Course notes, module 2 week 6
Course introduction, linux fundamentals, linux desktop environment

Kjeld Jensen kjen@sdu.dk

Agenda

1. Recap of the exercises from last module
2. Preparation for next module
3. LEO1 kit
4. Networking
5. Embedded linux
6. Exercises
7. Optional exercises

2. Preparation for next module

The module next week will be a continuation of this module. We will contine working with the Raspberry Pi (RPi), Raspberry Pi OS, networks etc.

Please remember to bring the full LEO1 kit to all remaining modules of the course.

6. Exercises

6.1 Install Raspberry Pi OS

The aim of this exercise is to write a recent version of the Raspberry Pi OS lite to the MicroSD card.

Install rpi-imager on your linux desktop environment:

```
$ snap install rpi-imager
```

Now run rpi-imager to write Raspberry Pi OS to a MicroSD card.

`CHOOSE OS`
: You should choose “Raspberry Pi OS (other)” and then “Raspberry Pi OS Lite (32-bit)”.

`CHOOSE STORAGE`
: Be careful to select the correct MicroSD card so that you do not overvrite another disk.

Do not install the MicroSD in the RPi yet.

6.2 Configure Raspberry Pi OS networking

The aim of this exercise is to ensure that you can access the raspberry pi via the ethernet interface after boot.

Mount the MicroSD card under your linux desktop environment.

In the boot partition create an empty file named ssh. This will enable remote ssh after boot. Please replace user with your username in the below example:

```
$ touch /media/user/boot/ssh
```

To setup a static IP address add the below text to the bottom of the `/etc/dhcpcd.conf` file

```
$ sudo nano /media/user/rootfs/etc/dhcpcd.conf
```

```
interface eth0
static ip_address=10.0.0.10/24
static routers=10.0.0.1
static domain_name_servers=8.8.8.8 8.8.4.4
```

Remember to use `SHIFT-CTRL-V` to paste into the terminal

Please note that the domain name servers `8.8.8.8` and `8.8.4.4` are google public servers. You may want to switch to more appropriate servers depending on your internet connection.

Press `CTRL-X Y <enter>` to save the file

Then unmount the two MicroSD card partitions

```
$ sudo umount /media/user/boot
$ sudo umount /media/user/rootfs
```

6.3 Power on the Raspberry Pi

Insert the MicroSD card on the RPi. Please notice that you can not insert the MicroSD card and then place the RPi in the enclosure. It has to be inserted in the enclosure before installing the MicroSD card.

Connect the RPi to your linux desktop environment wia ethernet

Power up the RPi by plugging in the USB-C connector

Next to the USB-C connection on the enclosure end you will find a small window with a red and a green LED. The red LED should light up solid indicating a good power supply and the green LED should flash during disk activity in the boot process.

6.4 Connect to the Raspberry Pi

On your linux desktop environment go to Settings – Network. Create a new Wired profile (click +). Name it “RasPi” and select the MAC address of your ethernet port. Set the IPv4 configuration to Manual and key in 10.0.0.1 as IP address and 255.255.255.0 as netmask. Do not add a gateway

Now in a shell on your linux desktop environment (press ALT-CTL-T) run the command:

```
$ ssh pi@10.0.0.10
```

Type yes to `accept` to connect to this new host

Use the password: `raspberry`

If everything wen't well, you are now remotely logged into the RPi. If not then you will have to troubleshoot until it works ;-)

Change the password to something else:

```
$ passwd
```

6.5 Power off the Raspberry Pi

When you wish to power off the RPi please don't just disconnect the power. Raspberry Pi OS uses a journaling file system, but still it may corrupt of power is lost. Therefore you should login to the RPi via ssh and perform this:

```
$ sudo poweroff
```

Then wait until the green LED stops flashing before disconnecting the USB-C power.

6.6 Raspberry Pi internet access

The purpose of this exercise is to establish internet access from the RPi by routing packets from the RPi to the internet connection of you linux desktop environment. In this exercise the internet connection will likely be your connection to SDU Wifi.

Currently your RPi is connected to your linux desktop environment via a private subnet over ethernet. In the file `/etc/dhcpcd.conf` you have defined a static router to be `10.0.0.1` which is also the IP address of your ethernet port in your linux desktop environment.

Run this command to se a list of network interfaces. Note down the name of the interface which provides internet access:

```
$ ifconfig
```

In order to establish internet connection from the RPi you need to tell you linux desktop environment to route all TCP/IP packets from to your computers' internet connection. First enable IPv4 traffic forwarding:

```
$ sudo sysctl -w net.ipv4.ip_forward=1
```

Then configure nat to route packets from the RPi to your computers internet connection. Replace `wlan0` by the interface name you noted down:

```
$ sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

Now ssh into your RPi and try to ping `google.com`

6.7 Update Raspberry Pi OS

With internet access to the RPi it is time to update any upgraded packages installed on the Raspberry Pi OS:

```
$ sudo apt update
$ sudo apt upgrade
```

Rerun the last command again to ensure that your Raspberry Pi OS is fully updated

6.8 Configure the Raspberry Pi

Run the below command and look into the various configuration possibilities that the RPi has:

```
$ raspi-config
```

6.9 Install a webserver on the Raspberry Pi

Run the command below to install the Apache2 webserver on the RPi:

$ sudo apt intall apache2

Then on your linux desktop environment use a web browser and visit `http://10.0.0.10` which is the RPi webserver default page

7. Optional exercises

The below exercises are not required and are considered for linux proficiency level 2 and 3 only. You may choose freely between them

- Create a shell script on your linux desktop environment that automates the routing setup in exercise 6.6.
- Create a shell script named `doihaveinternet` that outputs `yes` if your RPi has internet access and `no` if not.
- Create a shell script named `aremypackagesupdated` that outputs `yes` if all your RPi packages are updated and `no` if not.
- On the RPi use iptables to forward all https requests (port 443) to http (port 80) Then try to access `https://10.0.0.10` and you should be able to see the Apache2 default page despites it is currently only serving port 80.
- On your linux desktop environment use iptables to forward all https requests to sdu.dk
- Try changing the RPi DNS server to `10.0.0.1` and set up a DNS relay on your linux desktop environment so that DNS requests are relayed through your linux desktop environment rather than routed directly to an external DNS server.