ECE471: Embedded Systems – Homework 2

Raspberry Pi and Linux

Due: Friday, 16 September 2022, 10:00am

This Homework is meant to get you started with the Raspberry Pi so that you will be prepared for future homework assignments.

1. **Get Linux Working**

   - Install some form of Linux onto your Raspberry Pi. I recommend Raspbian, which has recently been renamed to "Raspberry Pi OS". You may have obtained an SD card that already has Linux installed, or you can download images and find instructions here: `https://www.raspberrypi.com/software/`
   - Power up the Pi and ensure you can login. The most straightforward way to do this is have a USB keyboard and a TV/monitor connected via HDMI. Be sure the SD card with Linux is in the slot, and connect the power via USB and it should boot up. There are many tutorials on how to do this online.
   - You will probably be booted into a GUI interface. If it takes you to text, you can login at the prompt. Otherwise, login. Raspbian the default login is `pi` and default password is `raspberry`.
   - You can change the password with the `passwd` command. You can also add users with the `adduser` command, though you probably need to be root to do that: `sudo adduser`
   - If the GUI doesn't start by default, often you can use `startx` after logging in to get a graphics interface.
   - If you have trouble with any of these steps and need help, let me know as soon as possible!

2. **Copy the Assignment to your Raspberry Pi**

   - Download the code from:
     `https://web.eece.maine.edu/~vweaver/classes/ece471/ece471_hw2_code.tar.gz`
   - There are various ways you can do this. Using a USB key or the network are probably the easiest.
     - USB key
       - Ideally you just plug the USB key in and if you are running the GUI it will pop up a window and let you find the file that way.
       - If you are in text mode the USB key will appear somewhere in the /media directory tree.
       - If you're really unlucky you might have to mount the USB key by hand, something like
         `sudo mount /dev/sda1 -o rw /media/usbkey`
       - To copy `ece471_hw2_code.tar.gz` to your home directory type something like the following (it will vary based on exactly where the USB key got mounted)
         `cp /media/usbkey/ece471_hw2_code.tar.gz ~`
       - When you are done with the usbkey, **before** removing it you have to unmount it; you should be able to do this from the GUI, or else you can manually
         `sudo umount /media/usbkey`

  - Network
    - If you have your Pi on the local network, you can log in with ssh and you can copy files using `scp`
    - There are also a number of GUI scp or sftp clients you can use to connect.
    - If you are using a Mac, you can look into installing netatalk in order to access your Pi over the network.

3. **Unpack the homework files**

   Uncompress/unpack it with the command

   ```
   tar -xzvf ece471_hw2_code.tar.gz
   ```

4. **Build the C files**

   - Change into the ece471_hw2_code directory

     ```
     cd ece471_hw2_code
     ```

   - Run `make` to build the code.
   - Run `./hello_world` and it should run!

5. **Modify the `hello_world.c` file** (5 pts)

   - Change the file so the output is 10 lines, each line looking something like this:

     ```
     #1:  ECE471 MY_MESSAGE
     ```

     Where the number after the # increments each line, and MY_MESSAGE is any message you want to print. Start numbering from 1. Full credit requires using some sort of loop (not just cut/paste).
   - You can use any text editor you want to do this coding. A simple one that is available on the Pi is `nano`, you can start it by running `nano hello_world.c`
   - Once you have updated the code, you can simply run `make` and it should recompile your code. Then test by running `./hello_world`
   - Be sure to comment your code! Also fix all compiler warnings!

6. **"Something Cool"** (1 pt)

   Copy your working code on top of the `something_cool.c` file.

   ```
   cp hello_world.c something_cool.c
   ```

   Then do one of the following.

   - Change the number of times your printing code loops based on a command line argument. The `atoi()` or `strtod()` functions might be useful for this.
   - Modify your `hello_world.c` file further to print the lines in different colors.
     HINT: Look up "ANSI escape codes"
   - (harder) Instead of printing everything one color, print each line a different color.
   - (even harder) Print each individual letter of the output a different color.

7. **Answer the following questions** (4pts total)

   Short answers are fine. Put your answers in the README file using a text editor, it will be automatically included in the submission process.

   (a) If you want to know more about the `ls` program, what command can you run?

   (b) What does the `-a` option to the `ls` program do?

   (c) List one reason why C is often used on embedded systems.

   (d) What method are you using to connect to your pi? (monitor/keyboard, network, serial, etc.?) How are you copying files back and forth?

8. **Submit your work**

   - Be sure your name and answers to questions are in the README file.
   - Run `make submit` which will create a `hw2_submit.tar.gz` file containing `README`, `Makefile`, `hello_world.c` and `something_cool.c`.
     You can verify the contents with `tar -tzvf hw2_submit.tar.gz`
   - e-mail the `hw2_submit.tar.gz` file to me by the homework deadline. Be sure to send the proper file!