Running a Python script at startup

This question has been asked quite a few times before but I didn't seem to get it working using the existing information.

My Pi runs Raspbian. I have a Python script named dnscheck.py which loops forever.

I need it to run at boot. I know I have to create a .sh file containing something like

```
sudo python dnscheck.py &
```

What I don't know is where this file should be or if it should contain anything else. I know about the init.d folder, but seeing the skeleton example I imagine there should be a simpler way to do this simple task.

raspbian python

asked Dec 26 '12 at 18:06

Vlad Schnakovszki
443 2 8 16

for a real easy way checkout this step by step tutorial-->youtu.be/Tvnrx-2QaUU make as much launchers as you need and adress all of them in crontab – Hossein RM Dec 28 '17 at 14:21

9 Answers

If you want to control the process with commands like start, stop, restart etc using the skelton script and altering it for your purposes might be the best option.

If you just want the process to start and that's it, you might consider just putting the command, as you wrote it in your question, into the /etc/rc.local. (I don't have my RPi at hand, but I read online that there is an 'exit 0' line in there, you should put your command above this line)

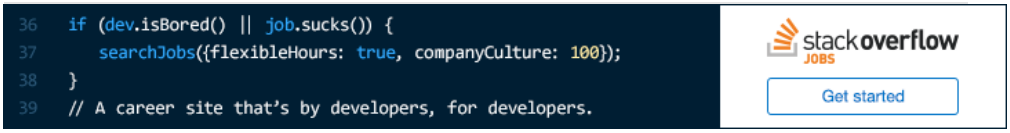
answered Dec 26 '12 at 20:42

ikku
3,884 1 17 27

It worked! I must add that setting the permissions for the script and rc.local back to 755 (read/write/execute) is a must. Not sure if both need this setting, but it worked for me. Thanks a lot for the help! – Vlad Schnakovszki Dec 27 '12 at 0:07

3 *"as you wrote it in your question"* Not quite -- you don't need sudo as `rc.local` runs root. You should also specify the complete path to the script, obviously. **You should also add `&`** at the end so that the script forks, e.g. `/path/to/foobar.py &` . – goldilocks ♦ Feb 20 '15 at 13:15

@goldilocks could you please explain why do I need to add `&` ? Because when I use without it, everything still working as expected. – Huy.PhamNhu Jul 29 '17 at 3:21



Move your script (we will save it to the file `dnscheck`) to `/etc/init.d/`, and set the permissions so it can be run:

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

Add LSB init tags to the top of your script. You will probably want to change Required-Start/Stop and the Description Tags to fit your script.

```
### BEGIN INIT INFO
# Provides:          dnscheck
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start daemon at boot time
# Description:       Enable service provided by daemon.
### END INIT INFO
```

Then create the symbolic links by running

```
update-rc.d /etc/init.d/dnscheck defaults
```

edited Feb 20 '15 at 13:24
Izzy
398 4 13

answered Dec 26 '12 at 18:55
Bert
557 4 8

Please not that while this was a decent answer 3 1/2 years ago, and will still work because Raspbian's new init system is backward compatible with LSB/SysV style scripts, new users would be better off learning to use the new system instead (*systemd*) if just adding a line to `/etc/rc.local` is insufficient. – goldilocks ♦ Jul 3 '16 at 17:32

There are many ways to do this, of course, but don't forget using `cron`. If you put a `@reboot` line in your crontab, that command will be executed on every restart.

To test, I just added the following line to my user crontab with `crontab -e`:

```
@reboot echo "$(date)" >> ~/boot.txt
```

The bonus to this method is that you can call the job as required at other intervals besides just boot time, and you don't have to edit init scripts.

answered Aug 25 '15 at 19:34
bobstro
3,046 5 19

Here is the solution that I constantly use.

Create a desktop file

```
xyz.desktop
```

type the following into it

```
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=<Application Name Goes here>
Comment=
Exec=  python /home/pi/Desktop/execute_on_boot.py
StartupNotify=false
Terminal=true
Hidden=false
```

paste this file into the

```
/home/pi/.config/autostart/
```

and restart your raspberry pi and it should automatically run your program in a new terminal

answered Aug 25 '15 at 19:07
evolutionizer
286 1 5

please don't cut and paste answers to multiple questions. If the answer is the same the newer version should be flagged as a duplicate. – Steve Robillard Aug 25 '15 at 19:29

The only reason I didnt do it was cause this page had a larger number of view as opposed to the other one. – evolutionizer Aug 25 '15 at 19:40

@SteveRobillard I also dont think I have the rep required to do so – evolutionizer Aug 25 '15 at 19:46

Flagging a post only takes 15 rep. Deciding what to do about it is the job of the moderators - so the number of views is irrelevant. Duplicate answers are automatically flagged by the system. They are a form of gaming the system. Therefore, I deleted the third one. – Steve Robillard Aug 25 '15 at 19:50

@SteveRobillard Thanks for the information and forgive my ignorance. I have there for flagged the previous question as duplicate. – evolutionizer Aug 25 '15 at 19:53

---

if you use rc.local file, this may be helpful for troubleshooting. You can add logging lines to log errors (stderr) and command output (stdout) into log file. According to this example that file saves in /tmp/rc.local.log

```
#!/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.

exec 2> /tmp/rc.local.log      # send stderr from rc.local to a log file
exec 1>&2                      # send stdout to the same log file

# Your other commands...

exit 0
```

Read more

edited May 23 '17 at 12:37
Community ♦
1

answered May 17 '17 at 16:05
Tharanga
121 3

To use a .py file, just put the line `#!/usr/bin/python` at the very start of your file. Then make it executable with `chmod +x filename` . Next, add the line `/path/to/file.py &` to `/etc/rc.local` before the `exit 0` line (swapping `/path/to/file.py` with the path to your script). This will make your python script execute at the end of boot.

edited Jul 29 '17 at 19:05

answered May 25 '15 at 15:47
An Epic Person
369 4 17

I'm really surprised djb's daemontools is not mentioned here. Daemontools does proper process supervision and you can add cool features like automatically rotated logging. TL;DR if you're not familiar with any of this, your process will be restarted every time it fails and start automatically when your Pi turns on. This is great if you wrote a bad python program that has failure modes but you don't want it to just die if an error is encountered.

Install:

> sudo apt-get install daemontools daemontools-run

Then follow steps to create daemonized processes:

- http://samliu.github.io/2017/01/10/daemontools-cheatsheet.html

It's mostly as simple as copying a `run` script into `/etc/service/<my_custom_service_name>` Another perk: you can run as any user or root! Details in the link.

FWIW I had a Pi project where I had 3 different python processes (each had an execution loop using CPU time so by using 3 processes I allowed each process to leverage 1 CPU core). Daemontools allowed me to make sure all 3 would automatically run and stay running after I plugged in the Pi.

answered Jan 13 at 20:07
Sam
111 2

Here's an even easier method that worked for me. Modify the autostart in LXDE.

Open a terminal and edit the autostart file as follows:

```
sudo nano /home/pi/.config/lxsession/LXDE-pi/autostart
```

add the following line of text to the bottom(modify the path as needed to where your example.py is located)

```
@/usr/bin/python /home/pi/example.py
```

ctr-x, and save. You may need to make the python script executable as follows:

```
sudo chmod +x /home/pi/example.py
```

reference for autostart in LXDE and reference for making python executable

edited Mar 9 '16 at 5:39

answered Mar 9 '16 at 5:25

hydronics

1 2

Note that this will not work if you don't start up in a gui (ie: run your pi headless) – Havnar Mar 9 '16 at 13:12

this worked for me. thanks! – hydronics Mar 16 '16 at 22:49

---

these solutions didn't work for me trying to start a python script with running Feh. The following worked. It starts a script after login.

Open a terminal session and edit the file

```
sudo nano /etc/profile
```

Add the following line to the end of the file

```
/home/pi/your_script_name.sh
```

replace the script name and path with correct name and path of your start-up script. Save and Exit

Press Ctrl+X to exit nano editor followed by Y to save the file.

Here's what my script.sh looked like:

```
#!/bin/sh
cd /
cd home/pi/
sudo python your_python_sript.py &
exit 0
cd /
```

I think I made both the script.sh and script.py executable using the chmod

```
sudo chmod +x home/pi/your_script_name.sh
sudo chmod +x home/pi/your_python_script.py
```

edited Mar 9 '16 at 5:40

answered Mar 8 '16 at 23:21

hydronics

1 2
