Installation and use of Panama frog experiment Raspberry Pis

Mario Santos Mira

06/01/2023

Installation

OS installation

For these experiments we will be using the legacy version of Raspberry Pi. The legacy version has support for the picamera python module which will make things easier with my knowledge.

Using the Raspberry Pi Imager, we can select an alternative version of the Raspberry Pi OS. Here we want to select the "Raspberry Pi OS (legacy)" (or the Lite version). Before starting the installation, we need to change some options:

1) Enable SSH, so we can connect to the Pis without an extra monitor and keyboard;
2) Set username and password. Here I assigned each of the Pis a number and standardized both according to number. Usernames are panamaX, and passwords are panamaXpi, where X is the assigned number;
3) Configure wireless LAN. This might be necessary to update and install things on the Pis if there is no available Ethernet internet connection. Just write down the name and password for the connection. Make sure to also select the correct country for Wi-Fi. "eduroam" might not work.
4) Set the time zone.

After all the options are set, select the storage location where the OS will be installed making sure it is the SD card and not some other drive.

Setup

Once the installation is done, we can plug the Pi to power and wait for it to boot. If the Pi connected to a network (wireless or wired) it is possible to find its local IP address. This can be done by accessing the router and seeing which new IP address appeared, or you can use a Linux program with the following command:

```
sudo arp-scan -l
```

This will show all devices connected to the same network as the device running the command. With the local IP address, we can connect to the Pis using an SSH connection. In Linux this is already included in most installations, so you can just run:

```
ssh panamaX@xxx.xxx.xxx.xxx
```

Where X is the number assigned to the specific Pi, and the xxx.xxx.xxx.xxx is whatever IP address it has. This will ask for the password (and a confirmation if you actually trust the Pi when you first connect) and it will log you into the Pi.

Specific for the storage Pi The storage Pi, named panama1, will be assigning IP addresses in our offline local network, so we need to install an extra package:

```
sudo apt install dnsmasq
```

Once installed, we can configure the DHCP server to assign specific IPs. We do this by modifying a specific file:

```
sudo nano /etc/dnsmasq.conf
```

And adding the following at the end of the file:

```
interface=eth0
bind-dynamic
domain-needed
bogus-priv
dhcp-range=192.168.178.150,192.168.178.199,255.255.255.0,12h
```

As specified, the Pi will only work as a router (assigning IPs) when connected with Ethernet (the eth0 part), and it will assign IPs between 192.168.178.150 and 192.168.178.199.

Once added, we restart the server, so the changes can be applied:

```
sudo service dnsmasq restart
```

I followed this page to install this.

All Pis

Static IP address First we will assign static IP addresses to each Pi when using Ethernet. This will make it so that connecting to any of them will be much simpler, and we won't have to always look up their IP addresses. The information can be found here. It just involves modifying one file in each Pi:

```
sudo nano /etc/dhcpcd.conf
```

By adding the following at the end of the file or uncommenting and modifying the already existing section:

```
interface eth0
static ip_address=192.168.178.15X/24
static routers=192.168.178.150
static domain_name_servers=192.168.178.150
```

The X is the number of the Pi minus 1 (so panama1 is 150, panama2 is 151, and panama3 is 152). The router and domain name parts should point to the storage Pi. This Ethernet IP should be setup after a reboot.

Updating and installing packages To update the Pis we need to run two commands. First, we update the package manager:

```
sudo apt update
```

Next we upgrade whatever packages we already have installed:

```
sudo apt upgrade
```

This will ask for a confirmation. You can write "y" and press Enter.

Once updates are finished we can install the last necessary bit. The first one is `screen`. This package allows you to store you terminal instance and disconnect from the Pis without interrupting whatever you were doing. We install this with"

```
sudo apt install screen
```

Confirm all prompts and it will be installed.

Next we need to install a Python package to transfer files securely between the Pis. From here we see the package needs the Rust compiler. So we will install the Rust compiler first with:

```
curl https://sh.rustup.rs -sSf | sh
```

This will ask if you want the default installation. You can type “1” to continue with defaults. You also might need to restart the terminal to have the Rust compiler available. If you are connected via SSH, you will have t disconnect and connect again. Now that Rust is installed, we can install `paramiko` by using:

```
pip install paramiko
```

It should just install without any problems.

Disabling Wi-Fi We can disable the Wi-Fi access in the Pis since, if everything goes right, we won’t need it any more, and it is very likely it won’t connect to eduroam. To do this we run this command on all Pis:

```
rfkill block wifi
```

To undo this step, we run:

```
rfkill unblock wifi
```

To connect again to a wireless network, we can use the `raspi-config` options to again type the name and password of a Wi-Fi network.

Specific for camera Pis

Enabling the camera The camera Pis also need to have their camera modules enabled. To do this we go into the `raspi-config` options:

```
sudo raspi-config
```

Go into “Interface options” > “Camera” and enable it. It will prompt you to reboot the Pi. Once booted the camera should be operational.

SSH without password With this step we won't need to use passwords each time we want to log into a Pi. The full instructions are found here. We start by generating SSH keys in our laptops:

```
ssh-keygen -t rsa
```

There's no need to choose a passphrase or anything else. Just use all the default options and leave the passphrase empty. Next, go into the ssh directory and confirm the keys were created:

```
cd ~/.ssh/
```

And

```
ls
```

You should see at least `id_rsa` and `id_rsa.pub`. The pub file will be stored in the target Pis so that it "knows" your laptop and allows you in without password. Next we copy this file to the target Pi:

```
ssh-copy-id -i id_rsa.pub panamaX@192.168.178.xxx
```

Once again, the Xs correspond to whatever Pi you are targetting. This should be repeated for each Pi. Once this is done, you can connect with:

```
ssh panamaX@192.168.178.xxx
```

And it should just connect you.

Use

To use all the Pis you will have to connect via SSH. I recommend using a Linux distribution or installing a Linux distribution on your Windows laptop, as seen here. This guide (up to step 4) will allow you to use an Ubuntu terminal as if you had the whole Ubuntu OS.

Connecting to the Pis

At any point, you can connect to any Pi if you know it's username, IP address and password:

```
ssh panamaX@192.168.178.xxx
```

However, we can make things simpler in two ways. First, we can remove the need for a password (as done in the last step of the Installation section). And second, we can remove the need to type all that by creating aliases.

SSH without password (again) See the previous section with the same name. Using the Ubuntu Windows subsystem should be the same process. I haven't tried other programs.

Aliases To create aliases (that is you can create a simpler word that will do the same command) we need to modify the `.bashrc` file.

```
nano ~/.bashrc
```

In this file, at the end, you can add whichever aliases you want. For example, my aliases for the Pis are:

```
alias sshpa1='ssh panama1@192.168.178.150'
alias sshpa2='ssh panama2@192.168.178.151'
alias sshpa3='ssh panama3@192.168.178.152'
```

And with passwordless ssh once I type `sshpa1` and hit Enter, I'm automatically logged into the storage Pi (panama1).

Running the trials

Once logged into a camera Pi you should change directories (`cd`) to the Code folder:

```
cd Code
```

In here you will find three Python scripts. One of them just contains a function to transfer files (`transfer.py`), one of them has a script to make a small video (~7 seconds) to test position of stuff, and the other will run the trial (`trial.py`). **Before starting, you should run the `screen` program.**

```
screen
```

This will allow you to exit the Pi connection while allowing the script to continue running.

We should confirm if the camera is in the correct place. To do this we will take a simple photo with the same resolution as the videos:

```
raspistill -o PICTURE.jpg -w 1280 -h 720
```

This will take a photo with the camera with a 1280 by 720 resolution and save it as the "PICTURE.jpg" file. Next, we copy the file into our laptop to have a look (this step should be done in a second terminal. Just open another window of your terminal of choice):

```
scp panamaX@192.168.178.XXX:~/Code/PICTURE.jpg .
```

`scp` is a command to securely copy files between computers. `panamaX@192.168.178.XXX` is the site/computer you want to connect to and `:~/Code/PICTURE.jpg` is what you want to copy over. The final `.` is important. It specifies the location where you want to store the file. A single dot means "here", so where ever you have you terminal running. Make sure you `cd` to the folder you want to or change the `.` to the relevant path.

Once you have the .jpg file in your laptop you can just open it to see if every thing is correct.

To run the trial script you use the following command:

```
python3 trial.py
```

The first step in the script is to input the code for the name of the trial. This should be in the form of FXMYMZ, where X is the female number and Y and Z and the male numbers. Once you input that, the script will wait for a keypress to start the wait and trial timers. At the end of the trial and after transferring the video file to the storage drive, the script will ask if you want to delete the file. Answering "Yes" will delete the file and any other answer will skip this step.

While waiting, you can disconnect from the Pi and unplug the Ethernet cable. **To do this, you need to press Ctrl+A followed by Ctrl+D to store your `screen` instance.** To return to the same instance when you return, use the following command"

```
screen -r
```

Backing up the videos

Location By default, the python script for the trials will transfer each video to the server Pi just before finishing. The video will end up in the `~/external/Frog_videos/` directory. The `~/external/` folder is not part of the Pi. It is the mount point for the external hard drive. To mount the drive in this location I used the following command:

```
sudo mount /dev/sda1 external/
```

The `/dev/sda1` is the designation for the main partition on the external hard drive.

Moving videos Within the Frog_videos/ directory, I have been manually creating a new folder with the current date in the format YYYYMMDD (for example, today is 20230125):

```
mkdir 20230125/
```

For each trial (so, two videos) I manually move the files in Frog_videos/ into Frog_videos/YYYYMMDD. I am not doing this manually because it has been working as a conscious check. Each day you make sure to confirm the current date and also that 16 videos end up in the correct folder.

Duplicating videos To make sure we have the videos safe, We are going to upload all of them to the shared Google Drive. This can be done while the following trial is running (if the internet speeds allow) or overnight. To do this, we need to duplicate the videos into a laptop or a separate drive.

In the laptop If you are duplicating the videos into your laptop, you can run a simple command in your terminal (that is, when not logged into one of the Pis). This command assumes that the videos have moved inside the relevant date folder:

```
rsync -rhvu --progress=info2
panama1@192.168.178.150:~/external/Frog_videos/YYYYMMDD/
/mnt/c/Users/Mario\ Santos\ Mira/Desktop/Frogs/YYYYMMDD/
```

There is on newline in the command. I had to press enter for these to appear okay in the PDF

`rsync` is a program similar to `scp`, but it will not try to copy files that already exist in the target directory. The options (the letters after the "-") are: `r` is to copy the whole directory (recursive); `h` is to show human-readable units; `v` is to show more information (verbose); and `u` is to only copy new files (update). The `--progress=info2` displays even more information.

The other two arguments should be familiar from the `scp` command. The first one, `panama1@192.168.178.150:~/external/Frog_videos/YYYYMMDD/` is the directory we want to copy. The second one, `/mnt/c/Users/Mario\ Santos\ Mira/Desktop/Frogs/YYYYMMDD/` is the directory you want to copy into. **Notice the / at the end of both. This is necessary because it makes sure both are directories and not files!**

In another drive Here, I am using Joana's 60Gb USB flash drive. Other USB drives may have different designations.

In the Raspberry Pi, these drives aren't automatically mounted (that is, we can not access them when plugging them in), so first we find the name of the flash drive partition:

```
sudo fdisk -l
```

Be careful with `fdisk`, it can format drives so make sure you are running this exact command In my case the flash drive partition had the designation `/dev/sdb1`.

Next, we can make a directory on the Pi where we will mount the flash drive to:

```
mkdir flashdrive/
```

And mount the flash drive in `~/flashdrive/`:

```
sudo mount /dev/sdb1 flashdrive/
```

All the data in the flash drive is now found on the `~/flashdrive/` directory.

Similarly to the laptop option, now we make directories inside the flash drive, and we can rsync into them (see the laptop option to understand all the options I used):

```
rsync -urhv --info=progress2
external/Frog_videos/YYYYMMDD/
flashdrive/Frog_videos/YYYYMMDD/
```

There is on newline in the command. I had to press enter for these to appear okay in the PDF

Important: When you want to remove the flash drive, you will have to unmount it. Similar to mounting, you can use:

```
sudo umount /dev/sdb1
```

Only then can you unplug the USB stick.