

# Gérer les services lancés au démarrage

## 1. Introduction

A la différence des distributions telles que RedHat ou Mandrake, la distribution Debian GNU/Linux n'utilise pas de fichier *rc.local*. Ce fichier permet la gestion des scripts de démarrage. La commande *update-rc.d* provenant du paquet *sysv-rc* spécifique à Debian remplace avantageusement ce script.

## 2. Comment s'effectue le lancement des services au démarrage et à l'arrêt de votre Debian GNU/Linux ?

### 2.1. Services et niveaux de démarrage

La commande *init* a pour rôle de démarrer vos *services* par l'intermédiaire du fichier de configuration */etc/inittab*.



#### Définition d'un service

Un *service* (ou *daemon* en anglais) est une application qui s'exécute en tâche de fond pendant toute la durée d'utilisateur de votre machine.

On assimile souvent un *service* à un serveur.

Le serveur web *Apache* utilise par exemple un service nommé *apached* pour *Apache Daemon*. Ce service traite les requêtes qui arrivent sur la port HTTP de la machine sur laquelle Apache est installé pour ensuite renvoyer à l'expéditeur la page Web que ce dernier a demandé.

Le fichier */etc/inittab* permet de définir quels scripts sont lancés au démarrage ainsi que le niveau de démarrage par défaut de votre Debian GNU/Linux



#### Définition d'un niveau de démarrage

Un *niveau de démarrage* (ou *runlevel* en anglais) correspond un état de votre Debian GNU/Linux. Cet état est défini par un certains nombre de service qui s'exécutent dans cet état.

Il existe par convention 7 niveaux de démarrage :

- *0* : arrêt du système (halt)
- *1* : mode utilisateur unique (single user). Ce mode est utilisé généralement si vous rencontrez des problèmes avec certains services lancés au démarrage car le minimum de services sont démarrés et vous pouvez seulement vous identifier en tant qu'utilisateur root.
- *2 à 5* : mode multi-utilisateur car sous Debian tous ces niveaux sont identiques. Ce mode vous permet de vous identifier en n'importe quel utilisateur.
- *6* : redémarrage (reboot)

L'environnement d'un niveau de démarrage est défini par les liens présents dans le répertoire `/etc/rcX.d` où `X` représente le niveau de démarrage concerné. Les scripts quand à eux se situent dans le répertoire `/etc/init.d`.

## 2.2. Changer de niveau de démarrage

Le changement du niveau de démarrage peut s'effectuer uniquement le super-utilisateur (root) grâce à la commande `init`.

Pour passer par exemple du niveau de démarrage courant au niveau de démarrage numéro 1 il suffit de taper :

```
# init 1
```

Le niveau de démarrage par défaut est défini dans les premières lignes du fichier `/etc/inittab`. Dans la version présente en annexe de cet article le niveau de démarrage par défaut est le niveau numéro 2 comme on peut le voir sur la ligne ci-dessous :

```
id:2:initdefault:
```

## 2.3. Lancement automatique des services au démarrage de votre Debian GNU/Linux

Soit un service `mon_script`. Pour que ce service se lance au niveaux de démarrage 2 et 6 il faut faire des liens symboliques dans les répertoires `/etc/rcX.d` correspondant vers notre script placé dans le répertoire `/etc/init.d`. Pour cela :

```
# ln -s /etc/init.d/mon_script /etc/rc2.d/S20mon_script
# ln -s /etc/init.d/mon_script /etc/rc6.d/S20mon_script
```

Ces liens sont nommés selon une méthode très précise : une lettre (K ou S) suivi d'un chiffre (20 dans cet exemple) puis du nom du script.

- Le `S` signifie que le script doit être démarré (l'argument passé au script sera `start`) alors que le `K` signifie que le script doit être arrêté (l'argument passé au script sera `stop`).
- Le chiffre définit la priorité de lancement et d'arrêt des scripts. On lance les script de démarrage et d'arrêt par ordre de priorité en commençant par le script ayant la priorité la plus faible jusqu'au script ayant la priorité la plus haute.
- Par exemple `S20mon_script` sera démarré après le script `S10un_autre_script`, et le script `K20mon_script` sera arrêté après le script `K12encore_un_autre_script`.

## 2.4. Lancement manuel des services

Les scripts doivent être fait de tel sorte que l'on puisse les démarrer et les arrêter manuellement à l'aide des paramètres `start` et `stop`.

Ainsi pour démarrer manuellement notre script on devrait taper :

```
# /etc/init.d/mon_script start
```

Et pour l'arrêter on devrait saisir :

```
# /etc/init.d/mon_script stop
```

### 3. Création d'un script de démarrage

Un script de démarrage a un minimum de fonctions à remplir si l'on veut qu'il soit correctement utilisé par le système. Le script suivant est suffisamment commenté pour vous permettre d'en comprendre le contenu. Vous aurez toutefois besoin de connaître les bases de la programmation par script.

#### Programmation de script

Dans notre exemple nous avons utilisé le langage de l'environnement *bash*. Pour plus d'information sur ce puissant environnement reportez vous à la page du manuel d'utilisation de votre Debian GNU/Linux grâce à la commande :

```
$ man bash
```

Voici un exemple commenté de script de démarrage :

```
#!/bin/sh

## Script basique donné à titre d'exemple pour montrer le fonctionnement
## d'un script de démarrage.

## /etc/init.d/apache exemple: démarre et arrête le service apache

## On définit la valeur de quelques variables avant de commencer le script
proprement dit.
# Répertoires contenant de nombreux programmes exécutables
PATH=/bin:/usr/bin:/sbin:/usr/sbin
# Chemin vers le programme qui démarrera Apache
DAEMON=/usr/sbin/apache
# Optionnel : définit le nom et la description qui s'afficheront lors du
démarrage ou de
# l'arrêt du script
NAME=apache
DESC="Webserver"

## Début du script proprement dit
# On vérifie tout d'abord que le programme apachectl est exécutable, sinon
fin du script
test -x $DAEMON || exit 0

# Suivant l'argument que l'on va ajouter après le nom du script, on va
définir différentes
# possibilités aux script grâce à l'instruction case
case "$1" in
    # si on tape # /etc/init.d/apache_exemple start, les instructions qui
suivent vont être
    # exécutées
    start)
        # Affiche : « Starting Webserver: apache »
        # $DESC et $NAME sont bien entendu les variables que l'on a défini
plus haut
```

```

    echo -n "Starting $DESC: $NAME"
    # Lance le serveur web Apache
    apache & >& /dev/null
    echo "."
    ;;
# Fin des instructions si on met l'argument start après le nom du
script
# si on tape # /etc/init.d/apache_exemple stop, les instructions
suivantes vont être
# exécutées
stop)
    # Affiche : « Stopping Webserver: apache »
    echo -n "Stopping $DESC: $NAME"
    # Tue tous les processus portant le nom « apache » sauvagement (« -
9 ») et n'affiche
    # pas le résultat (« /dev/null »)
    killall -9 apache >& /dev/null
    echo "."
    ;;
# Fin des instructions si on met l'argument stop après le nom du script
# si on tape /etc/init.d/apache_exemple restart, les instructions
suivantes vont être
# exécutées
restart)
    # Affiche : « Restarting Webserver: apache»
    echo -n "Restarting $DESC: $NAME"
    # Vérifie si le processus apache est déjà lancé ou non
    if [ -z "$(ps ax | egrep apache)" ]; then
    # Si il n'est pas lancé alors on affiche : « Apache isn't running,
so not killed »
        echo " Apache isn't running, so not killed" ;
    else
    # Sinon on tue tous les processus portant le nom apache comme
décrit précédemment
        killall -9 apache >& /dev/null ;
    fi
    # Attente d'une seconde avant de continuer le script
    sleep 1
    # Enfin on démarre le serveur apache
    apache >& /dev/null
    echo "."
    ;;
# Si on tape # /etc/init.d/apache alors le script affichera les
différents arguments que
# l'on peut donner au script
*)
    echo "Usage: /etc/init.d/$NAME start|stop|restart" >&2
    exit 1
    ;;
esac

exit 0
# Fin du script !

```

Une fois que vous avez enregistré ce fichier dans le répertoire */etc/init.d* sous le nom *apache\_exemple*, il ne vous reste plus qu'à le rendre exécutable. Pour cela :

```
# chmod 755 /etc/init.d/apache_exemple
```

Si vous avez installé le serveur Apache installé sur votre Debian GNU/Linux vous pouvez vérifier que votre script fonctionne avec les commandes suivantes :

```
# /etc/init.d/apache_exemple start
# /etc/init.d/apache_exemple stop
```

## 4. Installation de liens vers les scripts de démarrage

Maintenant que notre script est créé nous devons le configurer pour qu'il s'exécute dans le niveau de démarrage voulu. Pour cela nous allons utiliser et décrire le fonctionnement de la commande *update-rc.d*.

Si vous voulez exécuter notre script *apache\_exemple* dans les niveaux de démarrage par défaut avec la priorité 20 :

```
# update-rc.d apache_exemple defaults 20
```

Ce qui produit le résultat suivant dans votre terminal :

```
Adding system startup for /etc/init.d/apache_exemple ...
/etc/rc0.d/K19apache_exemple -> ../init.d/apache_exemple
/etc/rc1.d/K19apache_exemple -> ../init.d/apache_exemple
/etc/rc6.d/K19apache_exemple -> ../init.d/apache_exemple
/etc/rc2.d/S19apache_exemple -> ../init.d/apache_exemple
/etc/rc3.d/S19apache_exemple -> ../init.d/apache_exemple
/etc/rc4.d/S19apache_exemple -> ../init.d/apache_exemple
/etc/rc5.d/S19apache_exemple -> ../init.d/apache_exemple
```

On constate ainsi que les liens sont tous créés avec la lettre S pour les niveaux de démarrage standard du système et K pour les autres.

Si vous voulez définir un niveau de démarrage et un niveau d'arrêt pour votre script vous pouvez utiliser la commande suivante :

```
# update-rc.d apache_exemple start 20 5 . stop 40 6 .
```

Ce qui produit le résultat suivant dans votre terminal :

```
Adding system startup for /etc/init.d/apache_exemple ...
/etc/rc6.d/K40apache_exemple -> ../init.d/apache_exemple
/etc/rc5.d/S10apache_exemple -> ../init.d/apache_exemple
```

De cette autre façon les liens sont créés uniquement pour les niveaux de démarrage 5 et 6. On a assigné des priorités différentes selon le niveau de démarrage. Le point à la fin de ligne est très important car il marque la fin de votre commande.

## 5. Désinstallation de liens vers les scripts de démarrage

Si vous souhaitez désinstaller le script précédent afin qu'il ne soit plus lancé au démarrage. Deux cas de figure se présentent à vous :

- Vous n'avez plus besoin du script donc vous supprimez le script du répertoire */etc/init.d*.
- Sinon vous gardez ce script afin de pouvoir éventuellement le réutiliser plus tard.

Dans le premier cas, il vous suffit de taper :

```
# rm /etc/init.d/apache_exemple
# update-rc.d apache_exemple remove
```

Dans le second cas vous devez seulement saisir :

```
# update-rc.d apache_exemple remove
```

## 6. Annexe : fichier */etc/inittab*

Voici un exemple de fichier */etc/inittab*

```
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $

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

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
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

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

# Action on special keypress (ALT-UpArrow).
#kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab to let this
work."

# What to do when the power fails/returns.
```

```
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
# <id>:<runlevels>:<action>:<process>
#
# Note that on most Debian systems tty7 is used by the X Window System,
# so if you want to add more getty's go ahead but skip tty7 if you run X.
#
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
```