

Ubuntu LTS

Emulation Using QEMU

ELECTGON
www.electgon.com
ma_ext@gmx.net

19.04.2018



Contents

1	Introduction	1
2	Emulation with QEMU	1
3	Prepare Packages	3

Abstract

Ubuntu is one of linux distributions that has many long term releases supporting your work level. As we are targeting Embedded Systems, an Ubuntu release was available at time of writing this tutorial. It is ubuntu 14.04 LTS with code name Trusty Tahr. This release is what we are going to demonstrate in this tutorial how to build it on ARM processor, namely on RaspberryPi 2 board. We will use QEMU as an emulator to show needed stpes for working with this ubuntu version.

1 Introduction

Many linux distributions have been produced each of which is adding specific or tiny features for specific users. The fact is linux is a powerful operating system at any computer level starting from embedded systems up to servers. For the sake of embedded systems, choosing linux with minimal features and hardware drivers modules is best option specially for real time specifications.

Many linux lightweight versions are available and can be utilized for systems with limited resources. But what is meant here for embedded system is to find distribution with minimal features to avoid any overhead caused by other packages or applications. Whenever any additional feature is needed, system developer can simply install it which gives full control for the developer to tailor his system according to his application.

It is important then to find suitable linux distribution that can fulfill embedded development conditions. You can find minimal version for any linux distribution in the following link

<https://wiki.ubuntu.com/Releases>

In this link you should find list of current and future releases for ubuntu. Note that it is better to work with releases that are appended with LTS, as LTS denotes for Long Term Support (at least 5 years)

<http://cdimage.ubuntu.com/releases/>

2 Emulation with QEMU

After downloading suitable version of the Ubuntu LTS (ubuntu-trusty.img in our tutorial), we need to expand size of this image

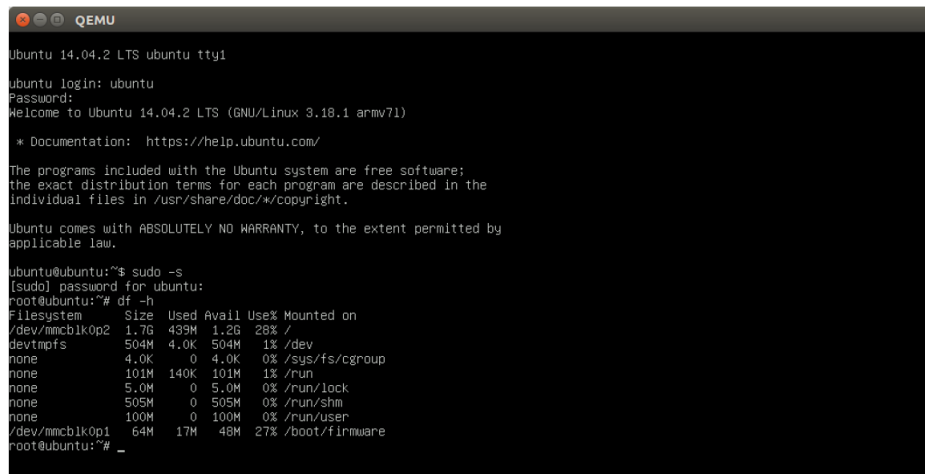
```
$qemu-img resize 2015-02-16-raspbian-wheezy.img +2G
```

We can start emulation in QEMU using the following command

```
$qemu-system-arm -kernel vexpress_04_zImage -append "root=/dev/mmcblk0p2 rw" -m 1G -cpu cortex-a9 -M vexpress-a9 -sd 2015-04-06-ubuntu-trusty.img
```

Installing ubuntu-trusty.img on any machine will be limited to the image size. We need then to expand the image to the available disk size. This can be done manually as follows.

At the beginning, you can run `dh -h` to see disk partitions distributions



```

Ubuntu 14.04.2 LTS ubuntu tty1
ubuntu login: ubuntu
Password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.18.1 armv7l)

 * Documentation:  https://help.ubuntu.com/

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu:~$ sudo -s
[sudo] password for ubuntu:
root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  1.7G  439M  1.2G  28% /
devtmpfs        504M  4.0K  504M   1% /dev
none            4.0K   0  4.0K   0% /sys/fs/cgroup
none           101M  140K  101M   1% /run
none            5.0M   0   5.0M   0% /run/lock
none           505M   0   505M   0% /run/shm
none           100M   0   100M   0% /run/user
/dev/mmcblk0p1   64M   17M   48M  27% /boot/firmware
root@ubuntu:~#

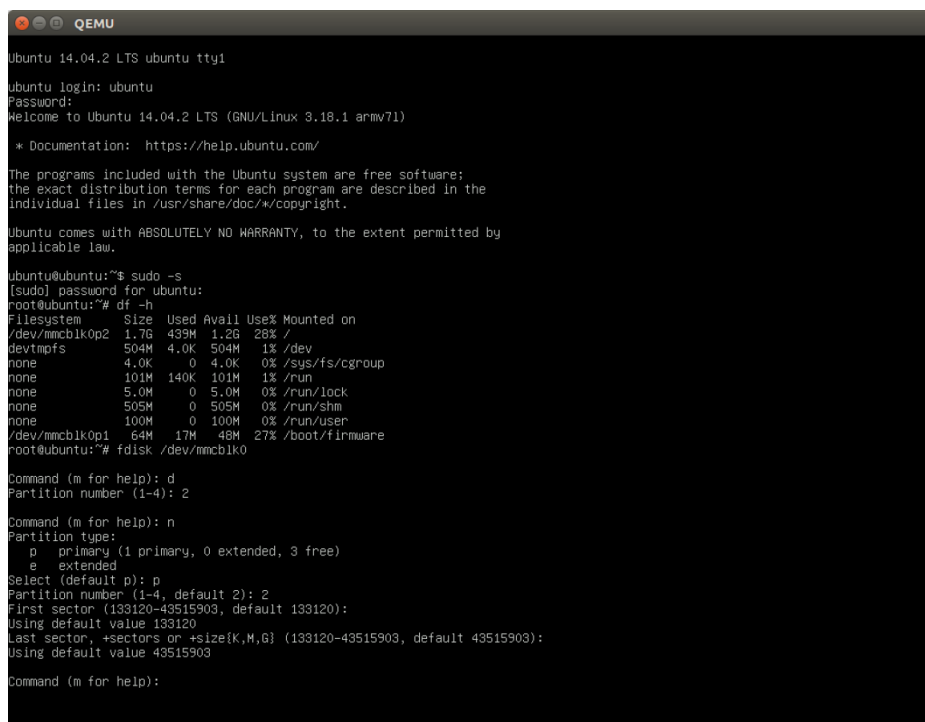
```

Figure 1: Check Disk Partitions

There are no basic specific utilities included, specifically no automatic root resizer. However, it's not hard to do manually. Once booted:

```
$sudo fdisk /dev/mmcblk0
```

Delete the second partition (d, 2), then re-create it using the defaults (n, p, 2, enter, enter), then write and exit (w). Reboot the system.



```

Ubuntu 14.04.2 LTS ubuntu tty1
ubuntu login: ubuntu
Password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.18.1 armv7l)

 * Documentation:  https://help.ubuntu.com/

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu:~$ sudo -s
[sudo] password for ubuntu:
root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  1.7G  439M  1.2G  28% /
devtmpfs        504M  4.0K  504M   1% /dev
none            4.0K   0  4.0K   0% /sys/fs/cgroup
none           101M  140K  101M   1% /run
none            5.0M   0   5.0M   0% /run/lock
none           505M   0   505M   0% /run/shm
none           100M   0   100M   0% /run/user
/dev/mmcblk0p1   64M   17M   48M  27% /boot/firmware
root@ubuntu:~# fdisk /dev/mmcblk0

Command (m for help): d
Partition number (1-4): 2

Command (m for help): n
Partition type:
   p  primary (1 primary, 0 extended, 3 free)
   e  extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (133120-43515903, default 133120):
Using default value 133120
Last sector, +sectors or +size{K,M,G} (133120-43515903, default 43515903):
Using default value 43515903

Command (m for help):

```

Figure 2: Partitioning Harddisk

when system finishes rebooting, we can make resizing using:

```
$sudo resize2fs /dev/mmcblk0p2
```

```

root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  1.7G  466M  1.1G   30% /
devtmpfs        504M   4.0K  504M    1% /dev
none            4.0K   0     4.0K    0% /sys/fs/cgroup
none           101M  140K  101M    1% /run
none            5.0M   0     5.0M    0% /run/lock
none           505M   0    505M    0% /run/shm
none           100M   0    100M    0% /run/user
/dev/mmcblk0p1   64M   17M   48M   27% /boot/firmware
root@ubuntu:~# resize2fs /dev/mmcblk0p2
resize2fs 1.42.9 (4-Feb-2014)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/mmcblk0p2 is now 5422848 blocks long.

root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p2  21G  468M   20G    3% /
devtmpfs        504M   4.0K  504M    1% /dev
none            4.0K   0     4.0K    0% /sys/fs/cgroup
none           101M  140K  101M    1% /run
none            5.0M   0     5.0M    0% /run/lock
none           505M   0    505M    0% /run/shm
none           100M   0    100M    0% /run/user
/dev/mmcblk0p1   64M   17M   48M   27% /boot/firmware
root@ubuntu:~# _

```

Figure 3: Resize FileSystem Partition

Because there is not always enough Random Access Memory (RAM) available for compilation processes, it is a good idea to use a small disk partition as swap space. There is no swap partition/file included. If you want swap, it's recommended you do:

```
$ sudo apt-get install dphys-swapfile
```

You should have a (resized) SD card at least 4GB, because by default it will ask to create a ~2GB swapfile. Previous operation may take about 10 minutes to be finished.

3 Prepare Packages

To start installing any package, a defect has to be fixed first. It is a source file that direct system where to find and install packages. We don't need this direction as we will install packages by apt-get. Without fixing this file, you will get an error message like: "Failed to Fetch http://ppa.launchpad....." when installing any package.

So to fix it,

- go to directory /etc/apt
- open file sources.list
- comment the notified failed file by adding # at line start.

If the notified failed file is not in sources.list, open directory /etc/apt/source.list.d it may be there.

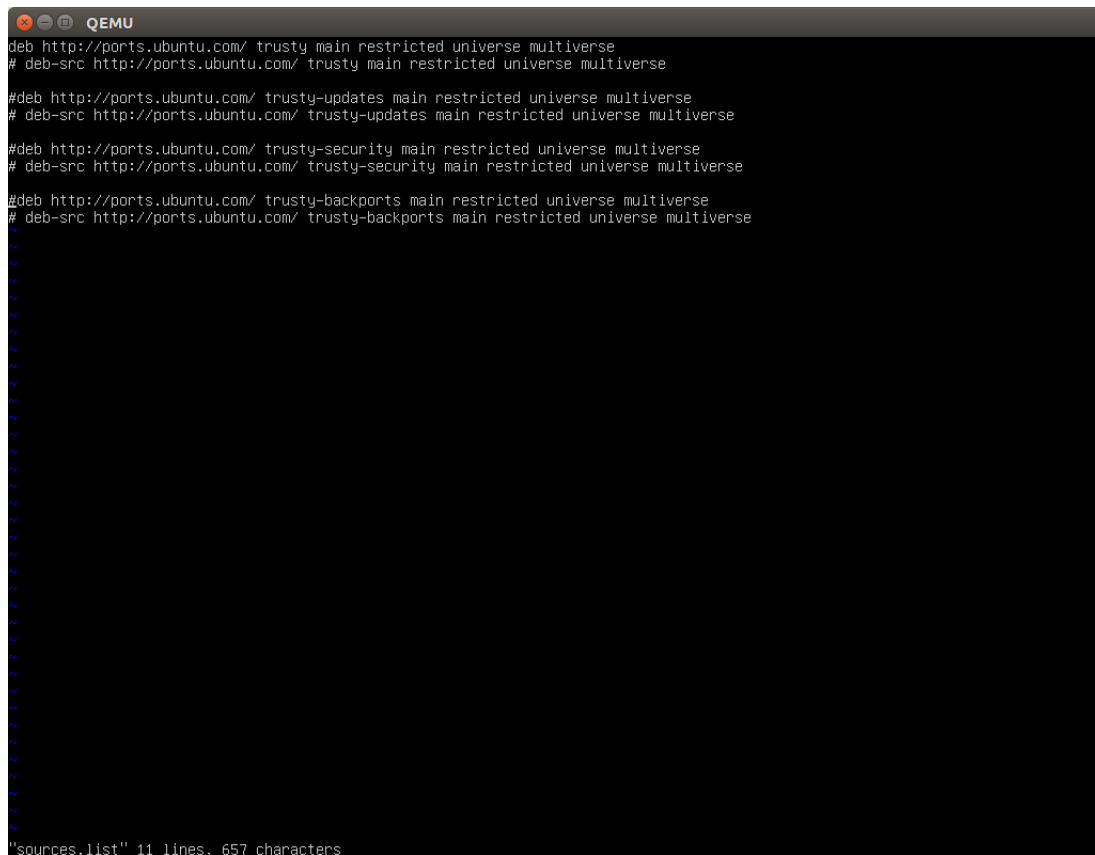


Figure 4: Disable Sources File

After that, you start installing basic utilities

```
$sudo apt-get install build-essential
$downgrade libc6 (sudo apt-get install libc6=2.19-0ubuntu6)
$sudo apt-get install libc6-dev
```

Python is needed also

```
$sudo apt-get install python
```

Git is needed to download source files. Type the following to let Git installed

```
$sudo apt-get install git
```

If you want a full desktop, go ahead and do so:

```
$ sudo apt-get install xubuntu-desktop # or
$ sudo apt-get install lubuntu-desktop # or
$ sudo apt-get install kubuntu-desktop # etc
```

Figure 5 shows xubuntu interface

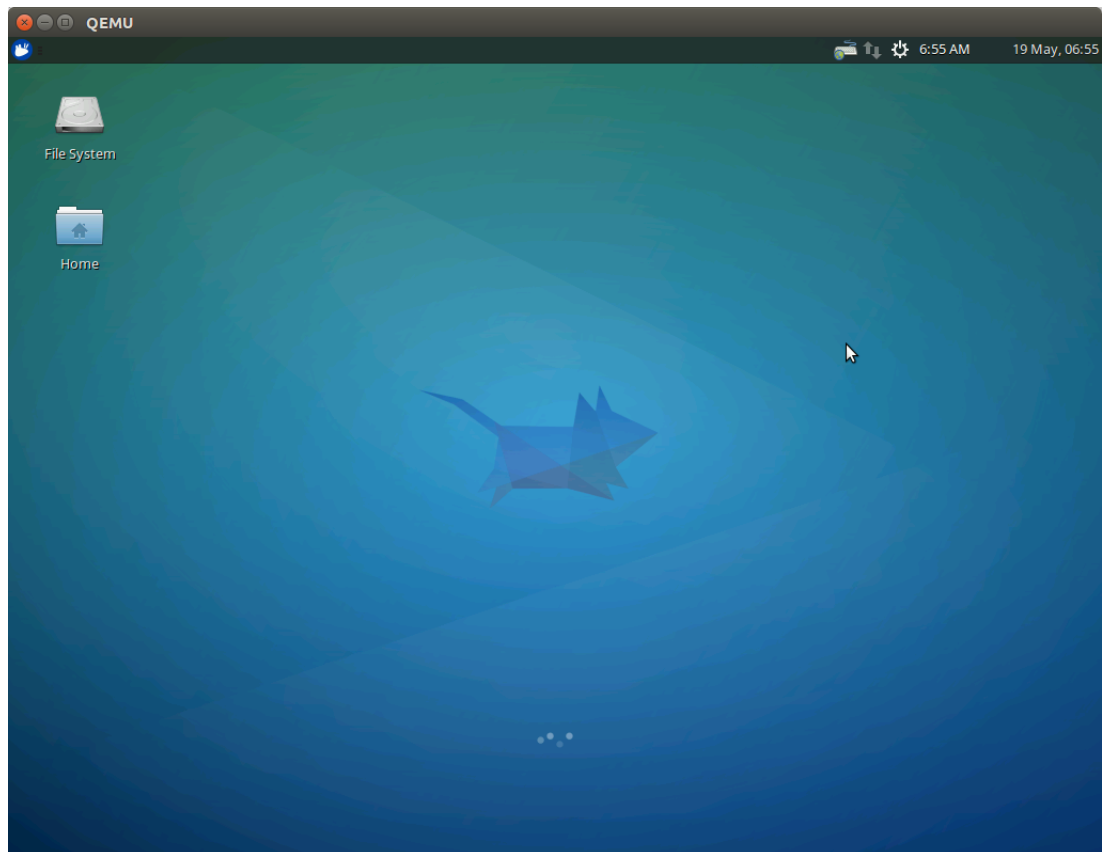


Figure 5: Xubuntu GUI

Bibliography

- [1] <https://wiki.ubuntu.com/ARM/RaspberryPi>
- [2] <http://www.finnie.org/>
- [3] http://gsoc.sitedethib.com/posts/apt-get_install_gcc-4.7-arm-linux-gnueabi/
- [4] <http://www.linuxfromscratch.org/>
- [5] <http://www.ibm.com/developerworks/library/l-busybox/>