Le processus init et les niveaux d'exécution

1. Les niveaux d'exécution

Le fonctionnement d'un système Linux est régi par des niveaux d'exécution. Même si ce concept apparaît davantage aujourd'hui comme un héritage du passé que comme un réel outil d'administration d'un poste de travail ou d'un serveur Linux, sa connaissance est indispensable à une bonne gestion du système.

Tout d'abord, il faut admettre qu'un système Linux est toujours dans un niveau d'exécution quelque soit son activité, qu'il s'agisse d'un serveur apache en train de répondre à une requête, ou d'un serveur neuf encore dans son carton. La gestion des niveaux d'exécution consistera à déterminer quel doit être le comportement du système quand il entre dans un niveau donné.

a. Qu'est-ce qu'un niveau d'exécution ?

Pour faire simple, un niveau d'exécution est un niveau fonctionnel dans lequel on aura déterminé la liste des services à arrêter ou à démarrer. Quand un système entre dans un niveau d'exécution, il regarde s'il doit arrêter et/ou démarrer des services.

b. Les niveaux d'exécution possibles

Le niveau 0

Le plus simple : le système est arrêté. Attention, cela ne signifie pas que ce niveau ne doit pas être configuré, il faut tout de même gérer ce qui se passe quand le système entre en niveau 0, c'est à dire quels sont les services à arrêter quand on éteint physiquement une machine.

Le niveau 1 ou single

Un niveau un peu particulier : il est réservé aux opérations de maintenance et ne permet qu'une seule connexion, celle du compte root. De plus, la plupart des services sont arrêtés dans ce niveau, ce qui signifie que le système a une activité minimum. C'est parfait pour l'administrateur qui souhaite effectuer des opérations de maintenance sans interférer avec la production.

Le niveau 2

Sur la plupart des systèmes, ce niveau n'est pas utilisé. Il est laissé à la disposition de l'administrateur qui pourra établir à partir de ce niveau un mode de fonctionnement particulier avec seulement certains services démarrés.

Sur les systèmes Debian et dérivés (Ubuntu par exemple), ce niveau est en revanche le niveau fonctionnel par défaut.

Le niveau 3

Sur la plupart des systèmes, le niveau 3 est fonctionnel, c'est-à-dire que tous les services sont démarrés, mais l'interface graphique n'est pas disponible.

Le niveau 4

Sur la plupart des systèmes, ce niveau n'est pas utilisé. Il est laissé à la disposition de l'administrateur qui pourra établir à partir de ce niveau un mode de fonctionnement particulier avec seulement certains services démarrés.

Le niveau 5

Sur la plupart des systèmes, le niveau 5 est fonctionnel, c'est-à-dire que tous les services sont démarrés, et l'interface graphique est disponible.

Sur les systèmes Debian et dérivés (Ubuntu par exemple), ce niveau n'est pas utilisé en général.

Le niveau 6

Temporaire par définition, le niveau 6 est celui d'un système en train de redémarrer. La configuration du niveau 6 consistera donc à déterminer quels services doivent être arrêtés au redémarrage du système. Après le redémarrage, un nouveau niveau d'exécution s'appliquera (en général le niveau par défaut) et les services associés à ce niveau seront démarrés.

c. Qui décide de ce qu'on met dans les différents niveaux ?

Dans l'immense majorité des cas, c'est la définition initiale des niveaux d'exécution qui est exploitée. C'est-à-dire que le gestionnaire de la distribution (Ubuntu, Mandriva, Red Hat, etc.) choisit ce qui doit se passer dans chacun des niveaux d'exécution donné, et l'administrateur du système fait avec.

Toutefois, il peut arriver que l'administrateur du système préfère gérer lui même la configuration de ses niveaux d'exécution. Il peut alors choisir à quel niveau fonctionnel correspond chacun des niveaux d'exécution et quels sont les services associés. À chaque niveau d'exécution correspond alors un ensemble de services.

2. Configuration du processus init

Nous avons parlé jusqu'à présent des niveaux d'exécution comme d'une liste de services à arrêter ou démarrer. La question est maintenant de savoir comment le système va prendre connaissance de son niveau d'exécution et des services qui y sont référencés.

a. Le premier processus démarré sur le système

Si on regarde quels sont les processus s'exécutant sur le système, on trouve en première position le processus init. Il porte un PPID inhabituel (0), et il est le père de nombreux autres processus.

Les processus

De nombreux processus ont le numéro 1 comme PPID.

```
[root@beta ~]# ps -ef | head
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 09:07 ?        00:00:12 init [5]
root         2     1  0 09:07 ?        00:00:00 [migration/0]
root         3     1  0 09:07 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 09:07 ?        00:00:00 [watchdog/0]
root         5     1  0 09:07 ?        00:00:09 [events/0]
root         6     1  0 09:07 ?        00:00:00 [khelper]
root         7     1  0 09:07 ?        00:00:00 [kthread]
root        10     7  0 09:07 ?        00:00:04 [kblockd/0]
root        11     7  0 09:07 ?        00:00:00 [kacpid]
[root@beta ~]#
```

Ce processus est le premier lancé au chargement du noyau. Il a évidemment un rôle privilégié, et son comportement est régi par un fichier de configuration : **/etc/inittab**.

b. Le fichier inittab

Selon les distributions, le fichier **/etc/inittab** revêt des contenus très différents, mais sa structure est toujours la même.

Structure du fichier /etc/inittab

identifiant:niveau:mode_action:commande

Fichier /etc/inittab : structure d'une ligne de définition	
identifiant	Chaîne alphanumérique d'un ou deux caractères. Identifie la ligne. Pas d'autres contraintes que d'éviter d'avoir deux lignes avec le même identifiant.
niveau	Le ou les niveaux d'exécution (en chiffres) pour lesquels la ligne est pertinente.
mode_action	À choisir parmi quelques mots-clés, définit la façon dont la commande du quatrième champ sera exécutée.
commande	La commande à exécuter au(x) niveau(x) défini(s) dans le deuxième champ selon le

	mode d'action du troisième champ.

Modes d'actions courants

- **initdefault** : un peu particulier, initdefault ne régit pas la façon dont la commande du quatrième champ sera exécutée. D'ailleurs, quand le mode d'action est initdefault, le quatrième champ est vide. initdefault ne sert en fait qu'à définir le niveau d'exécution du système par défaut.

- **sysinit** : sert à exécuter des scripts à l'initialisation du système, indépendamment du niveau d'exécution. Pour cette raison, sysinit n'admet pas de valeur pour le deuxième champ.

- **wait** : exécute la commande du quatrième champ (souvent un script), et attend la fin de cette exécution pour passer aux lignes suivantes du fichier inittab.

- **respawn** : exécute la commande du quatrième champ, et laisse tourner le processus à l'arrière-plan. Passe ensuite aux lignes suivantes du fichier inittab. Si le processus appelé par la commande s'arrête, init le relancera systématiquement.

Fichier inittab d'une distribution RedHat

Les commentaires ont été supprimés pour des raisons de lisibilité.

```
id:5:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
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
x:5:respawn:/etc/X11/prefdm -nodaemon
```

c. Rappels sur le lancement des services

Sur un système Linux, les services sont lancés par des scripts normalisés qui répondent à au moins deux conditions :

- Ils se trouvent tous dans le répertoire **/etc/init.d** (ou sont disponibles à cet emplacement sous forme de lien symbolique).

- Ils admettent tous les paramètres **start** et **stop** pour le lancement et l'arrêt du service.

Syntaxe universelle de gestion de service

`/etc/init.d/`*`nom action`*

Gestion de services avec la commande service

`service` *`nom action`*

Gestion de service : paramètres	
nom	Le nom du service à gérer.

action	**start** ou **stop** pour démarrer ou arrêter le service. **status** est aussi une option couramment supportée qui indique l'état du service.

La commande **service**, quand elle est disponible, peut être considérée comme préférable car elle lance le service en s'affranchissant autant que possible de l'environnement ambiant (pwd et variables). Le service est ainsi démarré dans un environnement plus neutre.

Format standard d'un script de gestion de service

```
#!/bin/bash
case $1 in
start)
  # commande de lancement du service
;;
stop)
  # commande d'arrêt du service
;;
esac
```

d. Liens entre les niveaux d'exécution et les services

Si on regarde le fichier **/etc/inittab**, on trouve une section contenant 7 lignes commandant pour chacun des niveaux d'exécution un script **/etc/init.d/rc** en mode **wait**. Nous ne détaillerons pas le fonctionnement de ce script ici, mais retenons simplement qu'il commande l'exécution de chaque fichier du répertoire **/etc/rc*n*.d** (n étant le numéro du niveau d'exécution) avec le paramètre **start** si la première lettre du nom du fichier est un **S**, et avec le paramètre **stop** si la première lettre du nom du fichier est un **K**. Chacun des fichiers de **/etc/rc*n*.d** est un lien symbolique vers un script de lancement de service de **/etc/init.d** et cette construction permet de dire quels services doivent être démarrés ou arrêtés pour chacun des niveaux d'exécution.

Selon les distributions, il se peut que les scripts rc et les répertoires rcn.d soient placés à des emplacements différents. La cohérence est assurée par la bonne gestion des chemins dans les scripts systèmes et la création de liens symboliques quand c'est nécessaire.

Ces liens peuvent être créés manuellement avec la commande **ln**.

Création de liens de gestion de services avec la commande ln

Ces liens doivent être créés pour chacun des niveaux d'exécutions possibles.

```
cd /etc/rcx.d
ln -s ../init.d/service Cnnservice
```

Lien de lancement de services : paramètres	
x	Le niveau d'exécution pour lequel on veut gérer le démarrage ou l'arrêt du service.
C	Commutateur de démarrage (S) ou d'arrêt (K).
nn	Numéro d'ordre à deux chiffres. Le script sera géré plus ou moins tôt par rapport aux autres du même service.
service	Nom du service à gérer.

e. Gestion des niveaux d'exécution

La commande **runlevel** indique le niveau d'exécution en cours.

Affichage du niveau d'exécution

```
runlevel
```

La commande **telinit** permet de changer à chaud le niveau d'exécution d'un système.

Changement de niveau d'exécution

```
telinit niveau
```

Où *niveau* représente le niveau d'exécution dans lequel on souhaite placer le système.

Gestion du niveau d'exécution

Le changement à chaud de niveau d'exécution ne devrait être réalisé que sur un système dont on connaît la configuration.

```
alpha:~# runlevel
N 2
alpha:~# telinit 3
alpha:~#
alpha:~# runlevel
N 3
alpha:~#
```

Ponctuellement, le niveau d'exécution à charger peut aussi être fourni au noyau en tant que paramètre lors de son chargement. Le choix du niveau d'exécution peut donc aussi se faire depuis le gestionnaire de démarrage en plaçant simplement le niveau souhaité sur la ligne de chargement du noyau.

f. Commandes de gestions des liens de services

Les commandes **update-rc.d** et **chkconfig** permettent de s'affranchir de la gestion contraignante des liens d'appels de services selon les niveaux d'exécution. Les deux commandes ne sont pas disponibles sur tous les systèmes, et il se peut même que la création manuelle de liens soit la seule solution fonctionnelle. Dans tous les cas, il est pédagogiquement intéressant de vérifier l'action de ces commandes sur les liens en place dans les répertoires **/etc/rc*n*.d**.

Création des liens de gestion de services

```
update-rc.d service defaults
```

```
chkconfig --add service
```

Où *service* représente le nom du service présent dans le répertoire **/etc/init.d**. Le paramètre defaults implique que le service sera démarré dans les niveaux fonctionnels par défaut, et arrêté dans les niveaux non fonctionnels (0 pour système arrêté, 1 pour le mode maintenance, et 6 pour une machine en cours de redémarrage).

Suppression des liens de gestion de services

```
update-rc.d service remove
```

```
chkconfig --del service
```

Vérification des états d'un service selon les niveaux

```
chkconfig --list service
```

Exemple d'utilisation de la commande chkconfig

La commande chkconfig permet aussi bien la création de liens que la visualisation des services selon les niveaux d'exécution.

```
[root@beta ~]# ls /etc/rc5.d/*nfs
ls: /etc/rc5.d/*nfs: No such file or directory
[root@beta ~]# chkconfig --add nfs
[root@beta ~]# ls /etc/rc5.d/*nfs
/etc/rc5.d/K20nfs
```

```
[root@beta ~]# chkconfig --list nfs
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@beta ~]#
```

Exemple d'utilisation de la commande update-rc.d

```
alpha:/etc/init.d# ls /etc/rc2.d/*cron
ls: ne peut accéder /etc/rc2.d/*cron: Aucun fichier ou répertoire de ce type
alpha:/etc/init.d# update-rc.d cron defaults
 Adding system startup for /etc/init.d/cron ...
   /etc/rc0.d/K20cron -> ../init.d/cron
   /etc/rc1.d/K20cron -> ../init.d/cron
   /etc/rc6.d/K20cron -> ../init.d/cron
   /etc/rc2.d/S20cron -> ../init.d/cron
   /etc/rc3.d/S20cron -> ../init.d/cron
   /etc/rc4.d/S20cron -> ../init.d/cron
   /etc/rc5.d/S20cron -> ../init.d/cron
alpha:/etc/init.d# ls /etc/rc2.d/*cron
/etc/rc2.d/S20cron
alpha:/etc/init.d#
```

g. Script indépendant du niveau d'exécution : rc.local

Une fois tous les scripts liés au niveau courant exécutés, un dernier script : **rc.local** est exécuté.

Script rc.local sur une distribution ubuntu

Prêt à servir...

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
```

Un script /etc/rc.boot peut se rencontrer sur certains systèmes anciens. Il est également appelé par le processus init.

3. Utilisation des niveaux d'exécution

Quelle que soit la distribution Linux, l'administrateur a toujours à sa disposition des niveaux d'exécution disponibles non utilisés par défaut. Bien entendu, il ne sert à rien de configurer des niveaux d'exécution pour le plaisir. Dans l'immense majorité des cas, le système prévoit un niveau fonctionnel par défaut, et tout le fonctionnement en production va se faire au sein de ce niveau. Dans quelques cas particuliers toutefois, l'administrateur peut choisir de configurer certains niveaux pour des besoins fonctionnels particuliers, et chaque niveau d'exécution correspondra à un mode de fonctionnement du serveur, avec tout ou partie des services démarrés.

En jouant sur les liens contenus dans les répertoires rc*n*.d, et en remplaçant le K de la première lettre par un S ou inversement, on provoque, pour le niveau d'exécution donné, le démarrage ou l'arrêt du service. Ainsi, si un service donné est appelé par un lien K en niveau 3 et un lien S en niveau 4, l'administrateur pourra en démarrant son système dans un de ces deux niveaux choisir le niveau fonctionnel du système.

On peut se demander quelle est l'importance du numéro d'ordre situé derrière le S ou le K. Les scripts sont traités dans l'ordre où le shell les présente, et ce sont les caractères alphanumériques du nom du lien qui déterminent l'ordre

de lancement ou d'arrêt des scripts. La seule contrainte pour l'affectation de ce numéro est donc le moment auquel le script doit être lancé. Si un service est dépendant d'un autre, le script à lancer en dernier doit alors avoir un numéro d'ordre supérieur au premier.

Les niveaux d'exécution n'étant plus guère utilisés en tant qu'outils d'administration, les arrêts de services sont souvent mal gérés par défaut. Il convient si on souhaite utiliser les niveaux d'exécution comme éléments de gestion d'un système d'inventorier précisément quels sont les services qui doivent démarrer et quels sont les services qui doivent s'arrêter à chaque changement.
