

Gawun Kim
Professor Macias
April 12 2018
Lab 1
Partner : Ryan Paulos, Deboye Sakho

Getting Familiar with the Rpi/USB system

In this lab, we used Linux system. The lab was pretty straightforward rather than Windows and Mac and we were able to deal with this device easily. We followed each of steps that is below.

First, we connected the Rpi/USB system to the Linux machine and installed minicom. It was available to connect them by using the commands: “sudo minicom -D /dev/ttyUSB0” or “sudo minicom”.

Second, once the computer was connected to the device and typed the command line, the screen was changed and it was only possible to give a command when inserting “(CTRL + A) + (some alphabet)”. “(CTRL+A) + Z” was a manual to be able to show all the commands. Based on this experience, we changed the setting by using “(CTRL + A) + O” and went into “Serial Port Setup”. In this option, would be able to change the setting and especially, we hit ‘F’ so that Hardware Flow Control was turned off and then returned the regular session by using double hits of ESCAPE.

Third, after the above steps, it allowed us to login into the system. We typed “pi” in username and raspberry in password. When we typed “df”, we were able to see like the below

Before expanding, the available space was 3.7 Gb. It could be checked from the line for /dev/root

```

Password:
Last login: Fri Mar 18 09:17:11 UTC 2016 on tty1
Linux raspberrypi 4.1.19+ #858 Tue Mar 15 15:52:03 GMT 2016 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ df

```

filesystem	1K-blocks	Used	Available	Use%	Mounted on
dev/root	3747352	3419224	118056	97%	/
evtmpfs	218416	0	218416	0%	/dev
mpfs	222688	0	222688	0%	/dev/shm
mpfs	222688	4512	218176	3%	/run
mpfs	5120	4	5116	1%	/run/lock
mpfs	222688	0	222688	0%	/sys/fs/cgroup
dev/mmcblk0p1	61384	26176	35208	43%	/boot
mpfs	44540	0	44540	0%	/run/user/1000

```

pi@raspberrypi:~$ df --block-size=MB

```

filesystem	1MB-blocks	Used	Available	Use%	Mounted on
dev/root	3838MB	3502MB	121MB	97%	/
evtmpfs	224MB	0MB	224MB	0%	/dev
mpfs	229MB	0MB	229MB	0%	/dev/shm
mpfs	229MB	5MB	224MB	3%	/run
mpfs	6MB	1MB	6MB	1%	/run/lock
mpfs	229MB	0MB	229MB	0%	/sys/fs/cgroup
dev/mmcblk0p1	63MB	27MB	37MB	43%	/boot
mpfs	46MB	0MB	46MB	0%	/run/user/1000

```

pi@raspberrypi:~$

```

After expanding, the available data is 7.6 Gb.

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/root	7511124	3420572	3729916	48%	/
devtmpfs	218416	0	218416	0%	/dev
tmpfs	222688	0	222688	0%	/dev/shm
tmpfs	222688	4512	218176	3%	/run
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	222688	0	222688	0%	/sys/fs/cgroup
/dev/mmcblk0p1	61384	26176	35208	43%	/boot
tmpfs	44540	0	44540	0%	/run/user/1000

When we typed “sudo raspi-config”

1 Expand Filesystem	Ensures that all of the SD card s s
2 Change User Password	Change password for the default u u
3 Boot Options	Choose whether to boot into a deses
4 Wait for Network at Boot	Choose whether to wait for networor
5 Internationalisation Options	Set up language and regional setttt
6 Enable Camera	Enable this Pi to work with the R R
7 Add to Rastrack	Add this Pi to the online Raspberer
8 Overclock	Configure overclocking for your P P
9 Advanced Options	Configure advanced settings
0 About raspi-config	Information about this configuratat

Above these options, we would be able to change the setting of the device. There are ten of different options: Expanding Filesystem, Change PW, Boot options, Wait for Network, Internationalisation Options, Enable Camera, Add to Rastrack, Overclock, Advanced Options, and About raspi-config. Specifically, we explored three other options.

We would change the password so that it could keep its own security and it did not allow other users to use this device. In section 5(internationalisation Options), user could change the language and regional setting in computer. Also, in the USB, there were some ports to be able to connected to some other device. The camera would be available for the camera and the setting can be arranged from the section 6 (Enable Camera).

We mainly dealt with Expanding Filesystem. After that we expanded Filesystem and then it had been rebooted when we hit finish botton.; we should login again with the account. The available data was expanded from 3.7Gb to 7.6 Gb in /dev/root. Personally, I think the initial total data was 4 Gb and total expanded data was 8 Gb.

When we confirmed the version of java in the device, we used the command which is “java -version”.

pi@raspberrypi:~\$ java -version
java version "1.8.0_65"
Java(TM) SE Runtime Environment (build 1.8.0_65-b17)
Java HotSpot(TM) Client VM (build 25.65-b01, mixed mode)

The version was “1.8.0_65”

When we typed “top” on the command line, the below box was shown on the page.
The memory on the system was 445376 total.

```
un@gawun-90X3A: /dev
top - 09:29:42 up 3 min, 3 users, load average: 0.20, 0.29, 0.14
Tasks: 83 total, 1 running, 82 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.7 sy, 0.0 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 445376 total, 178852 used, 266524 free, 14188 buffers
KiB Swap: 102396 total, 0 used, 102396 free. 105508 cached Mem

  PID USER      PR  NI   VIRT    RES    SHR  S  %CPU  %MEM     TIME+ COMMAND
 637 pi        20   0   5096    2472    2144 R   1.0   0.6   0:00.18 top
 506 pi        20   0  78772   22760   19436 S   0.3   5.1   0:01.99 lxpanel
    1 root       20   0    5416    3844    2784 S   0.0   0.9   0:05.06 systemd
    2 root       20   0         0         0         0 S   0.0   0.0   0:00.00 kthreadd
    3 root       20   0         0         0         0 S   0.0   0.0   0:00.00 ksoftirqd/0
    4 root       20   0         0         0         0 S   0.0   0.0   0:00.00 kworker/0:0
    5 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 kworker/0:0H
    6 root       20   0         0         0         0 S   0.0   0.0   0:00.00 kworker/u2:0
    7 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 khelper
    8 root       20   0         0         0         0 S   0.0   0.0   0:00.00 kdevtmpfs
    9 root       0 -20         0         0         0 S   0.0   0.0   0:00.01 netns
   10 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 perf
   11 root       20   0         0         0         0 S   0.0   0.0   0:00.00 khungtaskd
   12 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 writeback
   13 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 crypto
   14 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 bioset
   15 root       0 -20         0         0         0 S   0.0   0.0   0:00.00 kblockd
```

At that time, 83 of tasks were currently on our system. 1 was running and 82 was sleeping.

When we typed the command(cd /boot/XFER), we would be able to go into the folder.
There were some of files or folder inside. The files regarding that are below.

```
pi@raspberrypi:/boot/XFER$ ls
bashrc  exrc  j8header-b-plus.png  newuser  SETUP  wiringPi
C       HexBug  Java                packages  TODO.txt
```

Also, we changed directories to /boot/XFER/packages and explored the files into that folder.

For the last step, we run cowsay command and we were able to notice this output that is below.
pi@raspberrypi:~\$ cowsay Hello Nick Give me an A!

```
< Hello Nick Give me an A! >
```

```

      ^  ^
     /    \
    (oo)\_____)
    ( )\        )\/
    ||----w |
    ||     ||
```

After all the lab, we finished it with the command(sudo shutdown -h now) and it was able to go out.
pi@raspberrypi:~\$ sudo shutdown -h now
[513.346434] reboot: Power down