

Contenu

Lancer un script au boot du Raspberry Pi	3
Avant tout, le script	3
Activer le script	4
Oui mais si je dois interagir avec mon service ?	4
update-rc.d Cheat Sheet	5
What services are available for startup?	5
Start a service	7
Stop a service	7
Check the status of a service	7
Remove a service	7
Add a service	7
Setting Start and Kill priority	8
Manually set the RunLevel to Start and Kill a service	8
Run script at boot	9
Run script at boot	9
Re: Run script at boot	9
Re: Run script at boot	10
Re: Run script at boot	12
Re: Run script at boot	12
Re: Run script at boot	13
Re: Run script at boot	14
Re: Run script at boot	15
Re: Run script at boot	15
Re: Run script at boot	15
Re: Run script at boot	16
Re: Run script at boot	16
Re: Run script at boot	17
Re: Run script at boot	17
Re: Run script at boot	17

Exécution d'une application lors du démarrage.....	18
Exécution d'une application lors du démarrage.....	18
#2 17-01-2017 13:43:15	18
Re : Exécution d'une application lors du démarrage.....	18
#3 17-01-2017 14:22:12	19
Re : Exécution d'une application lors du démarrage.....	19
#4 17-01-2017 14:28:46	19
Re : Exécution d'une application lors du démarrage.....	19
#5 20-01-2017 15:45:30	19
Re : Exécution d'une application lors du démarrage.....	19
#6 20-01-2017 20:35:13	21
Re : Exécution d'une application lors du démarrage.....	21

Lancer un script au boot du Raspberry Pi

<https://www.pihomeserver.fr/2013/05/27/raspberry-pi-home-server-lancer-un-programme-automatiquement-au-demarrage/>

par [Chris](#) · Publié 27 mai 2013 · Mis à jour 12 mars 2015

Il arrive que lors de la création d'un serveur, on souhaite exécuter certains programmes dès le démarrage de l'OS. Voici comment faire sous Raspbian.

Lorsque vous installez un outil, généralement celui-ci va vous fournir le(s) script(s) pour être exécuté dès le démarrage de la machine. C'est le cas de Nginx, php-fpm, apache, mysql, etc.

Avant tout, le script

Mais si vous voulez faire un script vous même ou alors que l'outil est multi plateforme, ça se complique. Par exemple, si vous installez un serveur Minecraft (merci Pierre pour l'idée d'article).

Vous devez créer un fichier dans le répertoire */etc/init.d*. Ce fichier XYZ aura comme contenu de base :

```
#!/bin/bash

# Apres cette ligne les commandes seront executees systematiquement

# ...

# Apres cette ligne les commandes executees en fonction du parametre passe
en ligne de commande

case "$1" in

    start)

        # Commandes executees avec le parametre start (celui lors du boot)

        ;;

    stop)

        # Commandes executees avec le parametre stop (celui lors de l'arret
du systeme)

        ;;

    reload|restart)

        $0 stop

        $0 start

        ;;

    *)
```

```
    echo "Usage: $0 start|stop|restart|reload"

    exit 1

esac

exit 0
```

N'oubliez pas de rendre le script exécutable avec la commande *chmod* :

```
chmod 755 /etc/init.d/XYZ
```

Vous pouvez ensuite tester votre script avec la commande *service*. Vous passez en paramètre l'argument que vous souhaitez. Par exemple pour démarrer votre service :

```
service XYZ start
```

Activer le script

Si ça fonctionne comme vous voulez, vous pouvez alors activer le lancement du script au démarrage du système avec la commande *update-rc.d* :

```
update-rc.d XYZ defaults
```

Cette commande ajoute le script dans la séquence de démarrage de votre Raspberry. Vous pouvez utiliser d'autres valeurs à la place de *defaults* :

- *remove* : pour supprimer le script de la séquence de démarrage
- *disable* : pour désactiver le script
- *enable* : pour activer le script

D'autres paramètres permettent de régler la séquence de démarrage et les priorités du scripts en modifiant la configuration LSB (plus d'infos dans [ce post](#)).

Oui mais si je dois interagir avec mon service ?

Vous avez maintenant toutes les informations pour pouvoir créer vous même un service qui se lance au démarrage de votre Raspberry. Toutefois, il y a une situation particulière qui peut poser problème : comment faire si mon programme ne rend pas la main et est interactif ?

La commande magique c'est [screen](#). Cette commande va vous permettre de lancer une commande en la détachant de votre terminal mais de pouvoir continuer à interagir avec cette commande.

Par exemple :

```
screen -dmS ma_session /chemin/vers/mon/script
```

Cette commande va lancer le script dans une session appelée *ma_session*. Le programme s'exécute et fonctionne normalement mais en arrière plan.

Pour interagir avec vous pouvez toujours utiliser la commande *screen* :

```
screen -p 0 -S ma_session -X eval 'stuff "ma_commande_a_executer"15'
```

En activant la fenêtre par défaut de l'application attachée à *ma_session*, je demande d'exécuter la commande *ma_commande_a_executer*. Cette commande pourrait être mise dans le script XYZ (dans */etc/init.d*) avec un paramètre particulier, par exemple *do_my_cmd* et sera alors invoquée avec :

```
service XYZ do_my_cmd
```

update-rc.d Cheat Sheet

[update-rc.d Cheat Sheet | JamesCoyle.net Limited](#)

9-Nov-2013

Debian and Ubuntu use the **service** command to control services and **update-rc.d** for adding and removing services from start up. Using the **service** command we can start, stop, restart and display all available services. With **update-rc.d** we can add and remove services and add them to the Ubuntu/ Debian start up scripts. As Linux operating systems have multiple states, or runlevels, you need to make sure you add any new services to the correct runlevels. For example, you would not want to start a web service application before starting networking.

See my [post on runlevels](#) for more information about runlevels in Linux..

What services are available for startup?

Use the **status-all** switch to list all services which are registered with the OS and issues them a **status** command. You will then get one of the following displayed next to each service:

- **[+]** – Services with this sign are currently running.
- **[-]** – Services with this sign are not currently running..
- **[?]** – Services that do not have a **status** switch.

```
1 service --status-all
```

Sample output:

```
1 service --status-all
2 [ ? ] acpid
3 [ - ] apparmor
4 [ ? ] apport
5 [ ? ] atd
6 [ - ] bootlogd
7 [ ? ] console-setup
```

8 [?] cron
9 [?] dbus
10 [?] dmesg
11 [?] dns-clean
12 [?] friendly-recovery
13 [?] glusterfs-server
14 [-] grub-common
15 [?] gssd
16 [?] hostname
17 [?] hwclock
18 [?] hwclock-save
19 [?] idmapd
20 [?] irqbalance
21 [?] killprocs
22 [?] module-init-tools
23 [?] network-interface
24 [?] network-interface-container
25 [?] network-interface-security
26 [?] networking
27 [?] ondemand
28 [?] passwd
29 [?] plymouth
30 [?] plymouth-log
31 [?] plymouth-ready
32 [?] plymouth-splash
33 [?] plymouth-stop
34 [?] plymouth-upstart-bridge
35 [?] portmap
36 [?] portmap-wait
37 [?] pppd-dns
38 [?] procps
39 [?] rc.local
40 [?] resolvconf
41 [?] rpcbind-boot
42 [-] rsync
43 [?] rsyslog
44 [?] screen-cleanup
45 [?] sendsigs
46 [?] setvtrgb
47 [+] ssh
48 [?] statd
49 [?] statd-mounting
50 [-] stop-bootlogd
51 [-] stop-bootlogd-single
52 [?] sudo
53 [?] udev
54 [?] udev-fallback-graphics
55 [?] udev-finish
56 [?] udevmonitor
57 [?] udevtrigger
58 [?] ufw

```
59 [ ? ] umountfs
60 [ ? ] umountnfs.sh
61 [ ? ] umountroot
62 [ - ] unattended-upgrades
63 [ - ] urandom
64 [ ? ] whoopsie
```

Start a service

Starting a service is done using the command **service** followed by the service name and the command to start the service.

```
1 service apache2 start
```

Stop a service

Use the **stop** keyword with **service** to stop a service.

```
1 service apache2 stop
```

Check the status of a service

Each service has a status, usually running or not running. Some services, such as **network**, may have a different output and output more information on the service.

```
1 service apache2 status
```

Remove a service

Use the **remove** keyword with **update-rc.d** to remove the service start up command for an application. You will need to use the **-f** switch if the applications **/etc/init.d** start up file exists.

```
1 update-rc.d -f apache2 remove
```

Add a service

Adding a service to Ubuntu or Debian is done with the **update-rc.d** command. You can specify which runlevels to start and stop the new service or accept the defaults. The **init.d** file will be added to the relevant **rc.d** startup folders.

```
1 update-rc.d apache2 defaults
```

Setting Start and Kill priority

If you need more control over when your service is asked to start and stop, you may need to set the **start** and **kill** (S and K) values.

For a given [runlevel](#), you may have several services starting. For example, you may have `apache2` and `mysql` both starting on runlevel 3. Ideally, you'd want `mysql` to start before `apache2` and shutdown after `apache2`. In this case we need to give `mysql` the priority in startup, but `apache2` the priority in shutdown.

When starting, the lower the number, the earlier it will start. A start priority of 10 will start before a priority of 20. When killing, it's the opposite. A higher number will be killed before a lower number.

To set the start and kill priority we simply add them to the above **update-rc.d** command with the start priority first, followed by the kill priority.

```
1 update-rc.d apache2 defaults [START] [KILL]
```

The below command will start **mysql** first, then **apache2**. On shutdown, the kill will be the reverse of the start with **apache2** being killed first and **mysql** second.

```
1 update-rc.d apache2 defaults 90 90
```

```
2 update-rc.d mysql defaults 10 10
```

Because, in our example, both start and kill priorities are the same we can shorten the command to just

```
1 update-rc.d apache2 defaults 90
```

```
2 update-rc.d mysql defaults 10
```

Manually set the RunLevel to Start and Kill a service

You can manually specify the Linux RunLevel that the system must be in to Start and Kill your service. See my other blog post for more information on [RunLevels](#).

To extend the above example, we can specify exactly which RunLevel **apache2** will be started and stopped.

```
1 update-rc.d apache2 start 10 2 3 4 5 . stop 90 0 1 6 .
```

apache2 will be started (as long as it isn't already) when the system enters RunLevel **2, 3, 4** or **5** with a priority of **10**. It will then be asked to stop when the system enters RunLevel **0, 1** or **6** with a priority of **90**.

Run script at boot

<https://www.raspberrypi.org/forums/viewtopic.php?t=70520>

25/02/2014

[Run script at boot](#)

- [Quote](#)

[Tue Feb 25, 2014 2:40 pm](#)

Hi, I have a script file and I would like that that script was executed in every single boot, how can I do that?

I searched and nothing worked... I'm new in the rasp and linux, so I don't know much...

Thank you



[lorenz92](#)

[Re: Run script at boot](#)

- [Quote](#)

[Tue Feb 25, 2014 4:05 pm](#)

There are actually several ways for doing that.

One way I used in a project of mine (which I prefer) is by using the init run-level based rc scripts. Basically you will encounter the command "service" to start and stop system services i.e. your personal service.

Following this method you will also have the possibility of doing "service <your script> start" for example, which is neat and handy.

NOTE: the following steps usually require root permissions. So just issue every command with sudo or log into a root shell (sudo -s, beware that everything you do is done with root permissions!!)

Code: [Select all](#)

```
# update boot / reboot files
cp <your script> /etc/init.d/
# do it as soon as the device is going down,
# both for shutdown and reboot
update-rc.d /etc/init.d/<your script> defaults
(If I couldn't have been clear enough, this has to be done after looking at the following part...)
```

The <your script> part is something like the following:

Code: [Select all](#)

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          <your script name>
# Required-Start:    $all
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Manage my cool stuff
### END INIT INFO

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/bin

. /lib/init/vars.sh
. /lib/lsb/init-functions
# If you need to source some other scripts, do it here

case "$1" in
    start)
        log_begin_msg "Starting my super cool service"
# do something
        log_end_msg $?
        exit 0
        ;;
    stop)
        log_begin_msg "Stopping the coolest service ever unfortunately"

        # do something to kill the service or cleanup or nothing

        log_end_msg $?
        exit 0
        ;;
    *)
        echo "Usage: /etc/init.d/<your script> {start|stop}"
        exit 1
        ;;
esac
```

That's it 😊

Another way (but hackish) I used to do on an embedded system (but because that was not a debian, so beware, it seems more simple but an update of the system might screw it up) is to modify the /etc/profile.

[best97](#)

[Re: Run script at boot](#)

- [Quote](#)

[Tue Feb 25, 2014 8:15 pm](#)

Ok, thank you 😊, just have a couple of questions.

When I run:

Code: [Select all](#)

```
cp <your script> /etc/init.d/
```

In the <you script>, should I put my script name or the path to my script? If it is the name, does it matter if it is in the pi folder?

And I didn't get what I have to do when you say:

"The <your script> part is something like the following:

Code: [Select all](#)

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          <your script name>
# Required-Start:    $all
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Manage my cool stuff
### END INIT INFO

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/bin

. /lib/init/vars.sh
. /lib/lsb/init-functions
# If you need to source some other scripts, do it here

case "$1" in
    start)
        log_begin_msg "Starting my super cool service"
    # do something
        log_end_msg $?
        exit 0
        ;;
    stop)
        log_begin_msg "Stopping the coolest service ever unfortunately"

        # do something to kill the service or cleanup or nothing

        log_end_msg $?
        exit 0
        ;;
    *)
        echo "Usage: /etc/init.d/<your script> {start|stop}"
        exit 1
        ;;
esac
"
```

Do I have to create another file with this?

My script is very simple:

Code: [Select all](#)

```
#!/bin/sh

sudo service hostapd stop
```

```
sudo service udhcpd stop
```

```
sudo service hostapd start
```

```
sudo service udhcpd start
```

Thank you so much for helping me 😊

[lorenz92](#)

Re: Run script at boot

- [Quote](#)

[Tue Feb 25, 2014 8:23 pm](#)

Yes, <your script> is a path. If the script is in the current directory (i.e. the one at prompt) then actually you can just type its name.

Useless to say, avoid spaces ...

Sorry I was in a hurry and I did not really spent too much words 😊

So, exactly, you have to create yet another script. Just copy and paste my template.

Then replace # do something with the content of your original script, to say:

Code: [Select all](#)

```
service hostapd stop
service udhcpd stop
service hostapd start
service udhcpd start
```

NOTE: why not sudo? Simple 😊 The script, if directly executed by the system is trivially executed as root. If issued by the user, the same here, it is implicit -> sudo service myservice...

Hope this can help, otherwise just poke here your doubts.

[best97](#)

Re: Run script at boot

- [Quote](#)

[Tue Feb 25, 2014 9:09 pm](#)

Thank you, so I just edited the template, is it ok like this? (my script is called "hotspot" and it is in the pi folder)

Code: [Select all](#)

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          </home/pi/hotspot>
# Required-Start:    $all
# Required-Stop:
```


sudo is usually required when writing / entering system directories (notably /etc tree).
Let me know!

EDIT: I noticed a minor problem (actually, I don't really know if it is a problem)

Provides: </home/pi/hotspot>

it should be

Provides: hotspot

[best97](#)

Re: Run script at boot

- [Quote](#)

[Tue Feb 25, 2014 11:06 pm](#)

I changed the #Provides and then ran the commands

Code: [Select all](#)

```
cp /home/pi/hotspotboot /etc/init.d/  
update-rc.d /etc/init.d/home/pi/hotspotboot defaults
```

And it gives me an error:

Code: [Select all](#)

```
root@raspberrypi:/home/pi# cp /home/pi/hotspotboot /etc/init.d/  
root@raspberrypi:/home/pi# update-rc.d /etc/init.d/home/pi/hotspotboot  
defaults  
update-rc.d: using dependency based boot sequencing  
update-rc.d: error: unable to read  
/etc/init.d//etc/init.d/home/pi/hotspotboot
```

What should I do?

Thank you 😊

-----EDIT-----

I correct it, I did

Code: [Select all](#)

```
cp /home/pi/hotspotboot /etc/init.d/  
cd /etc/init.d/  
update-rc.d hotspotboot defaults
```

but now it gives me this error:

Code: [Select all](#)

```
oot@raspberrypi:/etc/init.d# update-rc.d hotspotboot defaultsupdate-rc.d:  
using dependency based boot sequencing  
update-rc.d: warning: default start runlevel arguments (2 3 4 5) do not  
match hotspotboot Default-Start values (      2 3 4 5)
```

update-rc.d: warning: default stop runlevel arguments (0 1 6) do not match
hotspotboot Default-Stop values (0 1 6)
insserv: script hotspotboot is not an executable regular file, skipped!
insserv: warning: script 'mathkernel' missing LSB tags and overrides
What should i do?

[rpdom](#)

[Re: Run script at boot](#)

- [Quote](#)

[Wed Feb 26, 2014 5:09 am](#)

The only line to really worry about is

Code: [Select all](#)

```
insserv: script hotspotboot is not an executable regular file, skipped!
```

This is because you haven't got the eXecute flag set on the file.

Try

Code: [Select all](#)

```
cd /etc/init.d  
sudo chmod u+x hotspotboot  
update-rc.d hotspotboot defaults
```

You will still get some warnings, but the error I mentioned above should have gone.

Last edited by [rpdom](#) on Wed Feb 26, 2014 7:37 am, edited 1 time in total.

[lorenz92](#)

[Re: Run script at boot](#)

- [Quote](#)

[Wed Feb 26, 2014 7:35 am](#)

Oh yes indeed, I forgot to tell you that...

The problem is for sure about executable bit ("+x").

[best97](#)

[Re: Run script at boot](#)

- [Quote](#)

[Wed Feb 26, 2014 8:18 am](#)

I tried that and then this appeared, i believe it is an error:

Code: [Select all](#)

```
insserv: missing valid name for `Provides:' please add.  
(just for you to know, My provides is like this: # Provides: hotspot)  
What should I do?
```

[iinnovations](#)

Re: Run script at boot

- [Quote](#)

[Sat Mar 01, 2014 5:32 pm](#)

It is much simpler to add your commands to the end of /etc/rc.local . Having a system service is nice, but an unnecessary level of complication for simple tasks.

Typically, I create a boot.sh script, invoke it from rc.local. Then I can keep boot.sh in a git repo or somewhere else handy.

As for the commands invoked here, you should probably just configure your dhcp and access point to not start on boot and turn them on when you want. I do this via a boot script as well, but using python. I have a python call read network configuration from a database and then act accordingly. Remember that you can invoke python scripts from rc.local as well.

Colin

CuPID Controls :: Open Source browser-based sensor and device control
interfaceinnovations.org/cupidcontrols.html
cupidcontrols.com

[Birintha](#)

Re: Run script at boot

- [Quote](#)

[Sun Nov 22, 2015 1:00 am](#)

what you mentioned <your script>. is it python script?
how can i change this steps to python script?

Last edited by [Birintha](#) on Sun Nov 22, 2015 1:36 am, edited 1 time in total.

[DirkS](#)

Re: Run script at boot

- [Quote](#)

[Sun Nov 22, 2015 1:06 am](#)

*Biruntha wrote:*what you mentioned <your script>. is it python script?

To which post are you referring?

Keep in mind that you're responding to a topic that's a year and a half old...

[Biruntha](#)

Re: Run script at boot

- [Quote](#)

[Sun Nov 22, 2015 1:38 am](#)

DirkS wrote:

*Biruntha wrote:*what you mentioned <your script>. is it python script?

To which post are you referring?

Keep in mind that you're responding to a topic that's a year and a half old...

second post

[fixermark](#)

Re: Run script at boot

- [Quote](#)

[Wed Jul 20, 2016 1:55 am](#)

Your script included the line

```
# Provides:                </home/pi/hotspot>
```

The < and > characters (and possibly also the forward-slash?) aren't accepted in that context.

Provides should be the name of the capability your init script is providing to the system, something more like "hwclock" or "alsa-utils". In your case,

```
# Provides:                hotspot
```

... would probably be appropriate.

Exécution d'une application lors du démarrage

<https://debian-facile.org/viewtopic.php?id=16525>

[galactic](#)

Exécution d'une application lors du démarrage

J'ai une très petite application java (25 lignes) que je veux exécuter au démarrage (ce n'est pas un service, elle s'exécute une seule fois en faisant une requête vers un serveur -donc avec le réseau opérationnel-, ensuite elle est inactive jusqu'à l'extinction de la machine). Je veux seulement qu'elle soit exécutée avant le démarrage du service tomcat7.

```
root@pcarm:/home/hubert# /opt/jdk1.7.0_60/bin/java -jar myapp.jar
```

Si je l'exécute bien en mode console, j'ai *bricolé* des liens symboliques au niveau de *rc2.d* et *rc3.d* et fait un petit script (qui utilise seulement le start)) dans *init.d* mais je n'arrive pas à avoir une exécution au démarrage.

Des pistes à explorer, des conseils? Merci.

#2 17-01-2017 13:43:15

[Croutons](#)

Re : Exécution d'une application lors du démarrage

Bonjour

Une piste peut être dans ton home avec le `~/.config/autostart/`, dans le répertoire tu créé un fichier.desktop

exemple: redshift.desktop

```
[Desktop Entry]
Encoding=UTF-8
Version=0.9.4
Type=Application
Name=redshift
Comment=
Exec=/usr/bin/redshift-config
OnlyShowIn=XFCE;
StartupNotify=false
Terminal=false
Hidden=false
```

par contre tu ne gère pas l'ordre de lancement, il faudrait peut être arrêter le service et inclure son lancement dans ton script

c'est une idée je suis pas un barbu moi 🤖

-->les cahiers du debutant<-- WikiDF-->[Découvrir les principales commandes Linux](#)<--
L'expérience, c'est le nom que chacun donne à ses erreurs. Oscar Wilde

#3 17-01-2017 14:22:12

bendia

Re : Exécution d'une application lors du démarrage

Tu es sur quelle version de Debian [Voir le tuto : Trop cool d'indiquer son installation dans son profil !](#)

Si c'est Debian8, c'est *systemd* le système d'init maintenant. A priori, il y a moyen de lancer des services de type *oneshot* ([http://doc.fedora-fr.org/wiki/Systemd#A ... oneshot.22](http://doc.fedora-fr.org/wiki/Systemd#A...oneshot.22)) et d'utiliser des dépendances pour indiquer par exemple que tomcat nécessite en premier lieu que ton script ait bien été là au préalable (genre *Requires=ton_script.service* dans le fichier de conf de tomcat, et *Before=tomcat* dans le fichier de conf de ton script).

Ceci-dit, j'ai jamais essayé encore 😊

Ben

La seule question bête, c'est celle qu'on ne pose pas.

En ligne

#4 17-01-2017 14:28:46

frei

Re : Exécution d'une application lors du démarrage

Si ton tomcat se lance seul tu dois avoir quelque chose dans ce gout là :
`/etc/init.d/tomcatX`

Edites le et regarde ce qu'il y a dedans.

En fonction de ce que tu cherches à faire il te faudra ajouter les lignes correspondantes dans les options.

Si ton app doit être exécutée systématiquement même sur un "restart" je ferai une modification du script de démarrage de Tomcat (`/usr/share/tomcatX/bin/startup.sh`)

@Croutons

Si c'est une install standard de Tomcat à l'exécution de la session c'est déjà trop tard.

"Glory. Rotting flower." John Tardy 1989

#5 20-01-2017 15:45:30

galactic

Re : Exécution d'une application lors du démarrage

De retour après quelques mises au point sur une application web qui s'exécute dans Tomcat.

Finalement pour l'autre petite application "oneshot" qui n'est pas un service mais qui doit s'exécuter au démarrage en dehors de Tomcat, j'ai traité le cas comme un service, ce qui me semblait plus

cohérent.

Donc l'application java **horloge.jar** (mise à l'heure du système) et sa lib est copiée dans */usr/local/lib/*
Et dans */etc/init.d/* j'ai créé le script horloge avec les bons droits :

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          skeleton
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Example initscript
# Description:       This file should be used to construct scripts
to be
#                   placed in /etc/init.d.
### END INIT INFO

DAEMON="/opt/jdk1.7.0_60/bin/java"
DAEMON_OPT="-jar /usr/local/lib/horloge.jar" # fichier à
exécuter
NAME="horloge.jar" # Nom du programme
PIDFILE=/var/run/$NAME.pid

PATH="/sbin:/bin:/usr/sbin:/usr/bin"
export JAVA_HOME=/opt/jdk1.7.0_60
export PATH=${PATH}:${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin

test -e $DAEMON || exit 0

. /lib/init/vars.sh
. /lib/lsb/init-functions

start_d () {
    log_daemon_msg "Start $NAME Daemon"
    start-stop-daemon --make-pidfile --start --quiet --pidfile
$PIDFILE --exec $DAEMON -- $DAEMON_OPT
}
stop_d () {
    # pas d'arrêt du daemon : le start termine son exécution;donc
seulement suppression du fichier *.pid
    log_daemon_msg "Stop $NAME Daemon"
    rm -f $PIDFILE
}

case "$1" in
    start)
        start_d
        ;;
```

```

stop)
    stop_d
    ;;
    .....

```

```

root@pcarm:/home/hubert# ls -l /etc/init.d/horloge
-rwxr--r-- 1 root root 2015 Jan 20 13:37 /etc/init.d/horloge

```

Le seul souci sur ce script a été sur l'option -jar en début de DAEMON_OPT et le -- entre les 2 variables.

et pour créer les liens :

```

root@pcarm:/home/hubert# update-rc.d horloge defaults Adding system
startup for /etc/init.d/horloge ...
/etc/rc0.d/K20horloge -> ../init.d/horloge
/etc/rc1.d/K20horloge -> ../init.d/horloge
/etc/rc6.d/K20horloge -> ../init.d/horloge
/etc/rc2.d/S20horloge -> ../init.d/horloge
/etc/rc3.d/S20horloge -> ../init.d/horloge
/etc/rc4.d/S20horloge -> ../init.d/horloge
/etc/rc5.d/S20horloge -> ../init.d/horloge

```

Le code priorité est S20 alors que pour Tomcat j'ai S92 donc, c'est très bien mon appli s'exécute avant le démarrage de Tomcat.

Mais une question, n'ayant pas la possibilité de choix du niveau de priorité, comment celui-ci est-il attribué?

Ce n'est peut-être pas très académique comme solution, mais cela fonctionne.

Question annexe : j'ai mis en début de script une init des variables d'environnement java; j'ai aussi une init dans :

```

root@pcarm:/home/hubert# cat /etc/profile.d/jdk.sh
#!/bin/sh
export JAVA_HOME=/opt/jdk1.7.0_60
export MANPATH="${MANPATH}:${JAVA_HOME}/man"
export PATH=${PATH}:${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin

```

Cet environnement est mis en place lors de l'ouverture d'un shell; mais qu'en est-il sur un serveur ou l'on ne passe pas par un shell de connexion?

[#6 20-01-2017 20:35:13](#)
[galactic](#)

Re : Exécution d'une application lors du démarrage

Petit souci, si la commande :

```
root@pcarm:/home/hubert# /etc/init.d/horloge start
* Start horloge.jar Daemon
Fri Jan 20 19:30:17 CET 2017
```

fonctionne très bien (mise à l'heure effective)

Lorsque je redémarre la machine, il n'y a pas de mise à l'heure : donc le script ne serait pas exécuté lors d'un reboot.

Il y a bien exécution du script mais pas de mise à l'heure.

Avant d'effectuer un reboot, j'ai supprimé le fichier */run/horloge.jar.pid*; et après un redémarrage, ce fichier, créé par le script, est bien présent.

Des idées?