **vérification** des systèmes de fichiers au prochain démarrage.

6-Contrôle du processus INIT

Connaître le niveau d'exécution courant

- ❑ La commande **who** permet d'indiquer le **niveau d'exécution** courant :

\$who -r

- ❑ On peut aussi utiliser une autre commande pour avoir cette information :

\$runlevel

❑ **Changement de niveau d'exécution**

Pour contrôler le processus **init** et changer de **runlevel**, il suffit d'appeler la commande **init** ou **telinit** avec le niveau d'exécution (**0, 1, 2, 3, 4, 5, 6, S, ou s**) en argument :

\$telinit n (n : le niveau)

6-Contrôle du processus INIT

Relire /etc/inittab

Lorsqu'il démarre, **Init** lit le fichier **/etc/inittab** ligne par ligne (si on modifie ce dernier, il n'est pas nécessaire de **rebooter** la machine. Il suffit d'envoyer un signal **HUP** à **Init** pour le forcer à le relire avec la commande :

\$kill -HUP 1

On peut aussi forcer **Init** à effectuer une relecture de **/etc/inittab** en utilisant la commande **telinit** avec l'option **q** ou **Q** :

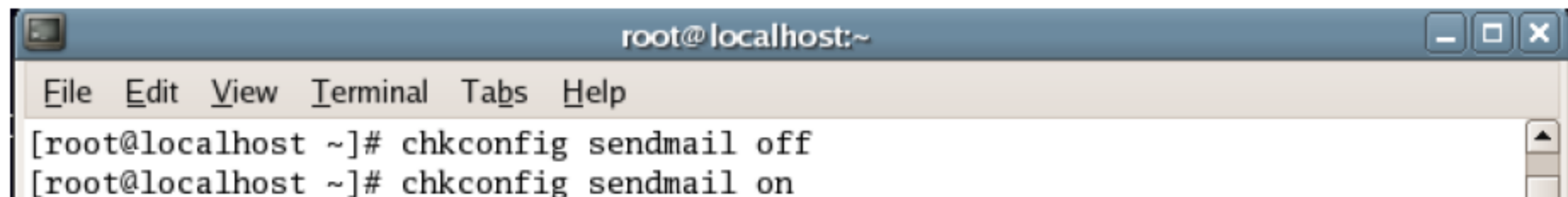
\$telinit q

7-Ajout et suppression de service au démarrage

Le programme `/etc/rc.d/rc` exécutant tous les scripts présents dans le répertoire du **runlevel**, il suffira d'ajouter un lien dans le répertoire vers le script adéquat pour **lancer** ou **arrêter** un service lorsque le système entre dans ce niveau d'exécution. Il existe des outils en ligne de **commandes** ou avec une interface **graphique** qui permettent **l'ajout** et **la suppression de services au démarrage** :

chkconfig

La commande **chkconfig** permet de *lister*, *d'ajouter* et de *supprimer* les différents services des **runlevels**.

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows two commands being executed: '[root@localhost ~]# chkconfig sendmail off' and '[root@localhost ~]# chkconfig sendmail on'.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# chkconfig sendmail off  
[root@localhost ~]# chkconfig sendmail on
```

7-Ajout et suppression de service au démarrage

ntsysv

- ❑ La commande **ntsysv** affiche dans un menu tous les scripts présents dans */etc/rc.d/init.d*
- ❑ **ntsysv** propose de les **ajouter** ou **supprimer** du niveau d'exécution courant ou de ceux spécifiés sur la ligne de commandes avec l'option **--level**.

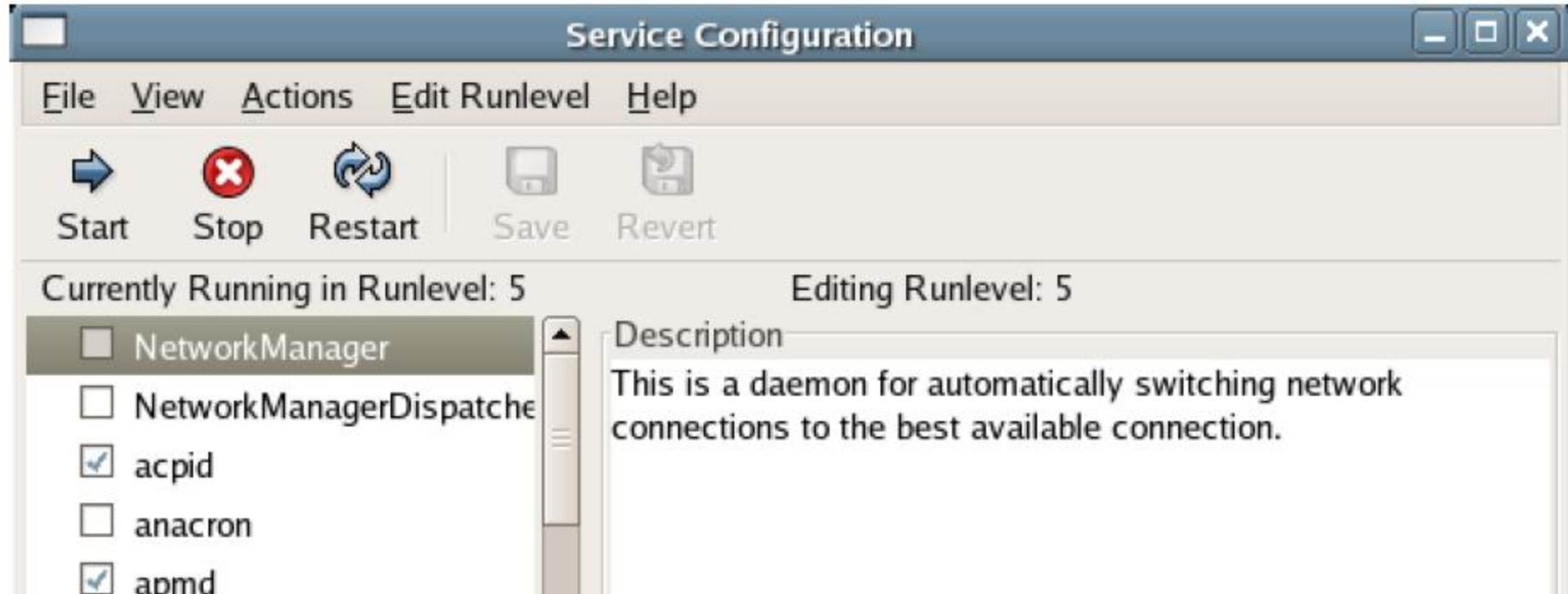
\$ntsysv

\$ntsysv --level 345

7-Ajout et suppression de service au démarrage

system-config-services

Cet outil est accessible via le menu **GNOME**.



8-MESSAGES

- ❑ La commande **dmesg** permet **d'examiner** ou de **contrôler** le **tampon des messages** du **noyau** (kernel).
- ❑ Cette commande permet d'afficher les messages du **démarrage de la machine**.

```
[ 107.212152] readahead-collector: starting delayed service auditd
[ 109.089264] readahead-collector: sorting
[ 114.628585] readahead-collector: finished
[ 119.232726] SELinux: initialized (dev mqueue, type mqueue), uses transition SIDs
[ 119.242794] SELinux: initialized (dev proc, type proc), uses genfs_contexts
[ 119.311518] SELinux: initialized (dev mqueue, type mqueue), uses transition SIDs
[ 119.388449] lo: Disabled Privacy Extensions
[ 119.393568] SELinux: initialized (dev proc, type proc), uses genfs_contexts
[ 198.482603] fuse init (API version 7.14)
[ 198.628133] SELinux: initialized (dev fuse, type fuse), uses genfs_contexts
[ 198.662481] SELinux: initialized (dev fusectl, type fusectl), uses genfs_contexts
[root@localhost amnai]#
```

Structure de la configuration :Lilo

Configuré par un fichier texte : **/etc/lilo.conf**

La configuration de **LILO** comporte :

- des options globales
- N entrée(s) de type « **image** » pour chaque **boot Linux**
- N entrée(s) de type « **other** » pour tout **autre système** d'exploitation

Pour appliquer des changement sur l'ordre de démarrage exécuter « **/sbin/lilo** »

```
# Options globales
boot=/dev/hda
lba32
keytable=/boot/fr-latin1.klt
prompt
timeout=300
default=Linux

# Section Linux
image=/boot/vmlinuz-2.4.10
    label=Linux
    read-only
    root=/dev/hda1

# Section NT
other=/dev/hda2
    label=NT
```

Structure de la configuration :grub

Installation : grub-install /dev/hda

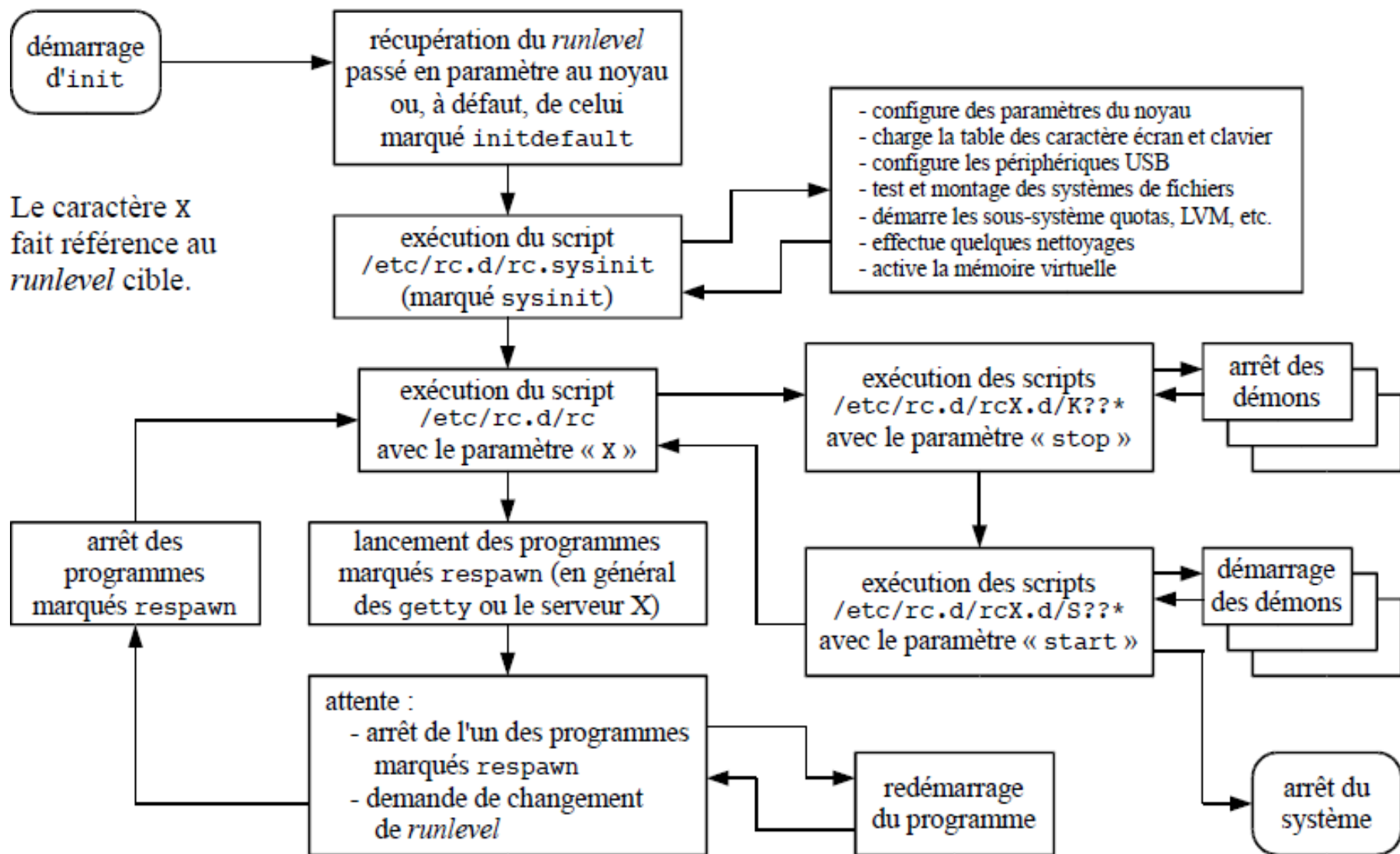
Configuration: On utilisera le fichier /boot/grub/menu.lst ou /etc/grub.conf

```
# Options globales
    boot=/dev/hda
    default=1
    timeout=10
    splashimage=(hd0,6)/boot/grub/splash.xpm.gz

# Section Linux
title Red Hat Linux (2.4.18-14)
    root (hd0,6)
    kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/
    initrd /boot/initrd-2.4.18-14.img

# Section NT
title Windows 2000 Pro
    root (hd0,0)
    makeactive
    chainloader +1
```

Démarrage – le processus complet



X Windows

X Windows

Définitions

- ☐ X-Window est l'interface graphique d'Unix
- ☐ XFree86 est le nom du projet de « portage » d'X-Window sur les Unix libres
- ☐ X-Window est **seulement** l'interface graphique
- ☐ Les environnements graphiques (Window Managers) sont des sur-couches de X-Window