

# Gérer les services lancés au démarrage

- [Introduction](#)
- [Comment sont lancés les services au démarrage et à l'arrêt de votre système](#)
- [Création d'un script de démarrage](#)
- [Installation de liens vers les scripts de démarrage](#)
- [Désinstallation de liens vers les scripts de démarrage](#)
- [Pour en savoir plus](#)
- [Remerciements](#)
- [Modifications du document :](#)

Article écrit par l'équipe d'Andesi et wikisé par Pierre Chignac.

## Introduction

A la différence des distributions telles que RedHat ou Mandrake, la distribution Debian n'utilise pas de fichier rc.local permettant de faire des liens vers des scripts de démarrage. La commande update-rc.d provenant du paquet sysv-rc est spécifique à Debian et remplace avantageusement ce script. La page de manuel de update-rc.d (**# man update-rc.d**) décrit assez bien la fonction de cet utilitaire : **update-rc.d met à jour automatiquement les liens vers les scripts d'initialisation de type System-V**. Pour le néophyte cela signifie que cette commande permet de gérer les liens vers les scripts de démarrage, si cela vous paraît un peu confus, c'est normal à ce stade, la suite de cet article devrait éclaircir cette explication.

## Comment sont lancés les services au démarrage et à l'arrêt de votre système

Init a pour rôle de démarrer vos services (comme par exemple gdm qui est un prompt graphique) par l'intermédiaire du fichier de configuration */etc/inittab* qui contient les scripts à démarrer. Ce fichier permet de définir un runlevel (ou niveau de démarrage) qui permettra seulement l'exécution d'un groupe de services. Il existe par convention 6 runlevels. Les runlevels 0 et 6 sont réservés respectivement à l'arrêt et au redémarrage du système, le niveau 1 permet de démarrer le mode utilisateur unique.

Ainsi les liens vers les scripts de démarrage se trouvent dans rcX.d où X représente le niveau de démarrage (runlevel) tandis que les scripts proprement dit se situent dans */etc/init.d*. Les 7 niveaux de démarrage sont donc :

- **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)

Le changement du niveau de démarrage peut être effectué par root avec la commande init en tapant par exemple : **# init 1**. Ce niveau de démarrage est défini dans le fichier */etc/inittab* par une ligne comme celle-ci où 2 représente le niveau de démarrage actuel :

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

Maintenant que nous avons défini les différents niveaux de démarrage et leur rôle, nous allons désormais expliquer de quelle manière se lancent les services. Pour qu'un service, que nous appellerons exemple, se lance au niveau de démarrage 0, 2 et 6 sans passer par la commande `update-rc.d`, il faut faire des liens symboliques vers le script de démarrage se trouvant dans `/etc/init.d` :

```
# ln -s /etc/init.d/exemple /etc/rc1.d/S20exemple
```

```
# ln -s /etc/init.d/exemple /etc/rc2.d/S20exemple
```

Nous pouvons également démarrer et arrêter manuellement un service en tapant respectivement :

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

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

Ces liens sont nommés selon une méthode précise : une lettre (K ou S suivant le niveau de démarrage) suivie d'un chiffre (20 dans cet exemple) et enfin du nom de votre script (exempleici). Le S signifie que le script doit être lancé (argument passé au script : start) alors que le K signifie que le script doit être arrêté (argument passé au script : stop). Enfin le chiffre définit l'ordre de lancement et d'arrêt des scripts, ainsi S20exemple sera démarré après S10syslogd, cependant K20exemple sera arrêté après K10syslogd.

Passons désormais à la pratique en écrivant un script de démarrage et en précisant le fonctionnement de la commande `update-rc.d` afin d'éviter la création des liens assez fastidieuse. Vous pouvez également utiliser des outils remplissant la même fonction que **update-rc.d** en mode graphique tel que `ksysv` (paquet du même nom).

## Création d'un script de démarrage

L'écriture d'un script de démarrage se révèle en fait plus simple que vous ne le pensez. Vous devez néanmoins posséder quelques bases en programmation shell. Pour obtenir plus d'informations consultez la page de manuel de votre shell (pour bash qui est le shell par défaut sur Debian : **# man bash**).

Voici un exemple commenté de script de démarrage possible :

```
#!/bin/sh

## Script basique donné à titre d'exemple et destiné à montrer
## comment fonctionne un script de démarrage.
## On pourrait utiliser la commande start-stop-daemon comme dans
## le script initial d'apache mais cela sortirait du cadre de cet
## article ;).
## On précise tout d'abord que c'est un script écrit en shell
## avec la ligne ci-dessus

## /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 au 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, vous pouvez procéder de deux manières qui aboutissent exactement au même résultat dans ce cas-là :

```
# chmod +x /etc/init.d/apache_exemple
```

ou

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

Vous pouvez désormais vérifier que votre script fonctionne avec les commandes suivantes :

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

```
# /etc/init.d/apache_exemple restart
```

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

Nous allons ensuite décrire le fonctionnement de la commande `update-rc.d` pour que votre script soit lancé lors du démarrage ou de l'arrêt du système.

## Installation de liens vers les scripts de démarrage

Imaginons que vous vouliez exécuter notre script `apache_exemple` au démarrage dans un runlevel particulier tel que décrit plus haut.

Commençons par un exemple simple qui devrait convenir à la plupart des utilisateurs :

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

```
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
```

De cette façon les liens sont tous créés avec la lettre S pour les niveaux de démarrage standard du système et K pour les autres, ainsi qu'avec le chiffre 20 quel que soit le niveau de démarrage.

Un autre exemple d'utilisation, cette fois-ci un peu plus poussé :

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

```
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 5 et 6 de démarrage. De plus on a assigné des numéros différents suivant le runlevel. Le point à la fin de ligne est très important car il marque la fin de votre commande.

Enfin un dernier exemple dans lequel le script `apache_exemple` sera pas lié dans les répertoires destinés à l'arrêt du système (notez le point à la fin de la commande) :

```
# update-rc.d apache_exemple start 20 2 3 4 5 .
```

```
Adding system startup for /etc/init.d/apache_exemple...
```

```
/etc/rc2.d/S20apache_exemple-> ../init.d/apache_exemple
```

```
/etc/rc3.d/S20apache_exemple-> ../init.d/apache_exemple
```

```
/etc/rc4.d/S20apache_exemple-> ../init.d/apache_exemple
```

```
/etc/rc5.d/S20apache_exemple-> ../init.d/apache_exemple
```

Vérifiez bien que les liens sont bien placés et grâce à cette magnifique commande qu'est `update-rc.d` votre script sera lancé au prochain redémarrage.

## Désinstallation de liens vers les scripts de démarrage

Imaginons maintenant que vous voulez 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 taperez alors :

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

## Pour en savoir plus

Si vous voulez en apprendre davantage sur la commande **update-rc.d** et sur le système `init`, les pages de manuel sont très utiles surtout si vous les installez en français (**# apt-get install manpages-fr**) :

```
# man update-rc.d
```

```
# man init
```

## Remerciements

Cet article a été écrit avec l'aide de visiteurs du site et du forum et modifié par moi même (Arnaud Fontaine). Voici les pseudo des personnes qui ont participé à cet article :

- kagou
- Stéphane
- crayongomme
- kakaouete
- y0m
- ptitlouis
- Fusco
- ioguix
- peyo
- pathe