Getting started with Raspberry Pi

Contents

This semester, we'll use the Raspberry Pi for a series of lab exercises and to prototype your project design. In this lab exercise, we'll practice connecting to the Pi using different configurations, so that you can work with your Pi both in the lab (if you attend in person) and at home.
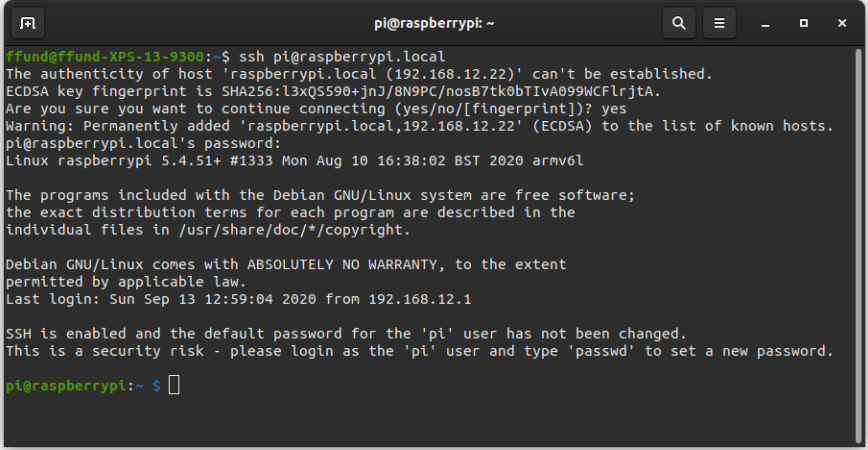
0.1 Notes

- Please be careful with lab equipment, and treat it gently. (For example: when removing the power supply from your Pi, don't yank the cable out of the jack, since you may pull the jack right off the PCB. Instead, hold the jack against the board with one hand while removing the cable with the other hand.)
- You will submit your lab work in Gradescope. You will upload screenshots/photos and answer some questions as described in the Gradescope assignment. You do not have to write anything else (e.g. no description of procedure, etc.)
- Read each subsection of this lab manual in its entirety before you start following the instructions in it. Some instructions are modified by explanations that come afterwards.
- Although you may work with a partner, this collaboration is limited to discussion and comparison. Your partner is not allowed to construct or modify your circuit, log in to your Pi, or run commands or write code on your Pi. Similarly, you are not allowed to do these things for your partner.
- For your lab report, you must submit data, code, screenshots, and photos from your own experiment. You are not allowed to use your lab partner's data, code, screenshots, or photos.
- For any question in the lab report that is marked "Individual work", you should *not* collaborate with your lab partner or anyone else (even via discussion). You can use your notes, the lab manual, or the lecture slides and video to help you answer these questions.

0.2 Prepare your workstation

In this lab session, you'll practice two ways to connect to your Pi: using a terminal session over an SSH connection, and using a GUI session over VNC.

Here's what an SSH connection looks like. Note that you only have access to a terminal on the Pi when using SSH - you can't run any applications that require a graphical interface. However, in most cases we'll only need a terminal anyway, and a terminal-only connection is much faster and smoother.

Figure 1: Connecting to a Pi over SSH.

For SSH, you will need a terminal and an SSH client:

- If you are using a Mac or Linux device, the built-in terminal that comes with your operating system includes an SSH client. Make sure you know how to open the Terminal application that comes with your operating system.
- If you are using a Windows device and you do not already have an SSH client, you will have to download and install one. I recommend Git Bash, which you can download here: https://git-scm.com/downloads. Once you have downloaded and installed this application, open a Git Bash terminal, like this:
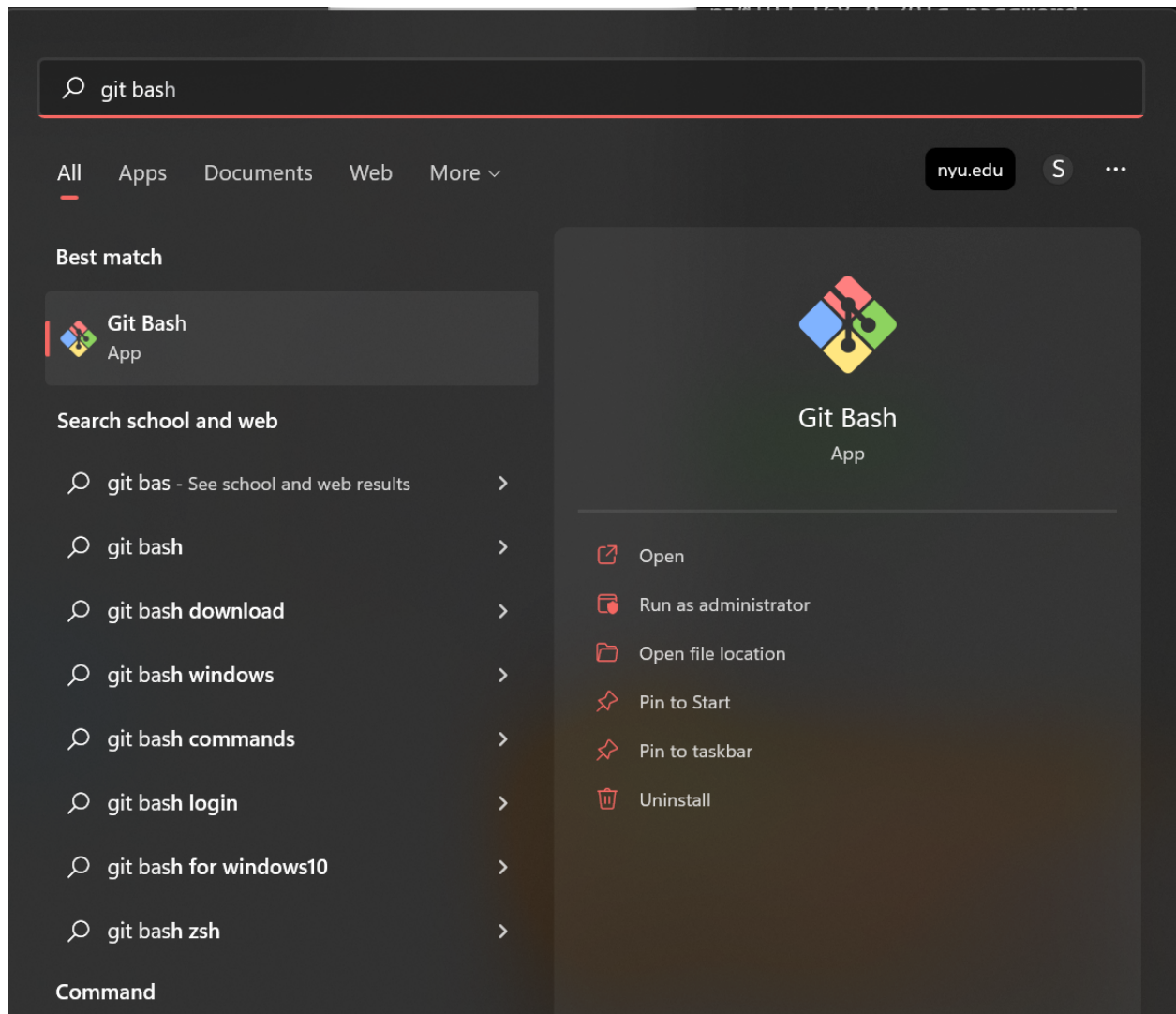
Figure 2: Open a Git Bash terminal - for Windows users.

Once you have a terminal window open, run

```
ssh
```

You should see some usage information, like this (although not necessarily identical to this):

```
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

This shows that you have an SSH client! If you aren't able to run an SSH client and see usage information, stop and ask for help.

While we'll mostly work directly in the terminal using SSH, sometimes we'll need to run an application on the Pi using a graphical user interface. We can establish a GUI connection using VNC.

Here's what a VNC connection looks like. Note that you have access to a graphical interface on the Pi when using VNC, but the connection may not be as smooth as an SSH session.
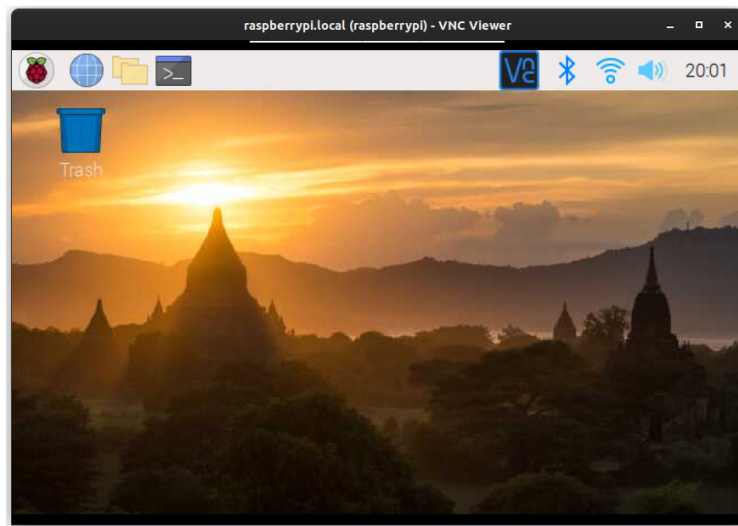
Figure 3: Connecting to a Pi with VNC.

To use VNC to connect to your Pi, you will need to download the RealVNC viewer. It is available for most platforms at https://www.realvnc.com/en/connect/download/viewer/.

Make sure you know how to open VNC viewer. It will look like this when you run it for the first time:
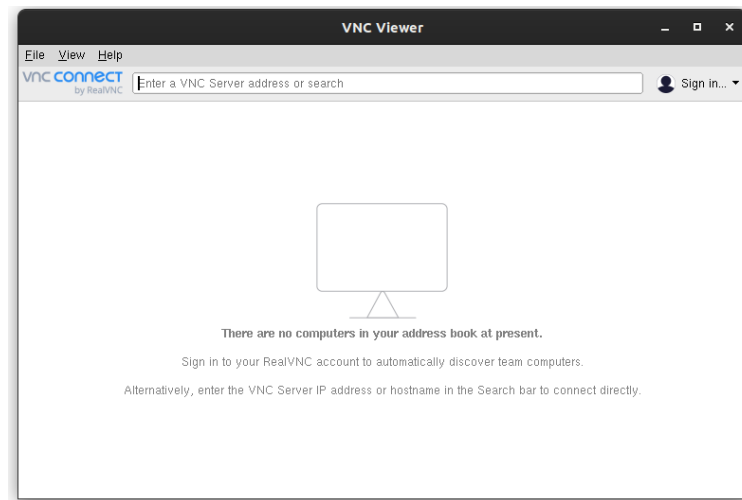
Figure 4: VNC viewer.

1   Connecting to your Pi in LC011

1.1   Turn on your Pi for a headless session

Your lab kit includes a Raspberry Pi Zero Wireless, a micro USB power supply, and a micro SD card. Look at your Pi, and identify the location of important jacks and status LEDs. Pay special attention to the location of the micro USB power jack; note that there are two micro USB jacks right next to one another. One of these has the words `PWR IN` printed next to it on the PCB, and the other has the word `USB`.
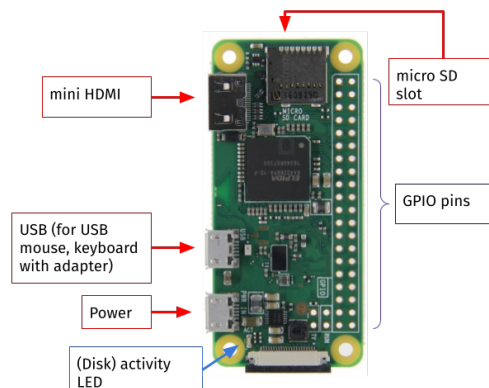
Figure 5: Annotated photo of the Pi Zero Wireless

Don't connect the power supply yet - first, you'll make sure the SD card is installed correctly. Put the micro SD card into the slot on the Pi. The gold contacts on the micro SD card will lie on the Pi and the writing on the micro SD card will face up, when it is seated properly.

Finally, plug in the micro USB power supply, and then connect it to the power jack on your Pi. The disk activity LED will flash occasionally as the Pi boots.

1.2 Establish an SSH connection over the ece4313 network

When you are working in the LC011 lab, you'll use the `ece4313` network in the lab to connect to your Pi. For this to work, both your Pi and your laptop should be connected to the `ece4313` network, which is available in our lab space (LC011).

Use the wireless network connection utility on your laptop to connect to the network `ece4313`, using `CompEDP1` as the password. Note that if your laptop is connected to the `nyu` network, you won't be able to reach your Pi. If at any point you lose connectivity to the `ece4313` network, you'll have to re-connect to the network in order to reach your Pi.

Next, you'll need to know the IP address that is assigned to your Pi on the `ece4313` network. Find the numbered sticker on your Pi. Your Pi will have address `192.168.0.N`, where `N` is the number written on the sticker. (This applies only to the `ece4313` network - on another network, your Pi will have a different address!)

Now, in a terminal on your laptop, run

```
ssh pi@192.168.0.N
```

where `N` is the number written on the sticker. You may see a message like:

```
The authenticity of host '192.168.0.N (192.168.0.N)' can't be established.
ECDSA key fingerprint is SHA256:l3xQS590+jnJ/8N9PC/nosB7tk0bTIvA099WCFlrjtA.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Type `yes` and hit `Enter`.Then, enter the password `raspberry` when prompted. You may not see anything appear in the terminal output when you type the password - this is a security feature, so that anyone eavesdropping over your shoulder can't tell how many characters are in the password. It will look like you're not typing anything, but that's OK - just type the word out anyway, and hit `Enter`.

Your terminal prompt should now look like this

```
pi@raspberrypi:~ $
```

indicating that you are in a terminal session on the Pi, and any commands you run will run on the Pi, not your laptop.

Lab report: Prepare an instruction/checklist document for yourself to use in the future when you need to establish an SSH connection to your Pi over the `ece4313` network. Your list should include instructions specifically for *your* laptop, describing how to:

1. Connect your laptop to the `ece4313` network. Show a screenshot that demonstrates how you can tell you have done this step successfully.
2. Power on the Pi and have it load the operating system from SD card. Show a photo that demonstrates how you can tell you have done this step successfully.
3. Open a terminal on your laptop. Show a screenshot that demonstrates how you can tell you have done this step successfully.
4. Identify the address of your Pi on the `ece4313` network. Show a photo to demonstrate this step.
5. Use SSH to connect to a terminal session on your Pi, with this address. Show a photo that demonstrates how you can tell you have done this step successfully.

1.3   Enable VNC

Now that you have a terminal session on the Pi, you can also enable VNC on the Pi, so that you can also establish a graphical connection. The Raspberry Pi OS includes a configuration tool called `raspi-config` that we can use to enable VNC. Run

```
sudo raspi-config
```

(`sudo` is a utility that lets you run any command with elevated privileges. To make changes to the system configuration, you will need elevated privileges.)

Within `raspi-config`,

- Use the arrow keys to navigate to `Interface Options` and hit `Enter`.
- Navigate to `VNC` and hit `Enter` again.
- When asked "Would you like the VNC server to be enabled" use the `Tab` key to toggle the answer so that `Yes` is highlighted. Then hit `Enter` to accept your choice.
- After a few seconds, you should see a message saying that "The VNC Server is enabled." Hit `Enter` to confirm.

The intial default VNC screen resolution is too small, so while we are in `raspi-config` we can also change that.

- Use the arrow keys to navigate to `Display Options` and hit `Enter`.
- Navigate to `VNC Resolution` and hit `Enter` again.
- Use the arrow keys to select `1024x768`. Then use the `Tab` key to select `OK` and hit `Enter`.
- When you see the confirmation that the resolution is set, select `OK` and then hit `Enter`.
- If prompted to reboot the Pi, you can accept the suggest by using the `Tab` key to select `Yes` and then hitting `Enter`. If you are not prompted to reboot, use `Esc` to close `raspi-config`.

Then, you can close your SSH session with the following command:

```
exit
```

(This was a one-time configuration - after you have done this, you'll be able to connect using VNC in the future without repeating the `raspi-config` step.)

1.4 Establish a VNC session over the ece4313 network

Now, open the RealVNC Viewer, and in the address bar at the top, type the address of *your* Pi. (It's not necessarily the same as the address shown in my screenshot!) Then, hit `Enter`.
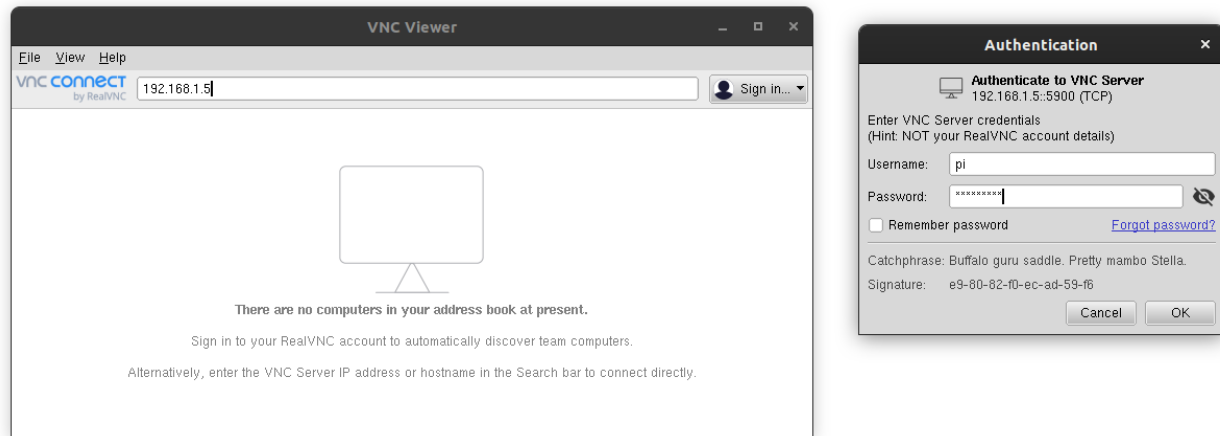
Figure 6: Connecting to a Pi using VNC.

On some occasions, you may see a message about the VNC server's signature, and will have to press `Yes` or `Continue` to confirm that you want to connect. Then, in the Authentication dialog, enter `pi` as the username and `raspberry` as the password, and hit `OK`.

The first time you connect using VNC, you may be prompted to set your time zone and keyboard configuration. Also, you may see a message warning you that "SSH is enabled and the default password for the 'pi' user has not been changed". You can press `OK` to acknowledge the warning.

You might be prompted to update the software on your Pi, but you *don't have to do that.* (It will take a long time to run software updates, so you may not want to do that right now.)

Then, you'll see the Raspberry Pi OS desktop:
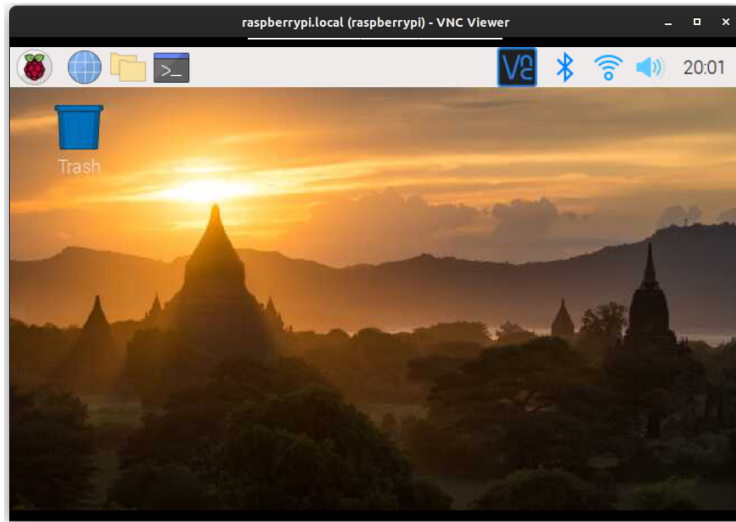
Figure 7: Connecting to a Pi with VNC.

Lab report: Prepare an instruction/checklist document for yourself to use in the future when you need to establish a VNC connection to your Pi over the `ece4313` network. Your list should include instructions specifically for *your* laptop, describing how to:

1. Connect your laptop to the `ece4313` network. Show a screenshot that demonstrates how you can tell you have done this step successfully. (You can copy your own answer from a previous section.)
2. Power on the Pi and have it load the operating system from SD card. Show a photo that demonstrates how you can tell you have done this step successfully. (You can copy your own answer from a previous section.)
3. Open the VNC viewer on your laptop. Show a screenshot that demonstrates how you can tell you have done this step successfully.
4. Identify the address of your Pi on the `ece4313` network. Show a photo to demonstrate. (You can copy your own answer from a previous section.)
5. Use VNC to connect to a GUI session on your Pi, with this address. Show a photo that demonstrates how you can tell you have done this step successfully.

---

2   Connecting to your Pi at home

Before you leave the lab, you should also make sure that you can connect to your Pi even outside the lab (when the `ece4313` network is not available to you.) It's important that you are able to connect to your Pi outside of this lab:

- when you don't finish a lab assignment during the lab session, and want to do the rest at home,
- in case you have to attend a lab session from home because you are not feeling well,
- for later in the semester - so that you can work on your project outside of scheduled lab sessions.

You can use your mobile phone or laptop to create a WiFi hotspot, and configure your Pi to connect to this hotspot. The following sections will show you how.

2.1   Create a hotspot

2.1.1   Create a hotspot on an iPhone

To create a hotspot on your iPhone, follow the instructions at https://support.apple.com/en-us/HT204023. However, to make sure your Pi is able to connect to this hotspot, there are some important additional instructions:

- Some iPhone versions have a "Maximize Compatibility" option in the "Personal Hotspot" configuration menu. If your iPhone has this option, you must make sure it is turned *on*.
- The default SSID of the iPhone hotspot is set according to the name of the iPhone, for example, "Fraida's iPhone". It's easier to work with an SSID that does not have any spaces or special characters, like the apostrophe. If you're willing, you can change the iPhones name in the General settings tab (General > About > Name) to a name which does not contain any special characters, save, and enable the hotspot again.
- Set your password so that it has no spaces or other special characters.

Once you have set this up, make a note of the network *name* and *password* - you will need these in a later step. Continue from the section titled *Edit configuration files on your Pi*.

2.1.2   Create a hotspot on an Android phone

To create a hotspot on your Android phone, follow the instructions at https://support.google.com/android/answer/9059108?hl=en.

Set the network name and password so that it has no spaces or other special characters.

If your wireless carrier limits tethering, you can turn off mobile data before you create the WiFi hotspot. If mobile data is turned off, the devices connected to your WiFi hotspot won't have Internet access, but they will be able to connect to one another over the hotspot network.

Once you have set this up, make a note of the network *name* and *password* - you will need these in a later step. Continue from the section titled *Edit configuration files on your Pi*.

2.1.3 Create a hotspot on a Windows 10 or Windows 11 laptop

To create a hotspot on your Windows 10 or Windows 11 laptop, follow the instructions at https://suppor t.microsoft.com/en-us/help/4027762/windows-use-your-pc-as-a-mobile-hotspot.

Set the network name and password so that it has no spaces or other special characters.

To make sure the Pi will connect to this hotspot, you need to make sure it is in the 2.4GHz band. If your hotspot shows "Any Band", then edit the properties:

Figure 8: Edit Windows hotspot properties.
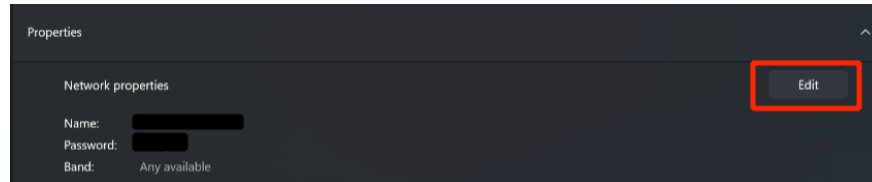
Change the "Network Band" option to 2.4GHz and save this setting:
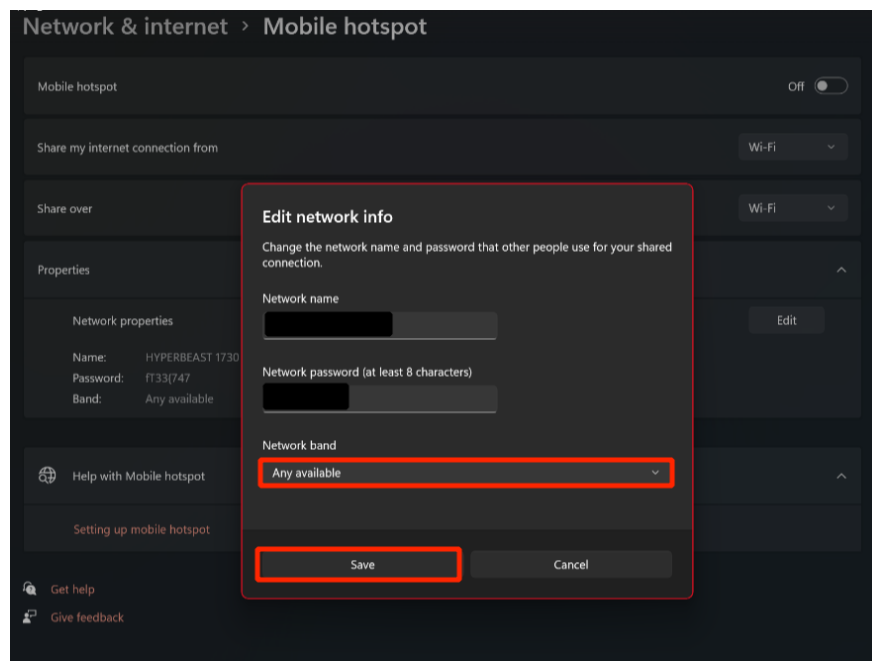
Figure 9: Change the network band.

Once you have set this up, make a note of the network *name* and *password* - you will need these in a later step. Then, you can stop the hotspot and re-connect your computer to the `ece4313` network. Continue from the section titled *Edit configuration files on your Pi*.

2.1.4 Create a hotspot on a Mac

This process is more complicated than the other options, so you should attempt this only if you cannot create a hotspot using iPhone, Android, or Windows. (For example, if your wireless carrier does not allow you to create a hotspot from your phone.)

To create a hotspot on a Mac, you will actually need a second device (e.g. an iPhone) to be connected to the Mac using a wired USB or Thunderbolt cable. Then, on the Mac, open System Preferences > Sharing and select "Internet Sharing". Configure it to share your connection *from* "iPhone USB" *to* "WiFi":
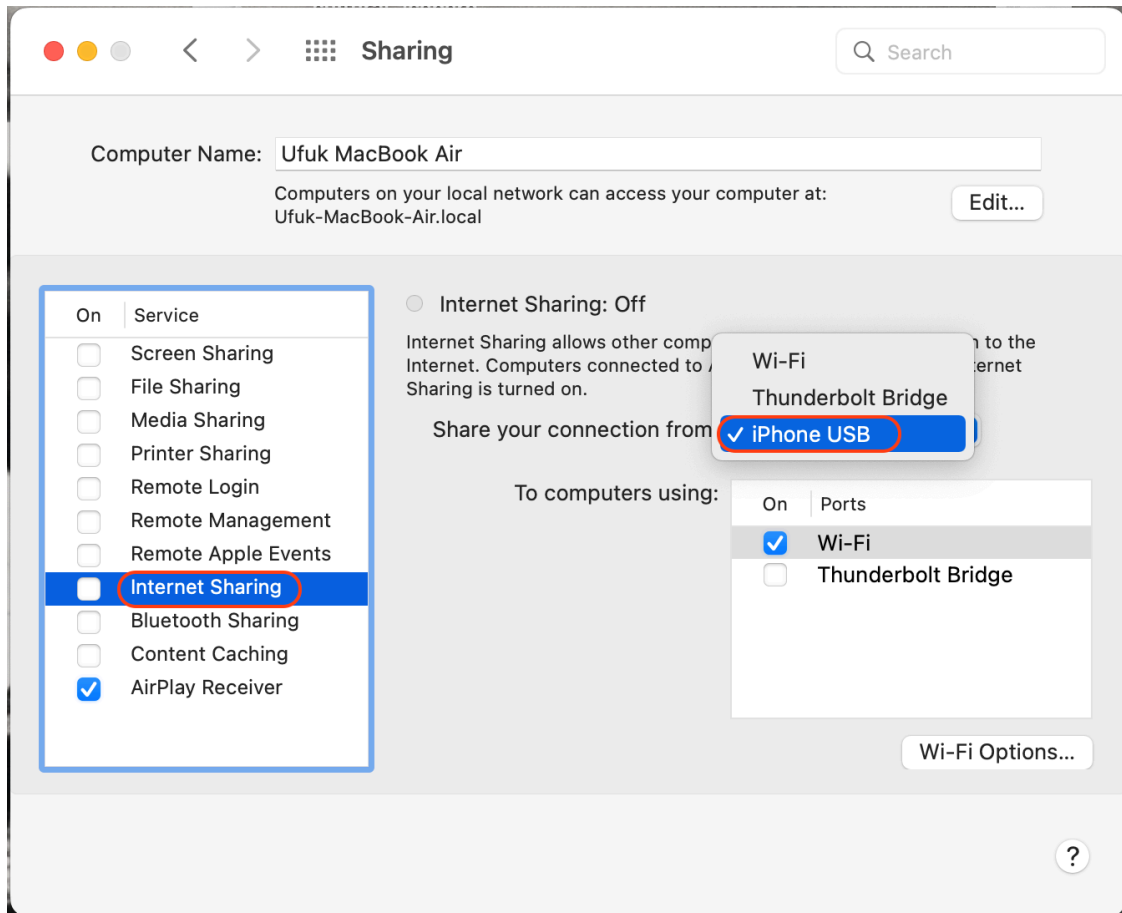
Figure 10: Configure Internet sharing on a Mac.

Then, click on "WiFi options". Set the network name and password so that it has no spaces or other special characters. Also, set the channel to either 1, 6, or 11.
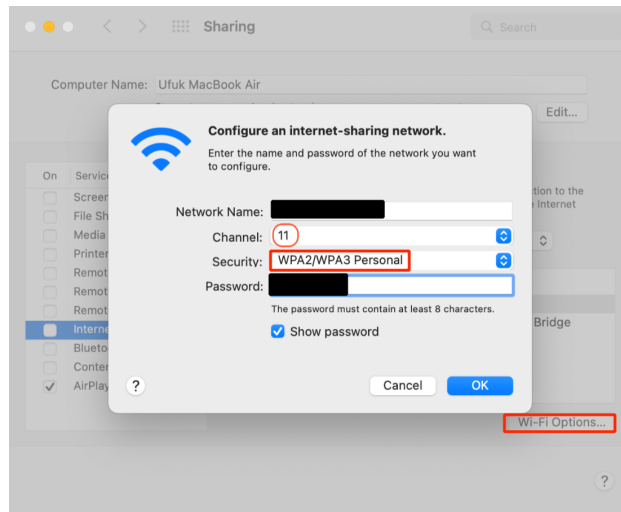
Figure 11: Configure WiFi settings on a Mac.

Finally, once everything is configured, check the box next to "Internet Sharing":
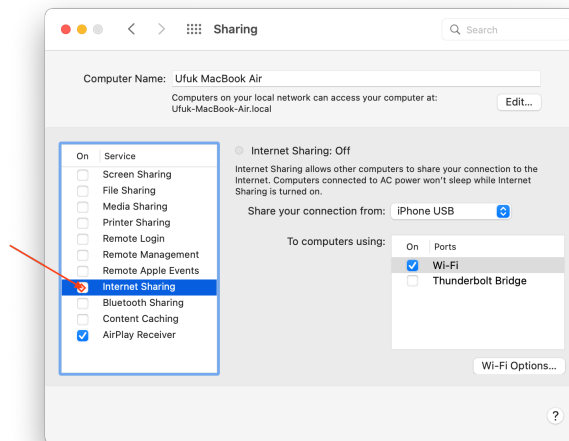
Figure 12: Turn on Internet sharing.

Once you have set this up, make a note of the network *name* and *password* - you will need these in a later step. Then, you can stop the hotspot and re-connect your computer to the `ece4313` network. Continue from the section titled *Edit configuration files on your Pi*.

2.2 Edit configuration files on your Pi

Now that you know the *name* and *password* of your hotspot network, you will need to configure your Pi to connect to this network.

You will need your Pi and your laptop to still be connected to the `ece4313` network, and you should connect to your Pi using SSH. We are going to edit *two* configuration files on the Pi.

**First**: we will edit the `sysctl` configuration file, which includes many system-wide configuration options. We can are going to add a line to this file to configure the system to respond to "broadcast pings" (more details on that later!) At the terminal *on your Pi*, run

```
echo "net.ipv4.icmp_echo_ignore_broadcasts=0" | sudo tee -a /etc/sysctl.conf
```

Then, to load the change, run

```
sudo sysctl -p
```

Finally, run

```
sysctl net.ipv4.icmp_echo_ignore_broadcasts
```

and verify that the output shows

```
net.ipv4.icmp_echo_ignore_broadcasts = 0
```

Second: we will edit the `wpa_supplicant` configuration file, which describes the wireless network configuration.

In a terminal session on your Pi, run

```
iwlist wlan0 scan
```

This will show you all of the WiFi networks visible to your Pi. Make sure you can see your personal hotspot! If not, try running

```
iwlist wlan0 scan
```

a second or third time - sometimes, it will only "see" the network it is currently connected to on the first try, but will find other networks in a subsequent scan.

Once you are sure that your Pi can see your hotspot network, run

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

This will open the network configuration file in `nano`, a terminal-based text editor. You should see the following file contents:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="ece4313"
    psk=0c9f2bfe3dd03e035634302d3a37b9f523718de95c91eaadfaad4129fcc2a4cf
}
```

indicating that the Pi is configured to connect to the `ece4313` network.

Add a network with the SSID and password of your mobile hotspot. For example, if your SSID is `E71210` and the passphrase is `00934596`, the file will look like this:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="ece4313"
  psk=0c9f2bfe3dd03e035634302d3a37b9f523718de95c91eaadfaad4129fcc2a4cf
}

network={
  ssid="E71210"
  psk="00934596"
}
```

Next, define priorities, so that if both networks are available your personal network will be preferred. Make sure to give your personal network a higher priority than the lab network - otherwise, the Pi will just connect to the lab network, and you won't be able to test whether your personal network works!

If you add additional network blocks for your home WiFi network, make sure that:

- your laptop/mobile hotspot has the *largest* priority
- your home WiFi network has the second largest priority
- the `ece4313` network has the smallest priority

A higher value indicates higher priority (this is the opposite of what you might expect) - for example, the network with priority 2 is preferred over the network with priority 1. Your personal network's priority should be a *larger number* than the lab network:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
  ssid="ece4313"
  psk=0c9f2bfe3dd03e035634302d3a37b9f523718de95c91eaadfaad4129fcc2a4cf
  priority=1
}

network={
  ssid="E71210"
  psk="00934596"
  priority=2
}
```

To save (write **o**ut) the file, use `Ctrl+O`, and hit `Enter` when prompted (at the bottom) to accept the file name. Look for a message indicating "Wrote N lines" (or similar). Then, quit `nano` with `Ctrl+X`.

(Note: you can add as many networks as you like to this file. For example, if you have a home WiFi network that you want your Pi to connect to, you can add it, too.)

2.3   Find out your Pi's address

If you did everything right, your Pi should now connect to your "personal" network when it reboots. But, in order to access it, you'll need to find out its address. On your "personal" network, your Pi's address may be different every time it connects to the network.

To find out your Pi's address,

1. First, switch your personal hotspot on.
2. Then, reboot your Pi. At your Pi's terminal prompt, run:

```
sudo reboot
```

3. If your personal hotspot is provided by your phone, connect your laptop to this hotspot. Remember that your Pi and your laptop need to be on the same network, so if your Pi connects to your mobile hotspot, so should your laptop!

Your Pi should boot **after a few minutes** and connect to your personal hotspot. However, you will *not* be able to connect to your Pi using the "192.168.0.N" address - that address is specific to the `ece4313` network. On your personal hotspot, your Pi will have a different address (and it may be a different address every time you connect your Pi to your hotspot.)

The process of finding out the address will, again, depend on the type of hotspot device. I will provide instructions below for Android hotspot, Windows hotspot, or "other". If the Android or Windows hotspot instructions don't work for you (for example, because you have a different device model), you can still use the "other" instructions.

Before you proceed:

- Make sure you have rebooted your Pi according to the instruction above, *and* waited a few minutes afterward for it to come back up.
- Make sure your laptop is connected to the same personal hotspot that you configured your Pi to use.

2.3.1   Find out your Pi's address using its name - any device

On some networks/with some devices, you can reach your Pi by *name* without knowing its address. You can try this by running the following command in your *local* terminal:

```
ping raspberrypiXX.local
```

where in place of `XX` you substitute the number on the sticker on your Pi. This command says, "If there is a device on the same network named `raspberrypi.XX`, please respond!" If you get reply like

```
64 bytes from Y.Y.Y.Y: icmp_seq=1 ttl=64 time=0.049 ms
```

make a note of the IP address `Y.Y.Y.Y` in the response. Then, continue from the section *SSH into your Pi*.

If you don't get any response, or if you get an error message like `Destination Host Unreachable` or `Name or service not known` in the response, that's OK! Move on to the next option that is appropriate for your device/hotspot.

2.3.2 Find out your Pi's address - Android hotspot

If you are creating a hotspot from an Android device, you may see a "Connected Devices" area on the mobile hotspot settings page. After you Pi connects, you'll see a "raspberrypi" entry in that list. Click on that entry:
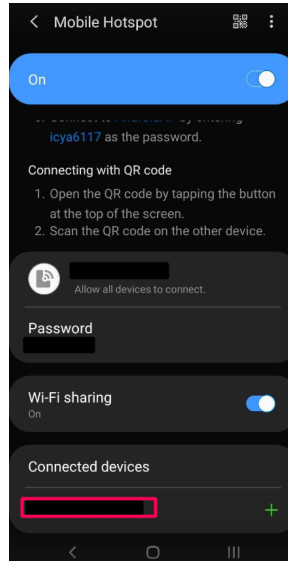

Figure 13: Find Raspberry Pi in connected device list on Android hotspot.

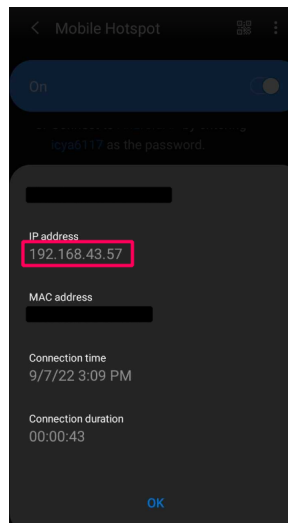Then, note the IP address that appears in the following window:

Figure 14: Find IP address from Android hotspot.

This should be the address of your Pi.

2.3.3 Find out your Pi's address – Windows hotspot

If you are creating a hotspot from a Windows laptop, in the Mobile Hotspot settings page you may see a list of connected devices. You can find your Pi in this list and note its IP address:
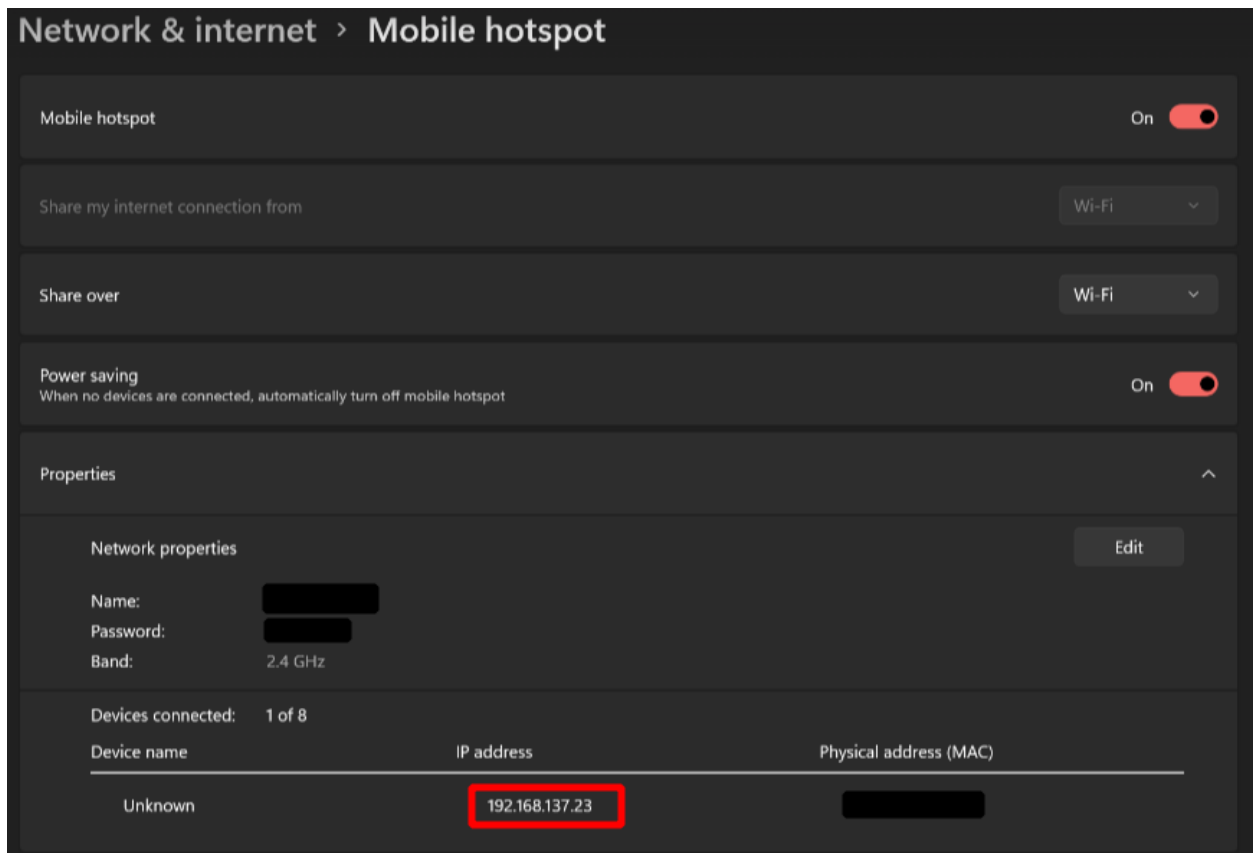

Figure 15: Find IP address from Windows hotspot.

2.3.4 Find out your Pi's address - Mac laptop

Otherwise, if your laptop is a Mac, you can find out your Pi's IP address as follows.

In a local terminal on your Mac (not on your Pi), run

```
ifconfig
```

In the output, find the block labeled `en0`. Look for the laptop's `inet` address (circled in blue in the example image below) and the `broadcast` address on the network (circled in purple in the example image below):

Figure 16: Finding your IP address and broadcast address on a Mac. In this example, an Android phone is providing the hotspot network.

This will look different, depending on the network you are on. Here's another example:

Figure 17: Finding your IP address and broadcast address on a Mac - another example. In this example, an iPhone is providing the hotspot network.

Then, in your Mac's local terminal, run

```
ping X.X.X.X
```

where in place of `X.X.X.X` you substitute the *broadcast address* (e.g. the address circled in purple in the examples above). This command says, "Every device on the same network as me, please respond!"

You will probably get at least one response like

```
64 bytes from X.X.X.X: icmp_seq=1 ttl=64 time=0.049 ms
```

from *yourself* - e.g. with the `inet` address you identified previously. You may also get a response from an address that ends in `.1`, e.g. `172.20.10.1` or `192.168.43.1` - this is the device that provides the hotspot. Look for a response from an address that is *not* one of those, and make a note of that address. Then, continue from the section *SSH into your Pi.*

19

2.3.5 Find out your Pi's address - Windows laptop, Windows laptop hotspot

If you use a Windows laptop, and this laptop provides the hotspot network but does not give you any list of "Connected Devices", you can find out your Pi's IP address as follows.

In a local terminal on your Windows laptop (not on your Pi), run

```
ipconfig
```

In the output, find the block for your WiFi adapter. is will be an entry that has an "IPv4 Address" associated with it, but that is not the "WiFi" adapter - for example, the block circled in blue in the image below:

Figure 18: Hotspot network adapter.

Your next step must be to find out the *broadcast address* of this network. Copy the value shown for "IPv4 Address" and the value shown for "Subnet Mask" into WolframAlpha, and look for the "broadcast ID" in the result, like this:

Figure 19: Find out Windows hotspot broadcast address.

Then, in your laptop's local terminal, run

```
ping X.X.X.X
```

where in place of `X.X.X.X` you substitute the *broadcast address* (e.g. the address circled in purple in the example above). This command says, "Every device on the same network as me, please respond!"

You will probably get at least one response like

```
64 bytes from X.X.X.X: icmp_seq=1 ttl=64 time=0.049 ms
```

from *yourself* - e.g. with the "IPv4 Address" you identified previously (in the blue-circled block above). Look for a response from an address that is *not* your laptop's own address, and make a note of that address. Then, continue from the section *SSH into your Pi*.

2.3.6   Find out your Pi's address - Windows laptop, Android or iPhone hotspot

If you use a Windows laptop, and your hotspot network is provided by an Android phone or iPhone, you can find out your Pi's IP address as follows.

In a local terminal on your Windows laptop (not on your Pi), run

```
ipconfig
```

In the output, find the block corresponding to your WiFi adapter.

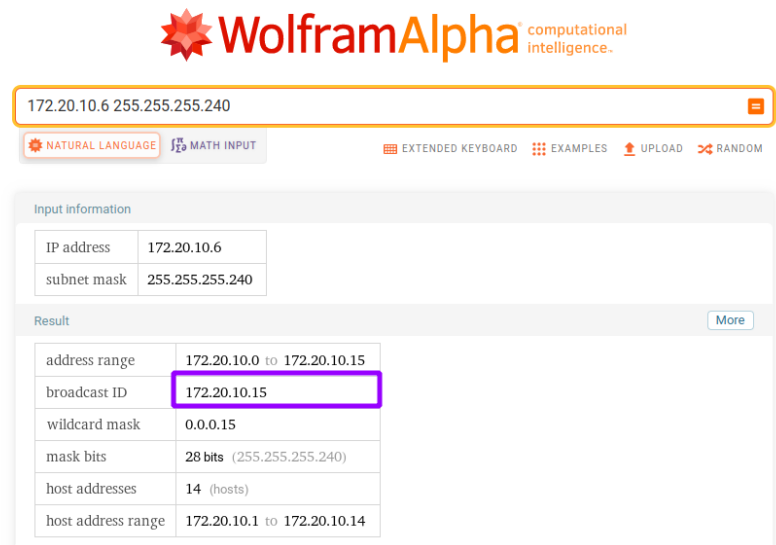Figure 20: Find hotspot network on Windows laptop. In this example, the hotspot is provided by an iPhone.

The network configuration will be different, depending on the type of hotspot device you are using. Here's another example:

Figure 21: Find hotspot network on Windows laptop. In this example, the hotspot is provided by an Android phone.

Your next step must be to find out the *broadcast address* of this network. Copy the value shown for "IPv4 Address" and the value shown for "Subnet Mask" into WolframAlpha (circled in blue in the examples above), and look for the "broadcast ID" in the result, like this:

Figure 22: Find out Windows hotspot broadcast address. This example shows the broadcast address for the "IPv4 Address" and "Subnet Mask" shown in the iPhone example.

Then, in your laptop's local terminal, run

```
ping X.X.X.X
```

where in place of `X.X.X.X` you substitute the *broadcast address* (e.g. the address circled in purple in the example above). This command says, "Every device on the same network as me, please respond!"

You will probably get at least one response like

```
64 bytes from X.X.X.X: icmp_seq=1 ttl=64 time=0.049 ms
```

from *yourself* - e.g. with the "IPv4 Address" you identified previously (in the blue-circled block above). You may also get a response from an address that ends in `.1`, e.g. `172.20.10.1` or `192.168.43.1`, that was identified as the "Default Gateway" in the `ipconfig` output - this is the device that provides the hotspot. Look for a response from an address that is *not* one of those, and make a note of that address. Then, continue from the section *SSH into your Pi*.

2.4   Establish an SSH connection over your personal hotspot network

Once you have identified the address of the Pi, make sure you can reach that address - in a terminal session, run

```
ping Y.Y.Y.Y
```

substituting the IP address you found in place of `Y.Y.Y.Y` above. (If the `ping` process does not stop by itself, you can stop it with `Ctrl+C`.) Make sure you get some response!

Then, verify that you can SSH into your Pi with

```
ssh pi@Y.Y.Y.Y
```

again, substituting the IP address you found in place of `Y.Y.Y.Y`, and use `raspberry` as the password.

On the Pi, run

```
iwconfig wlan0
```

to see the current network configuration.

2.5   Establish a VNC session over your personal hotspot network

As with SSH, you will need to substitute the IP address that you found ("Y.Y.Y.Y") in the address space of the VNC viewer application when you connect to your Pi using your personal hotspot device. Open the VNC viewer and practice this now.

Note: After you have finished this section, you can re-connect using the `ece4313` network in the lab for the rest of this assignment. The `ece4313` network may offer a better network connection than your personal hotspot.

Lab report: Imagine that six weeks from now, you need to use your Pi at home to finish a lab assignment. You don't want to spend an hour or two figuring out how to connect to your Pi at home *before* you can even continue your work! Write a step-by-step tutorial for "future you", with instructions specifically for *your* personal hotspot device, with images at each step that show what it looks like when you have carried out the step correctly. Your tutorial should include instructions to:

1. Start a personal (2.4GHz) WiFi network from your phone or laptop, and to connect your laptop to this network (if it's a phone hotspot). Show a screenshot that demonstrates how you can tell you have done this step successfully.
2. Power on the Pi and have it load the operating system from SD card. Show a photo that demonstrates how you can tell you have done this step successfully. (You can copy your own answer from a previous section.)
3. Identify the address of your Pi your *personal* network, and verify that it responds to a `ping` command at that address. Show a screenshot to demonstrate the steps of identifying the Pi's address, and a photo showing what it look when your `ping` to that address is successful. (Note that each time you connect your Pi to your personal hotspot network, it can have a different address - so you need to show the *steps* to finding that address.)
4. Open a terminal and use SSH to connect to a terminal session on your Pi over your *personal* network. Show a screenshot that demonstrates how you can tell you have done this step successfully.
5. Open a VNC viewer and use VNC to connect to a GUI session on your Pi over your *personal* network. Show a screenshot that demonstrates how you can tell you have done this step successfully.

(Your instructions should not include one-time configuration tasks - for example, you don't need to include the parts about editing the `wpa_supplicant.conf` file, or enabling VNC with `raspi-config`, since you have already done those steps and won't need to do them again.)

Lab report: Upload the contents of your final `wpa_supplicant.conf` file, with your personal network configuration. (In the event that you have to create a new SD card in the future, you will copy this configuration file onto your new SD card so that it can connect to your own *personal* network.)

---

3 Using your Pi

3.1 Practice using the Pi GUI

Open a VNC session on your Pi, and explore the applications that come pre-installed on the Pi.

Click on the Raspberry icon in the top left to see the application menu, and explore the available applications. You can also use the folder icon in the top left to explore the filesystem on the Pi, the globe icon to open a browser, or the terminal icon to open a terminal.

Lab report: Write instructions for "future you", describing how to open each of these applications on the Pi:

- a browser
- a text editor
- a terminal
- the file browser/manager

Also show a screenshot of each application.

3.2 Practice using the Linux terminal

Open an SSH session on your Pi, and practice using the terminal prompt on the Pi.

In this section, we'll learn some basic commands for use on the Pi's Linux-based operating system: `pwd`, `ls`, `mkdir`, `cd`, `nano`, `cat`, `mv`, `rm`, `sudo`, `shutdown`, `reboot`, `exit`. (Later, you'll also learn how to use `scp`.)

The first group of commands - `pwd`, `ls`, `mkdir`, `cd`, `mv`, `rm` - are useful for navigating and manipulating the filesystem.

First, check where you are currently located in the filesystem with the `pwd` ("**p**rint **w**orking **d**irectory") command:

```
pwd
```

Usually, when you open a new terminal window, it will be "in" your home directory - the directory allocated to your user account for storing files and configurations. In this case, the username we are logged in with is `pi` and our home directory is `/home/pi`.

Next, **lis**t the contents of the directory you are in:

```
ls
```

To create a new directory inside our current directory, run `mkdir` and specify a name for the new directory. Try it now; run

```
mkdir lab1
```

Run

```
ls
```

again and verify that you can see the new directory you have just created.

You can **c**hange **d**irectory by running `cd` and specifying the directory you want to change to. For example, to change to the directory you've just created, run

```
cd lab1
```

and then use

```
pwd
```

again to verify your current working directory.

You may have noticed that when you run the `pwd` command, it gives you a full path staring with /, followed by several directory names separated by a / character. Any path that starts with a / (the root of the filesystem) is a *full path* (or it may also be called an *absolute path*). For example, after running the commands above, I would see the following output for `pwd`:

```
/home/pi/lab1
```

When you run commands that involve a file or directory, you can always give a full path, which starts with a / and contains the entire directory tree up until the file or directory you are interested in.
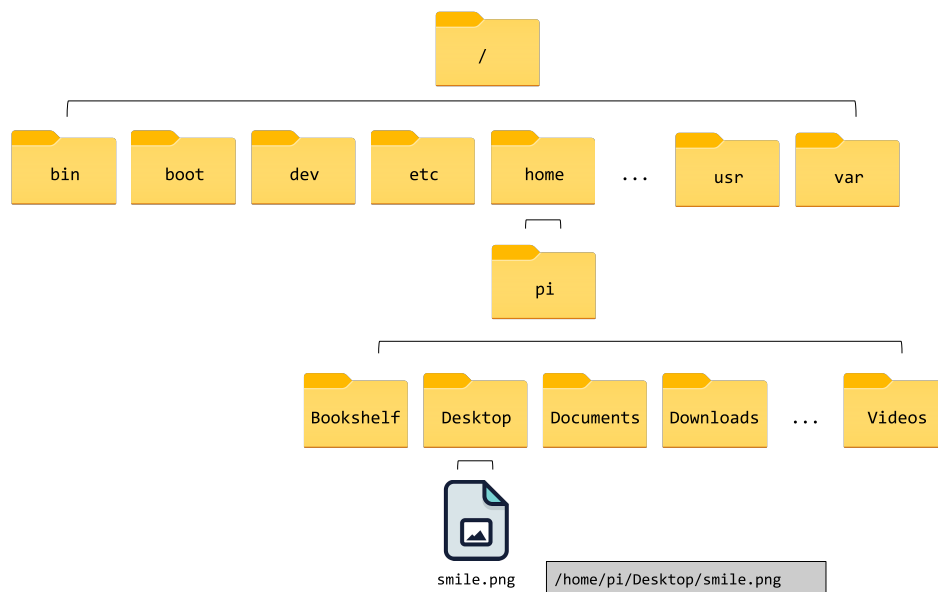
Figure 23: Illustration of the filesystem hierarchy on the Pi, showing how it relates to the *full path* of the `smile.png` file located on the user's Desktop.

For example, you can run

```
cd /home/pi/lab1
```

to navigate to the `lab1` directory . Alternatively, you can give a path that is *relative* to the directory you are in. For example, when I am inside my home directory (`/home/pi`), which has a directory called `lab1` inside it, I can navigate into the `lab1` directory with a relative path:

```
cd lab1
```

or the absolute path:

```
cd /home/pi/lab1
```

Next, we will create a file and edit its contents. Use `cd` to navigate to the `lab1` directory you created. Then we will create a `hello.txt` file and open it for editing in one command, using `nano`, a terminal-based text editor:

```
nano hello.txt
```

Type your name and net ID into this file, then use `Ctrl + O` to write it **o**ut to file, and hit `Enter` to confirm the file name to which to save. Near the bottom of the screen, it should say e.g. "[Wrote 1 line]". Then use `Ctrl + X` to exit `nano` and return to your terminal prompt.

To see the contents of a file, we can print the contents of the file to the terminal output with `cat`:

```
cat hello.txt
```

Instead of typing out the entire command, the terminal supports a useful feature known as tab auto-completion, in which you can start typing out a command or filename, and it will fill in the rest for you when you press the `Tab` key.

Try it now - type

```
cat he
```

and then hit the `Tab` key. Does it fill in the rest of the filename for you? You can then hit `Enter` to see the contents of the file again.

To copy a file, we use `cp`, and give the source and destination file names (or paths) as arguments. For example, if you want to put a copy of `hello.txt` in on your Desktop (`/home/pi/Desktop`), you can run

```
cp hello.txt /home/pi/Desktop/hello.txt
```

To move (or rename) a file, we use the `mv` command:

```
mv hello.txt hello-4313.txt
```

Note that we used a – in the file name - it's better to use – or _ characters rather than spaces in file names.

We can use `rm` to delete a file - for example:

```
rm /home/pi/lab1/hello-4313.txt
```

With `rm`, there is no "Recycle Bin" and no getting back files you've deleted accidentally - so be very, very careful.

You can also remove a directory with the `rm` command, although to remove a directory you'll need to add the `-R` argument:

```
rm -R /home/pi/lab1
```

It's often useful to be able to see and re-run commands you've previously run.

You can use the up arrow and down arrow keys to scroll through your previous commands. For example, if you want to re-run the previous command with a small modification, you would:

- Use the up arrow key to see the previous command,
- Use the right and left arrow keys to position your cursor within the command, and make the desired changes,
- then hit `Enter` to run the modified version.

Or, to see your command history all at once, run

```
history
```

The commands you have run so far did not require any special privileges. However, we'll sometimes need to run commands that are available only to the administrator, called the "root" user. To run a command as the "root" user, we preface the command with `sudo`.

For example, the `reboot` command, which performs a graceful (software) reboot, is only available to the "root" user. If you try to run

```
reboot
```

you will get a "Permission denied" error.

However, you can reboot the Pi by running

```
sudo reboot
```

You'll see the Pi reboot. Wait a few minutes for it to come up, then use SSH again to get a terminal on the Pi.

(As a general tip, if you ever get a permission error on a Linux system, consider whether you should have prefaced the command you ran with `sudo`!)

To end an SSH session without turning the board off, just run

```
exit
```

to disconnect from the board and return to your regular terminal prompt. Note that the Pi is still on and running!

3.3 Transfer files to and from your Pi with VNC

Next, we will learn how to use VNC to transfer files between a laptop and the Pi.

To send a file from your laptop to the Pi, move your mouse near the top of the VNC window, and when the VNC toolbar appears, click on the file transfer button:
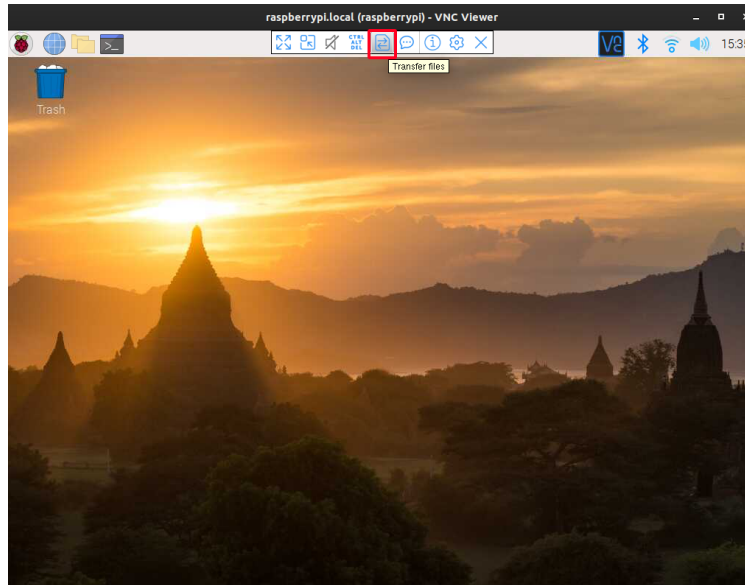

Figure 24: Opening the file transfer dialogue to copy a file from your laptop to your Pi.

Then, click on "Send Files" and select the file you want to transfer to your Pi. Locate the file in the selected location on the Pi, using the file browser/manager.
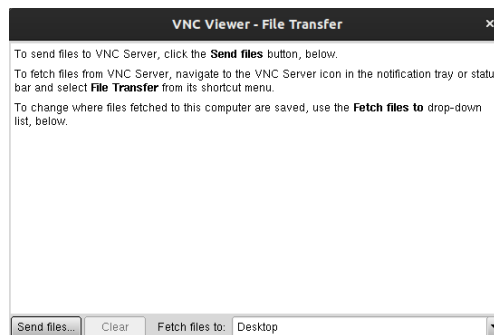

Figure 25: Send file dialogue in VNC.

To send a file *from* the Pi *to* your laptop, click on the VNC icon in the status bar on the Pi. Then, click the menu icon in the corner, and choose the "File Transfer" option.
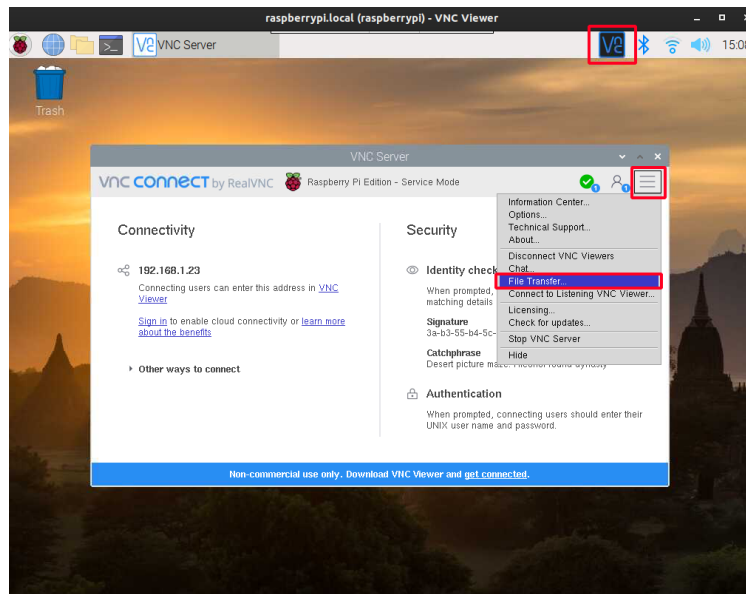

Figure 26: Opening the file transfer dialogue to copy a file from your Pi to your laptop.

3.4   Transfer files to and from your Pi with SCP

We will also learn how to use SCP - the SSH copy utility - to transfer files between a laptop and the Pi. This is useful for when you want to transfer files without opening a VNC session.

In a *local* terminal on your *laptop* (NOT on the Pi - check the terminal prompt to be sure!), find an appropriate location in the filesystem to copy files to, and move to that location (such as your Documents or Desktop folder). (You can use the `pwd`, `ls`, and `cd` commands you just learned in the terminal on Linux, Mac, or Git Bash on Windows to help you locate yourself in the filesystem on your laptop!)

After you have navigated to an appropriate location, you will run a command similar to the following, but replace values as indicated in the explanation below -

```
scp pi@192.168.0.5:/home/pi/hello.txt /Users/ffund/hello.txt
```

to copy the file from `/home/pi/hello.txt` on the Pi to the location `Users/ffund/hello.txt` on your laptop. Enter the password (`raspberry`) when prompted.

Here's a brief explanation of the different parts of the `scp` command:

- `scp` is followed by a space, then its two arguments. The first argument is the source of the file you want to copy, and the second argument is the destination you want to copy to.
- `pi` is your username on the system where the file you want to copy is located
- `@` separates the username and hostname or address
- `192.168.0.5` is the address of the system where the file you want to copy is located. (Alternatively, you can use the name, e.g. `raspberrypi.local`.)
- `:` separates the hostname and the absolute path of the file you want to copy.
- `/home/pi/hello.txt` is the absolute path of the file you want to copy.
- The last argument is the location you want to copy the file *to*. In this example, I used a path: `/Users/ffund/hello.txt`.Alternatively, instead of specifying a path, I could use the shorthand `.` which says "Copy to the directory that I'm running this command *from*."

Note: If you get a "Permission denied" error at this point, make sure you have the necessary privileges to write to the location on your *laptop* that you specified!

Make sure you can locate the files you've just transferred using the file explorer on your laptop.

To transfer the file in the reverse direction, you would simply swap the two arguments to `scp`, e.g.

```
scp /Users/ffund/hello.txt pi@192.168.0.5:/home/pi/hello.txt
```

Lab report: Create a file at `/home/pi/lab1/pi.txt` on your Pi, with your net ID as the file contents. Also create a file file `laptop.txt` at the location of your choice on your laptop, with your net ID as the file contents. Then, prepare an instruction/checklist document for "future you" explaining:

1. how to transfer the file from `/home/pi/lab1/pi.txt` on your Pi, to a location of your choice on your laptop, using VNC. Then, explain how to open the file on your laptop and verify its contents. Show a screenshot of the file open on your laptop, with the contents displayed.
2. how to transfer the file from `/home/pi/lab1/pi.txt` on your Pi, to a location of your choice on your laptop, using SCP. Then, explain how to open the file on your laptop and verify its contents. Show a screenshot of the file open on your laptop, with the contents displayed.
3. how to transfer the `laptop.txt` file from your laptop, to the Desktop on your Pi, using VNC. Then, explain how to open the file in the GUI on your Pi and verify its contents. Show a screenshot of a GUI session on your Pi with the file contents displayed in the text editor.
4. how to transfer the `laptop.txt` file from your laptop, to the Desktop on your Pi, using SCP. Then, explain how to open the file in a terminal-only SSH session on your Pi and verify its contents. Show a screenshot of a terminal session on your Pi with the file contents displayed.

3.5   Safely shut down your Pi

It's important to shut down your Pi "safely" - if you remove power while the Pi is still writing to the SD card, the SD card may become corrupted. Then it would need to be re-created (using the instructions in the Appendix).

When you're finished, use SSH to connect to your Pi again, then perform a graceful (software) shutdown - run

```
sudo shutdown now
```

Wait for the disk activity LED on your Pi to stop blinking completely. Then, carefully disconnect the micro USB power cable from your Pi. Don't pull the micro USB power cable out of the Pi forcefully, since you may break the jack off the board. Instead, apply pressure to the micro USB jack with one hand while you gently remove the power cable with the other.