Linux and Pi:
Getting Acquainted with Linux and the Raspberry Pi

Joshua Vasquez and Sarah Lichtman

December 8, 2013

Contents

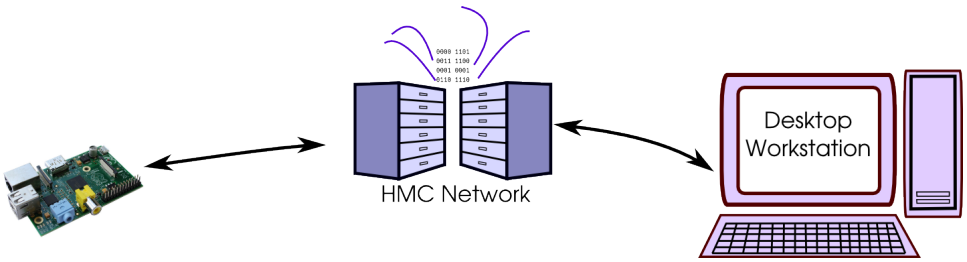
1 Introduction

Welcome (or welcome back!) to Linux, the open-source operating system used by scientists, programmers, and engineers worldwide.
Unlike other operating systems (OSes) such as Windows and Mac OS X, Linux is open source, and fully customizable to numerous applications. For this reason, the Linux variants can be found in numerous places, from computer servers that process large amounts of data, to embedded devices that control robots. For the Raspberry Pi, a reliable, lightweight version of Linux, called *Rasbpian* has been developed to deliver ease-of-use for new users while keeping those hair-raising bugs to a minimum.

2 Getting Started

2.1 Talking to the Pi

A number of ways exist to access the Pi and start writing code. Since the Pi is a computer, we could simply plug in a keyboard, mouse, and screen, and we'd be set. For our labs, though, we've chosen to operate the Pi *headless*. That is, we'll log into a Pi remotely and interact with it from another computer. To connect to a Pi, we've created the following setup:



A Pi is situated near each lab computer and connected to HMC's network. Each of these Pi's has been given a unique static IP address and name, which means you, the user, can talk to a Pi wherever you can connect to HMC's network. For your ease, you can use the lab computer next to the Pi.
To connect to a Pi, you'll login with the common username *Pi*, rather than creating a new user.

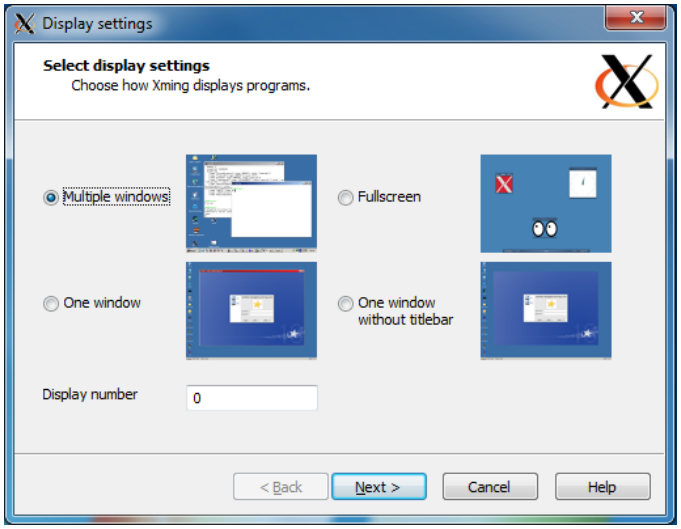
2.1.1 Logging in From a Lab computer:

1. Open *XMing* by typing

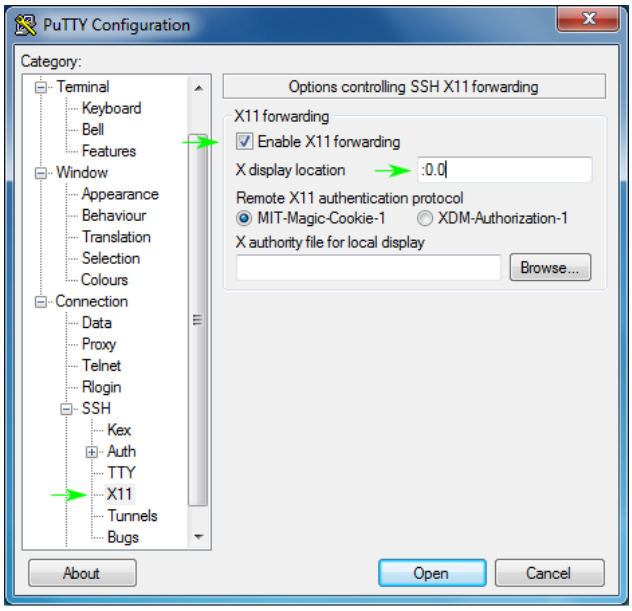
   ```
   xlaunch
   ```

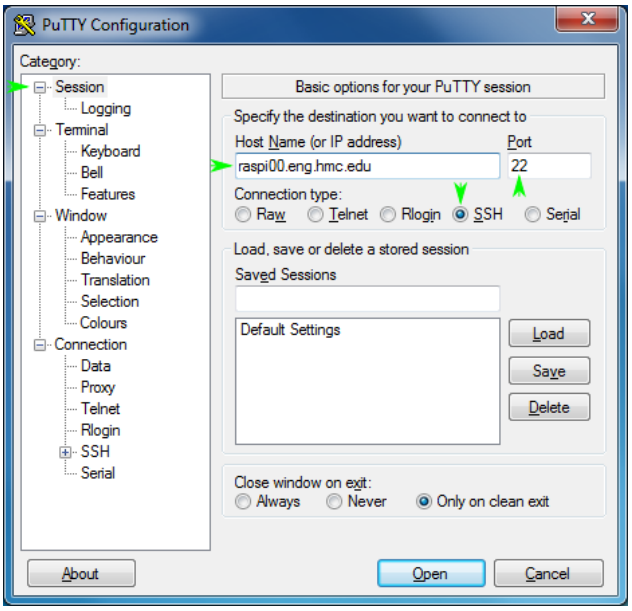
   into the run bar on the start menu.

2. In the settings, select **Multiple Windows**, and choose Display number **0**. Then contiue clicking **Next** for the remaining pages to enter the default settings. Complete these settings with **Finish**.



3. Open PUTTY, a terminal emulator for Windows.





4. In the file tree to the left, scroll down to **SSH**, and under the **X11** tab, select **Enable X11 forwarding** and set the display location to **:0.0** such that it matches display 0 in Xming.

5. To connect to the Pi at your workstation, scroll up to **Session** and enter the static IP of the Pi beside your desk. (They should be in the form of **raspi*xx*.eng.hmc.edu**, where *xx* pertains to the actual pi.

6. Click **Open**, and you will be prompted for a username and password. For the username, choose **pi**. For the password, input

```
passwordHERE
```

2.1.2 Want to use your own computer? Alternative ways to log in:

If you'd rather connect to the Pi through your laptop, rather than a lab computer, enjoying the comfort of your own familiar keyboard, here's what you can do:

1. Ensure that you are connected to HMC's network through either *Claremont-WPA* or through a wired connection
2. The next steps depend on which operating system you're using:
   (a) **Windows:** Download and install PUTTY; then, simply follow the lab setup instruction above.
   (b) **Mac OSX:** Open a terminal. A shortcut can be found in **Applications** → **Utilities** → **Terminal**. While you're in the **Utilities** folder, check that you have XQuartz installed. This program will allow you to have a graphical session with the Pi. If you don't have XQuartz, download and install it from `<http://xquartz.macosforge.org/landing/>`. From here, follow the Linux instructions below.
   (c) **Linux:** Open a terminal. **CTRL** + **ALT** + **T** works on a number of distributions, or however you prefer. In Terminal, enter the following:

   ```
   ssh raspixx.eng.hmc.edu -X
   ```

   where **xx** pertains to the actual pi that you want to connect to. The **-X** enables X11 forwarding, and must be uppercase X (lowercase x disables X11 forwarding).
   It may take a minute to connect. You can hit Ctrl-C to interrupt and cancel. If you get a *Operation timed out* message, try connecting to a different Pi.
   If you are able to connect to the Pi, you may see a message like:

   ```
   The authenticity of host 'raspi00.eng.hmc.edu (134.173.38.34)' can't be established.
   RSA key fingerprint is ea:8b:6b:cc:7a:ba:11:34:2d:ad:e1:44:12:10:3f:cd.
   Are you sure you want to continue connecting (yes/no)?
   ```

   Type yes. Enter your password when prompted. When the Pi is finished loading, you will see

   ```
   pi@raspberrypi ~ $
   ```

   You have sucessfully connected to the Pi!

3 The Linux Environment

When logged in from an ssh session, most of your interactions with the Linux file system will be handled through a terminal interface. The terminal is your command input for navigating through the Pi's file system, running programs, and executing scripts.
In what follows, this tutorial will guide you through basic file manipulation using commands at the terminal.

3.1 Directories

Upon logging in, you are placed at your home directory, generally:

```
/home/your_name
```

Since you are likely logged in as the user pi, your home directory will be located in

```
/home/pi
```

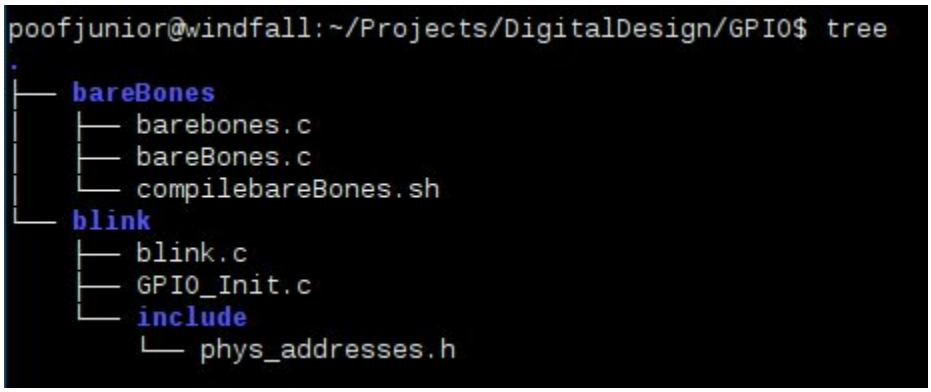
To **l**i**s**t the contents of a directory, type

```
ls
```

and the directories within your current directory will be listed.
If you'd like to see what directory you're currentl working in,

```
pwd
```

will print your working directory.
To **c**hange directories, you have two options. Your can either enter a file path *relative* to your working directory, or you can enter an *absolute* path beginning with your home directory when prepended with the ~ character.
For example, take a look the following file structure below, (output to the terminal window with the **tree** command)



To navigate to the bareBones directory from the GPIO (current) directory, one would type:

```
cd bareBones
```

To do so from the home directory, one would type the entire path starting from the home directory and prepended with the ~ character:

```
cd ~/Projects/DigitalDesign/GPIO/bareBones
```

To navigate back up one directory, type:

```
cd ..
```

If you ever want to go back to the home directory, just type **cd** without any arguments.

Pro Tip: Feeling lazy? Linux on the Pi supports *Tab completion*. In other words, you can simply type part of the file name, press the *tab* key, and the terminal window will auto-complete the remaining text if the file or folder is unique. This trick may save hours of unecessary typing and also guarantee that the file path is spelled correctly.

3.2 File Manipulation

Now that you can navigate up and down directories, let's try creating some of your own. to create a directory, type

```
mkdir temp_folder
```

Let's create a file inside of that folder.

```
cd temp_folder
touch test.txt
```

Now, **ls**ting the directory contents will display

```
test.txt
```

To copy-and-paste, use the cp command, followed by two arguments: The first argument is the name of the file you'd like to copy; the second file is the name that the copied file will have. For example:

```
cp test.txt otherTest.txt
```

will copy **test.txt**'s contents into a new file named **otherTest.txt**. To remove a file, use the **rm** command, followed by the name of the file.
To copy and entire folder (with its contents), you must add the **-r** flag. For example, beginning from the directory above the **temp_folder** directory, the following command

```
cp -r temp_folder another_temp_folder
```

will copy the folder temp_folder and its subcontents into the folder **another_temp_folder**.
Removing an empty directory is simple. Simply use the command **rmdir**.
Removing a directory with files inside must be handled by the **rm** command with the **-r** argument. Thus,

```
rm -r another_temp_folder
```

will remove **another_temp_folder** and its contents.
Finally, if you're just interested in **mo**v**e**ing a file from one location to another, you can use the **mv** command. The syntax is

```
mv <what> <where>
```

where the first argument is the file (or folder) that you're moving, and the second argument is the filepath of the new location. Take note: if you finish the filepath with the name of a new file, the original file will be both moved to the new location and have that new name. Be careful with this command. It's easy to overwrite another file by simply **mo**v**e**ing a file with the same name to the same location. It's also easy to overwrite a file by **mo**v**e**ing a file to a new location and giving it a new name of a file in that same location.

That's it for file manipulation! For anything more complicated, may the internet be your resource.

4 Getting Comfortable with DDD

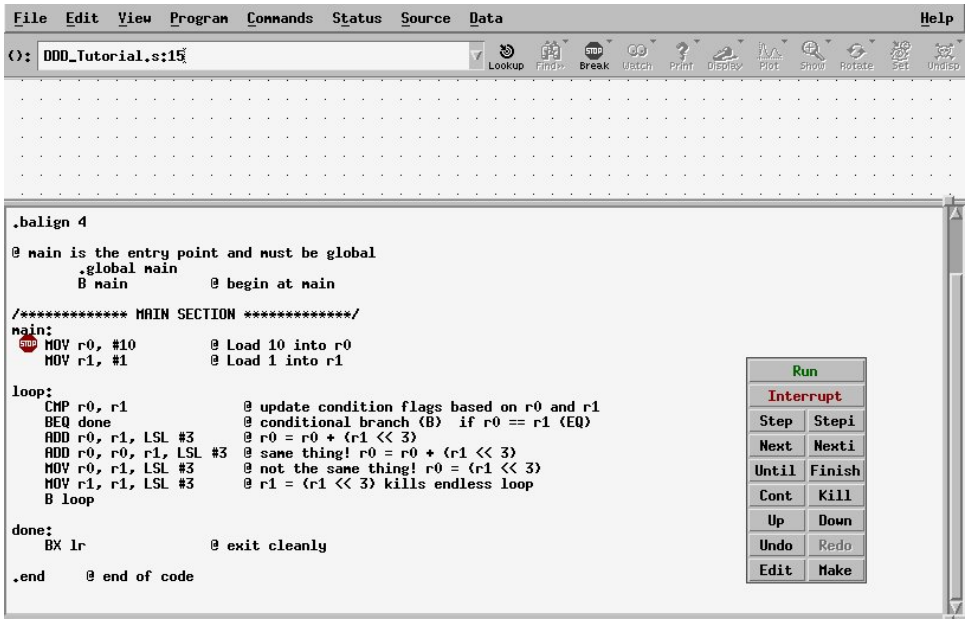
Data Display Debugger, a.k.a.: DDD, is a graphical debugging tool for debugging source code. DDD is actually just a nice graphical interface. DDD actually calls the program GDB (the GNU Project Debugger) to do the actual debugging. DDD lets you step through your source code (be it C or ARM Assembly), add breakpoints, and watch the value of your registers change instruction-by-instruction. For the purposes of the assembly lab assignment, to verify that you're code is behaving as you expect, you actually *have* to step through it with DDD, since no other manner in ARM Assembly lets you easily output data to the terminal.
Note: To make your code "debuggable" with DDD, you'll actually have to compile it with the *-g* flag.

1. Let's begin by navigating to the example folder **DDD_Tutorial**. Compile this code by typing **make** in the Terminal. Now, within this directory, open DDD with the command **ddd**. DDD should launch in a new window.

2. Now open the file **DDD_Tutorial** from
   File → *Open Program* If you compiled the code with the **-g** flag (the Makefile used by **make** should do this for you), the source code will appear in the window below. Before running the code, open the register listing from *Status* → *Registers...*

3. Finally, begin debugging your code by adding a breakpoint at the first line of code. To do so, double-click on the whitespace to the left of the line of interest, and a *stop* will appear by that line. (Here, it's just after **main**.) In this manner, the code will freeze, allowing you to step through subsequent lines. To delete a breakpoint later, simply right-click on it and select *Delete Breakpoint*.

4. You're now set to step through your code to verify that it's behaving as you expect! To step through your code, select *Stepi* on the panel. As you continue to step through your code, registers in the register window should change accordingly.



5. If you want to rerun your code, you must first *Kill* it in the panel window before running it again.

An Important Note: Each time you make changes to your code, you must first close DDD, edit it with a text-editor, and recompile it once more with the **-g** flag.

5 Conclusion

That's about it for the Linux Tutorial. Linux is a highly-customizable operating system. If you'd like to continue giving embedded Linux a shot, consider picking up a Raspberry Pi of your own and moving forward with your very own digital electronics projects. If you'd like to give a full-size operating system a try, consider installing it on your computer. Linux currently arrives in numerous *distributions*, each with its own distinct look and feel. As you continue your journey through digital electronics, we salute you in your endeavor to transform your computer into a tool that can more distinctly serve your specific needs.

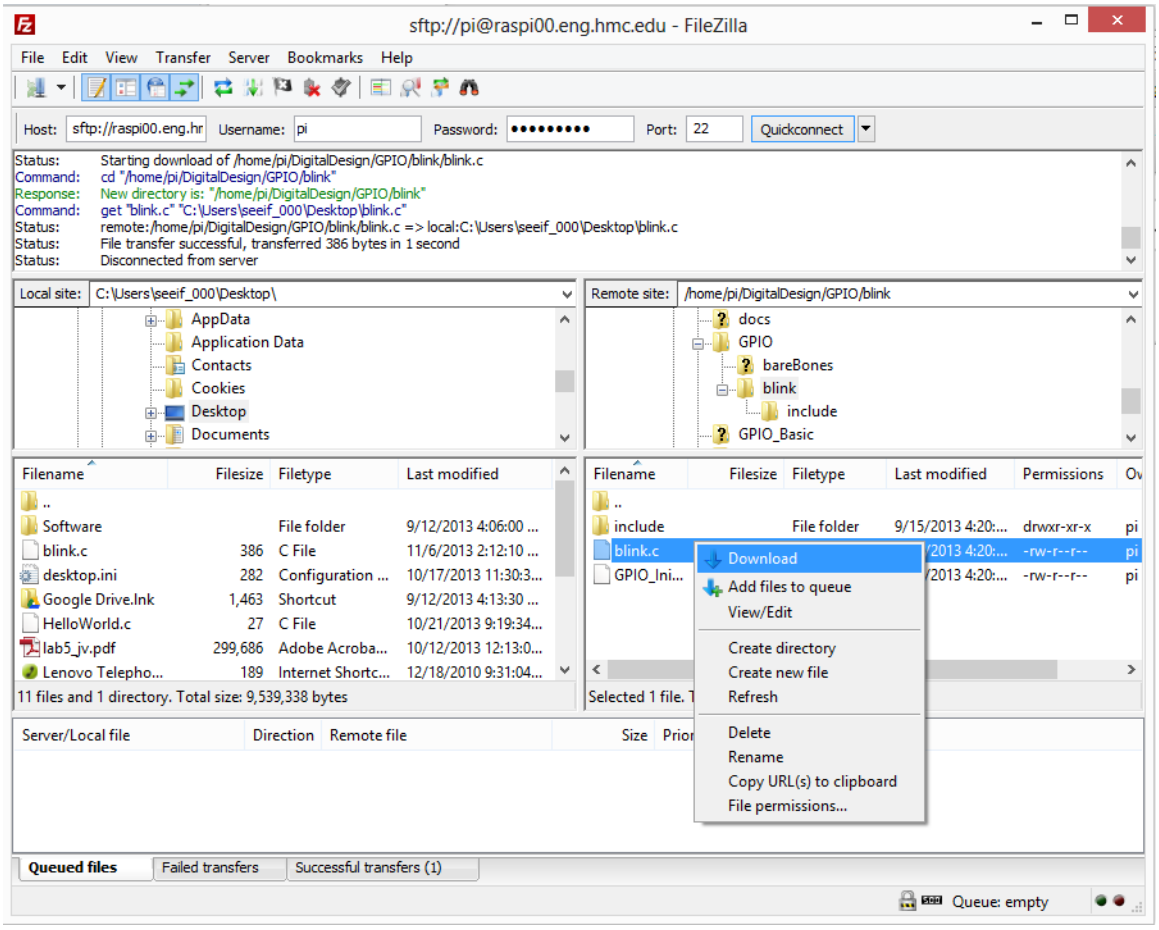
Finally, consider subscribing to **linux-l@hmc.edu**, HMC's official mailing list for all things Linux related.

6 Appendix

6.1 Transferring Files Between the Pi and another Computer

Transferring files over a remote session is actually fairly counterintuitve. While the good-old Copy-Paste can get the job done for some tasks, here's the full guide on getting this done remotely.

1. If using your own computer, install FileZilla. Otherwise, open FileZilla from a lab computer



   If using your own computer, ensure you have a connection to HMC's network (wired or Claremont-WPA).

2. To access the files, enter the following:

   host: raspi00.eng.hmc.edu
   username: pi
   password: raspberry
   port: 22

3. From here, you should be able to see the file structure of the Pi (including hidden files!) on the right. From here, you may transfer files to the home computer by right-clicking on the file (on the right) and selecting *Download*.