

# Run Program at Boot using Cron

Jan 2017



---

## Introduction

I was using the `.bashrc` file when logged in as a user (from your home directory, type `sudo nano .bashrc`) to launch scripts upon the Raspberry Pi bootup. However, while this works, it has one really annoying issue. If you login from another place, say from SSH, then the script runs again, which isn't great for my application.

So I am now using Cron, which works as I expect, Cron starts the script at boot, and I can login from anywhere without fear of the script running again.

Cron (short for the greek word for time - Chronos) is a scheduling utility that can run tasks on a time base of minutes, hours, days etc. (great for making automatic weekly backups). But Cron can also run tasks at boot, which is the part we are going to use.

To avoid putting loads of commands in to Cron, its best to group them all in to a file, then just reference that file in Cron. For this example we will make a file called `launcher.sh`.

---

## Create Launcher Script

To create our launcher script, first ensure you are in a directory you have rights to, on the Raspberry Pi this will generally be the Pi location in the Home folder.

```
cd /home/pi
```

Now enter the following:

```
sudo nano launcher.sh
```

Paste this code in to the `launcher.sh` file you have open.

```
#!/bin/sh
# launcher.sh
# navigate to home directory, then to this directory, then execute python script.

cd /
cd home/pi
sudo python nameOfScriptorProgram &
```

Be sure to replace the part `nameOfScriptorProgram` with the name of your script or program.

Save the file by using Ctrl-X and answering Yes.

## Execution Rights

The script will not run unless it is made executable. to do this enter the following:

```
sudo chmod 755 launcher.sh
```

---

## Add a logs folder

In case we get any cron errors, its best to add a folder that we can output any cron logs to, in the same folder that your script or program is, enter the following:

```
sudo mkdir logs
```

You should now have a logs folder, use ls to check.

---

## Testing the Launcher.sh

Before adding as a Cron job, it is a good idea to test the script, to do this enter:

```
./launcher.sh
```

Your script/program should now run. If it does then you can move on to the Cron part.

---

## Add Launcher.sh to Cron

Finally we need to add the launcher.sh to Cron. to do this we need to enter the following (if Cron has never been run, you will get a warning, just continue and a new Cron file will be created)

```
sudo crontab -e
```

Scroll to the bottom of the Cron file and enter:

```
# Cron Job to start script/program at boot
@reboot sh /home/pi/launcher.sh >/home/pi/logs/cronlog 2>&1
```

When you boot/reboot your Raspberry Pi, this will now run the launcher.sh, and if necessary, output any errors in the logs folder.

---

## References

This originally came from the following website, I have copied it here in case this site ever goes down. I'd like to say thank you to the original author.

<http://linux.about.com/od/linux101/f/sh-Linux-Command-Unix-Command.htm>

---

From:

<http://cameraangle.co.uk/> - WalkerWiki - [wiki.alanwalker.uk](http://wiki.alanwalker.uk)

Permanent link:

[http://cameraangle.co.uk/doku.php?id=run\\_program\\_at\\_boot\\_using\\_cron&rev=1485900914](http://cameraangle.co.uk/doku.php?id=run_program_at_boot_using_cron&rev=1485900914)

Last update: **2023/03/09 22:35**

