# Howto setup Raspberry Pi Emulation with Qemu on Linux or Windows

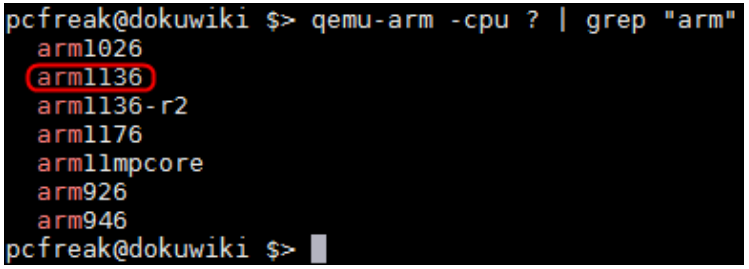
by

http://blog.pcfreak.de [http://blog.pcfreak.de] Der PCFreak - Hardware - Software - Repair

## Get Qemu

Download and/or install Qemu on your system (Linux or Windows). We need to emulate an ARM1176 CPU with Qemu, but some older versions of Qemu can not emulate this CPU and will not work. To check (on Linux) if your Qemu supports the ARM1176 or at lease the ARM1136-R2 execute the following command:

```
qemu-arm -cpu ?
```




The Qemu provided by your Linux distribution should just work (tested on latest Linux Mint - apt-get install qemu-system-arm)

For Windows, just use the latest Qemu which you can get here [http://lassauge.free.fr/qemu/] or here [http://www.omledom.com/].

It is possible, that you need this additional files in the Qemu folder when using Windows.

```
intl.dll
libglib-2.0-0.dll
libgthread-2.0-0.dll
libpng14-14.dll
libssp-0.dll
SDL.dll
zlib1.dll
```

This DLLs can be downloaded here [http://qemu.weilnetz.de/w32/dll/].

## Get a Raspberry image

Download your favourite Raspberry Pi image (linux based) and save it to your harddrive. This documentation used Raspbian

```
2013-07-26-wheezy-raspbian.img
```

based on Debian Wheezy downloaded from here [http://www.raspberrypi.org/downloads].

## Get a kernel

To be able to boot the Raspberry image, you need a kernel. You can download a working kernel from here [http://xecdesign.com/downloads/linux-qemu/kernel-qemu].

You should now have the file

```
kernel-qemu
```

on your harddrive.

## Verify prerequisites

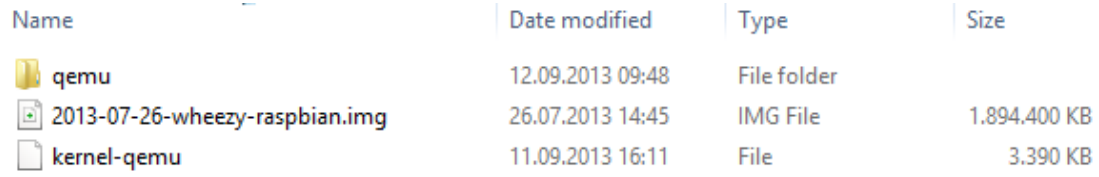
You should now have the following files on your system. Best practize would be to keep them in the same folder.

```
2013-07-26-wheezy-raspbian.img
kernel-qemu
```

On my Linux machine the folder looks like this:

```
pcfreak@dokuwiki $> ls -l
total 1897796
-rw-rw-r-- 1 pcfreak pcfreak 4087349248 Sep 11 16:49 2013-07-26-wheezy-raspbian.img
-rw-rw-r-- 1 pcfreak pcfreak    3470912 Dez  2  2012 kernel-qemu
pcfreak@dokuwiki $>
```

On my Windows machine I additionally have a subfolder where my Qemu lives:



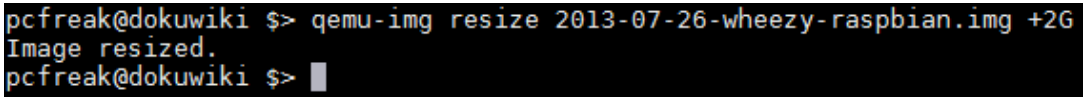
## Resize the image to have more space

A **real** Raspberry Pi uses an SD-Card and you have to copy the image to this card. We do an emulation of this, so we need to resize the image to the size we want to use later. To add 2Gb more space to the Raspbian image we have to execute the following Qemu command.

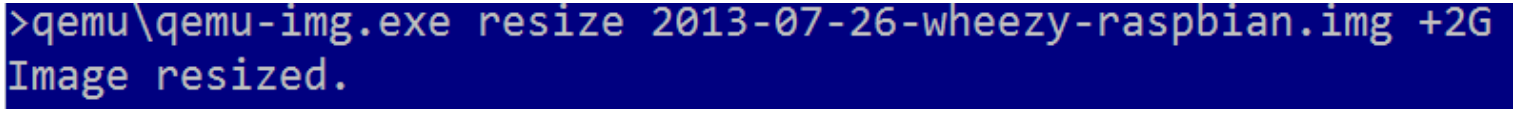
```
qemu-img resize 2013-07-26-wheezy-raspbian.img +2G
```

This should add additional 2Gb to 2013-07-26-wheezy-raspbian.img.

On Linux it looks like this:




And on Windows it should look similar:



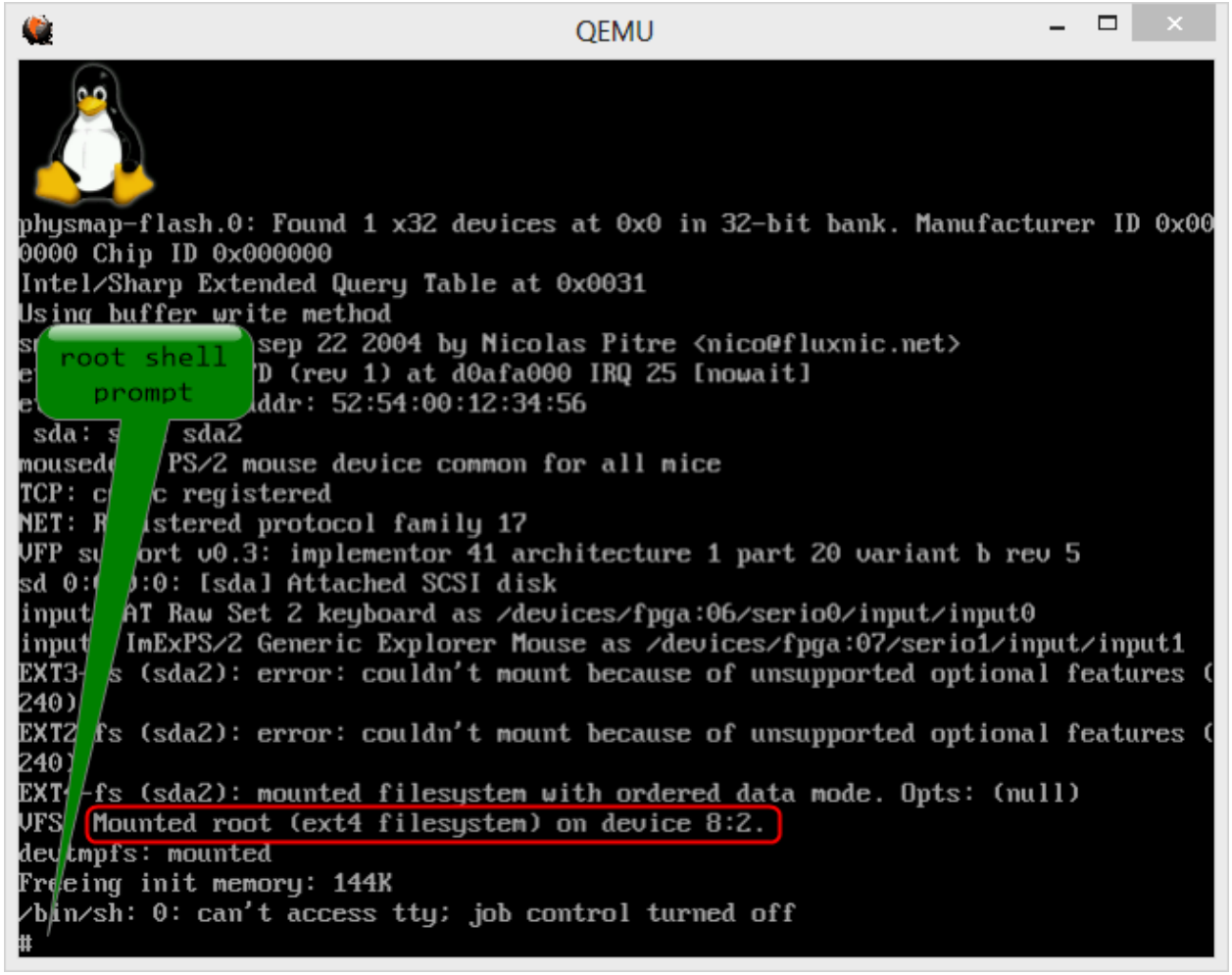

## Boot to root shell and fix some stuff

Before we can do a "real" boot with the Raspbian image, **we need to modify some files to make it work**. Some other documentations you can find on the net tell you to mount the image and then do the changes - which is easy under Linux - but when using Windows you cannot easily mount the image for modifications. An easier way is to boot Raspberrian into a root shell and do the changes there

Therefore fire-up Qemu with the following command (**Don't forget the** 'init=/bin/sh rw' **part**)

```
qemu-system-arm -kernel kernel-qemu -cpu arm1176 -m 256 -M versatilepb -append "root=/dev/sda2 panic=1 init=/bin/sh rw" -hda 2013-07-26-wheezy-raspbian.img
```

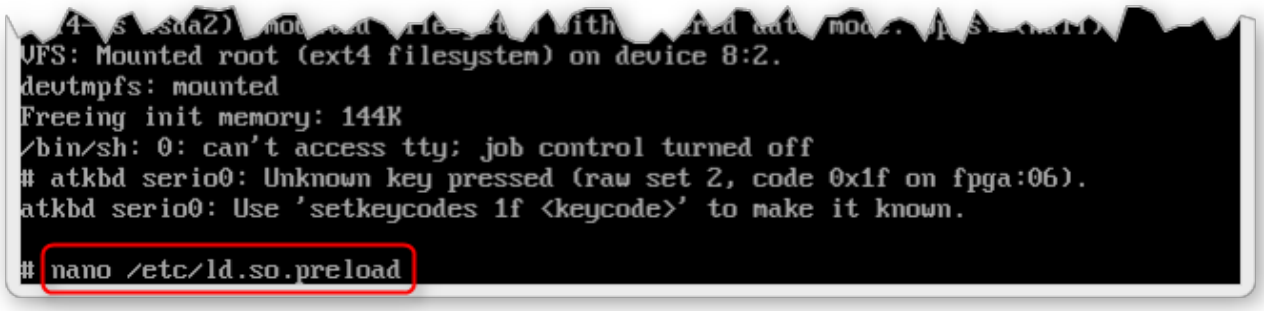
When the system has bootet into a /bin/sh root shell in read/write mode, it should look like this:




## Fix endless boot loop

At the root prompt open /etc/ld.so.preload with nano (**remember you have an english keyboard**)

```
nano /etc/ld.so.preload
```



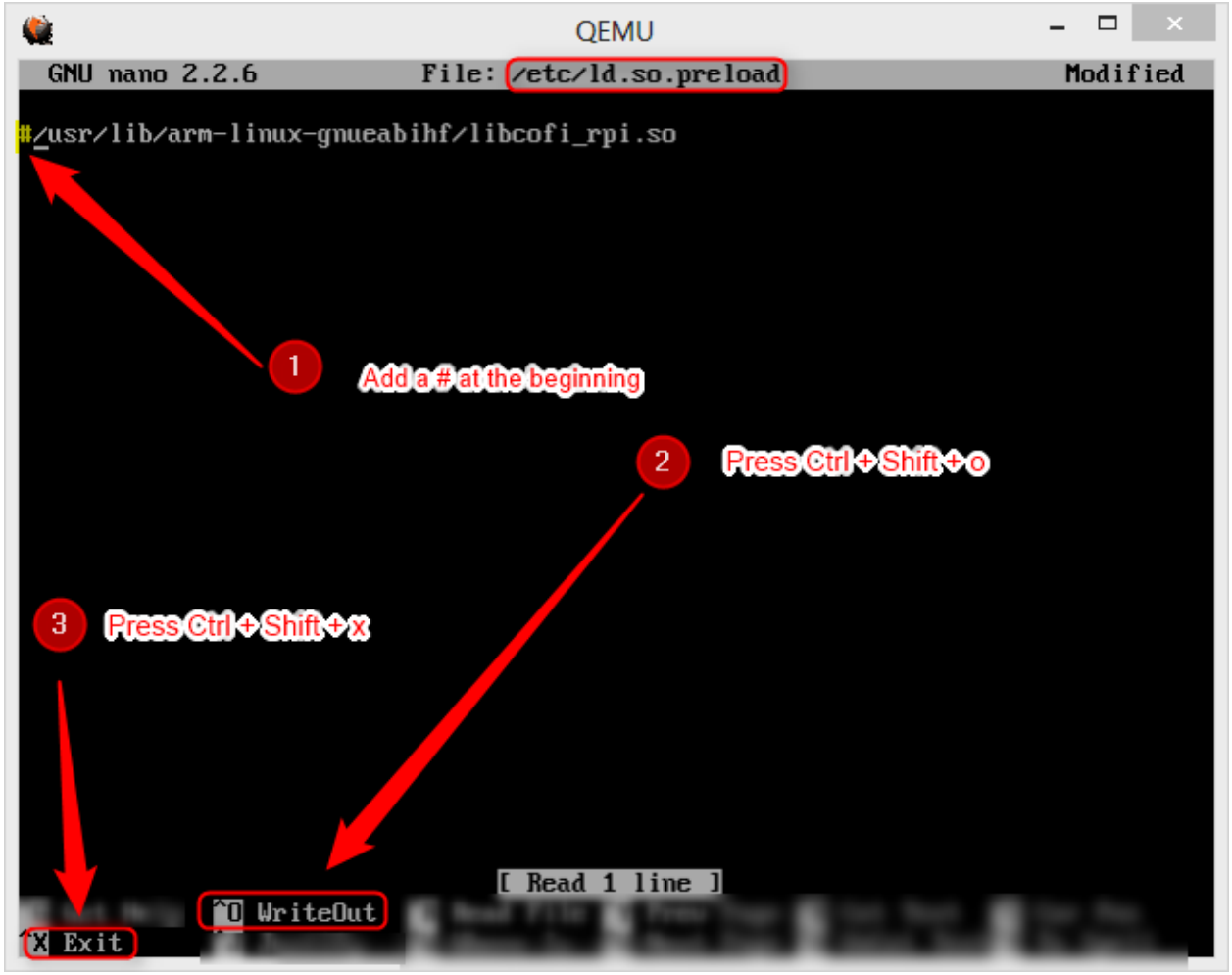

A file will open. It contains exactly one line.

```
/usr/lib/arm-linux-gnueabihf/libcofi_rpi.so
```

Add a # (SHIFT+3 on german keyboard) at the beginning of the line, so it looks now

```
#/usr/lib/arm-linux-gnueabihf/libcofi_rpi.so
```

Now press **Ctrl+O <ENTER>** to write the file and then **Ctrl+X** to exit the editor




This change will make sure, that libcofi_rpi.so will not be loaded. This change was necessary to avoid an endless boot loop.

## Fix disks for later use

Still at the root prompt create the file

```
/etc/udev/rules.d/90-qemu.rules
```

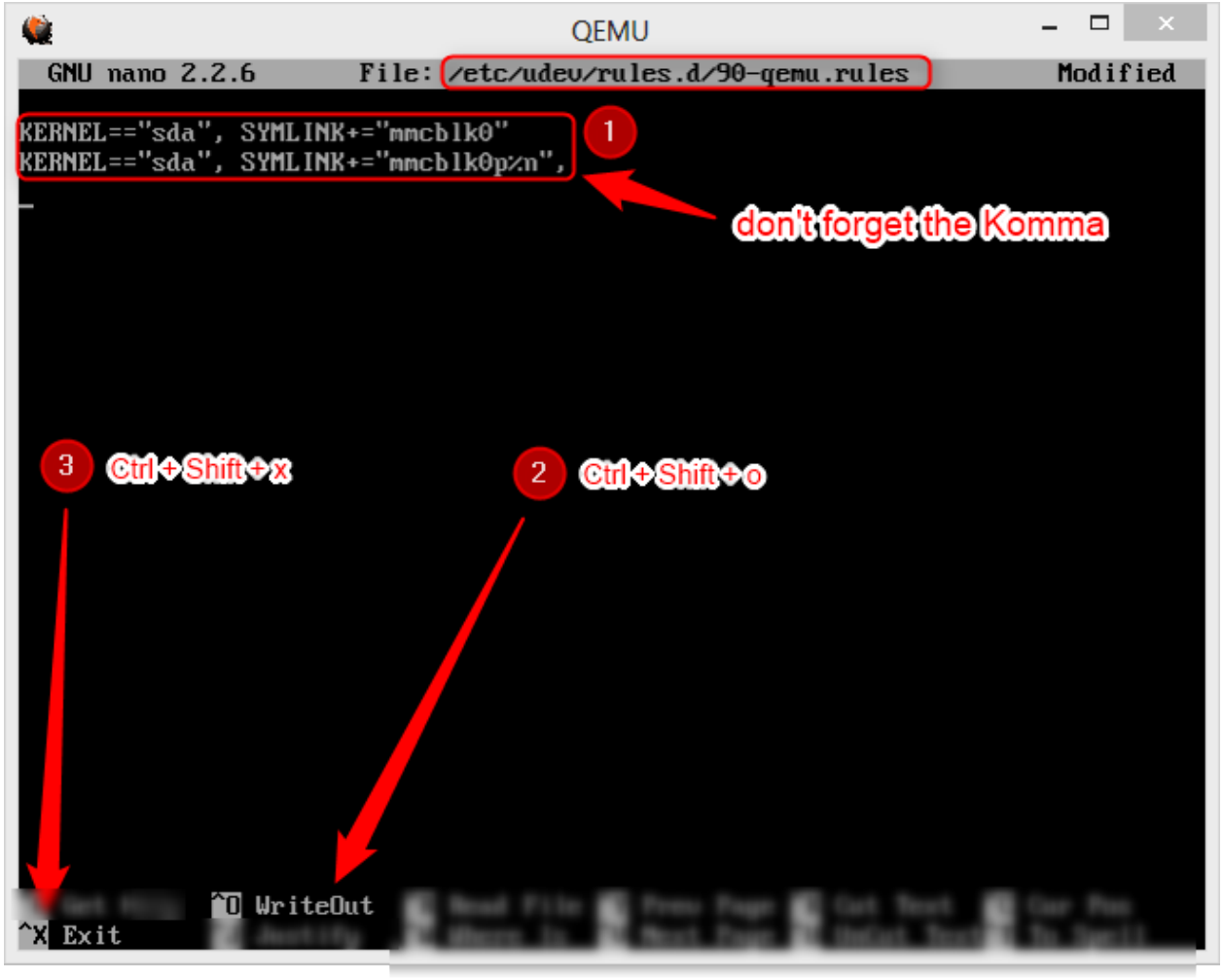
by executing the nano editor again:

```
nano /etc/udev/rules.d/90-qemu.rules
```


Enter the following lines into the file

```
KERNEL=="sda", SYMLINK+="mmcblk0"
KERNEL=="sda?", SYMLINK+="mmcblk0p%n",
```

Then press **Ctrl+O <ENTER>** to write the file and then **Ctrl+X** to exit the editor.




Finally enter the command

```
sync
```

to make sure everything is written to disk.


Now just close Qemu to stop the virtual machine.

Now we have applied all needed fixes to make the emulation boot without problems and being able to resize the root partition with raspi-config.

## First run

To fire-up your machine just execute the following command (pointing to the correct files).

```
qemu-system-arm -kernel kernel-qemu -cpu arm1176 -m 256 -M versatilepb -append "root=/dev/sda2 panic=1" -hda 2013-07-26-wheezy-raspbian.img
```

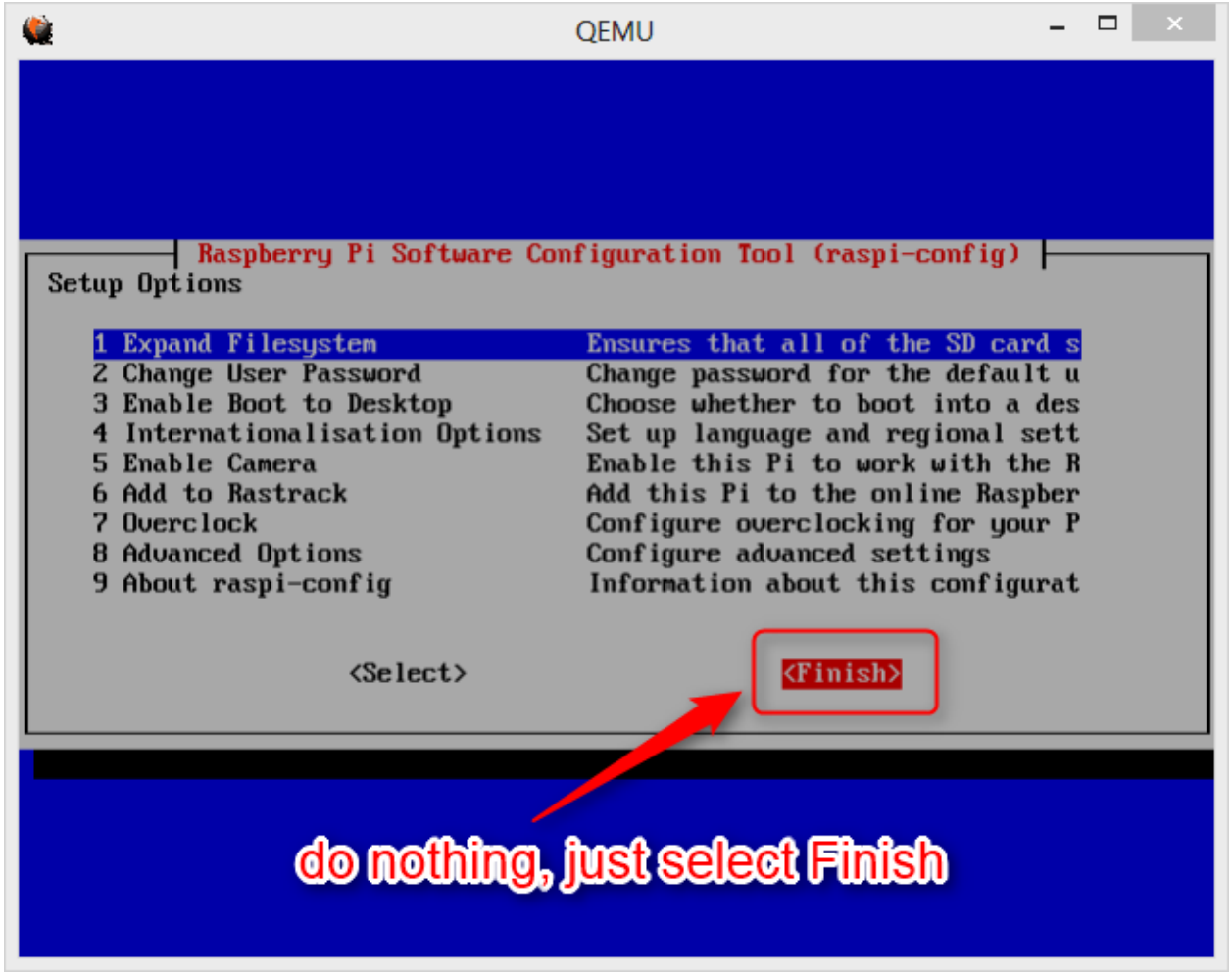
If Qemu automatically prompts you with a root shell asking for a **fsck** to fix the expanded file system execute the following

```
fsck /dev/sda2
```

Then reboot the emulation with

```
shutdown -r now
```

At the next boot Raspbian should automatically boot into raspi-config. **Immediately quit raspi-config** by selecting <FINISH>.




Then create a link with the following command

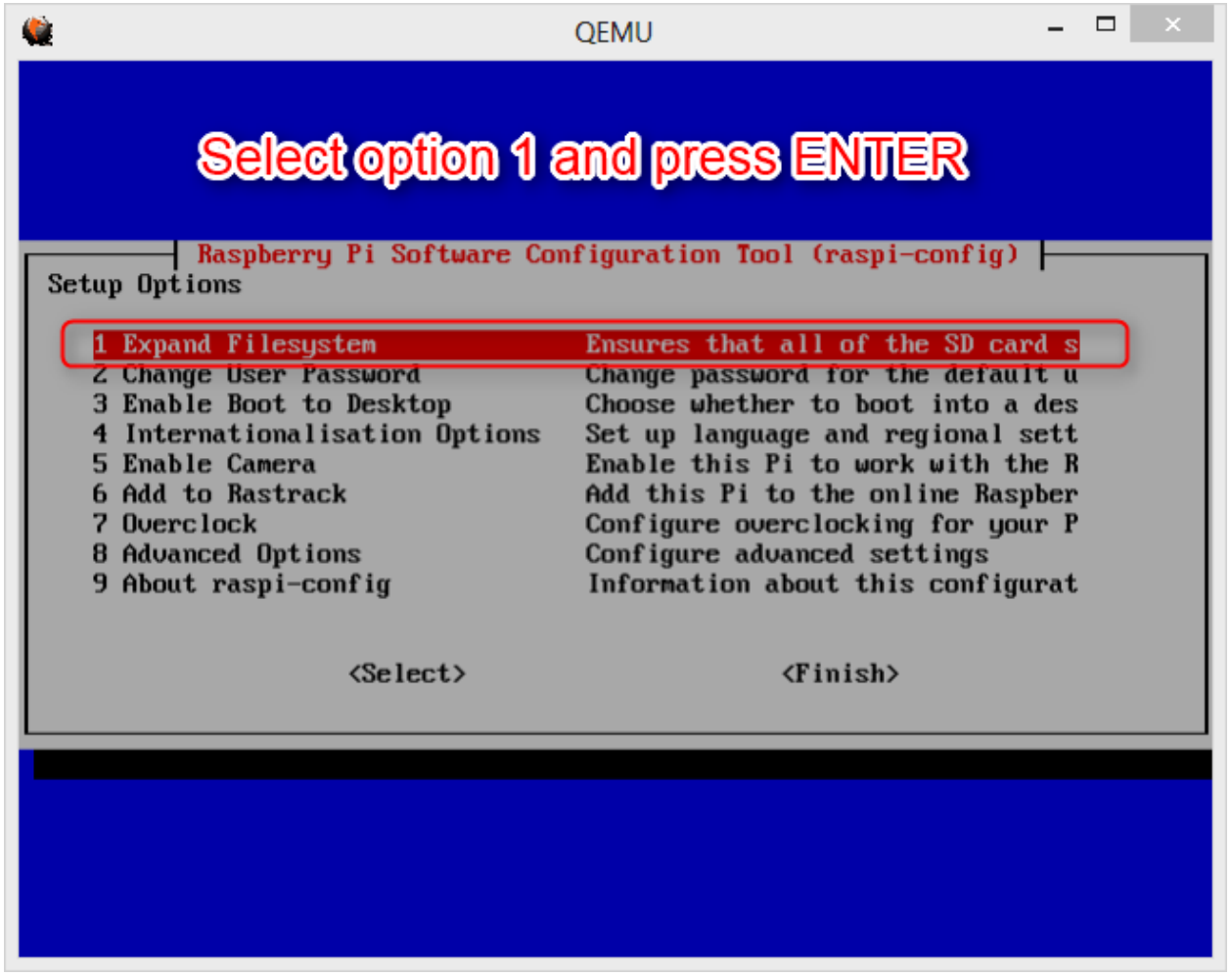
```
sudo ln -snf mmcblk0p2 /dev/root
```

Then manually restart raspi-config with

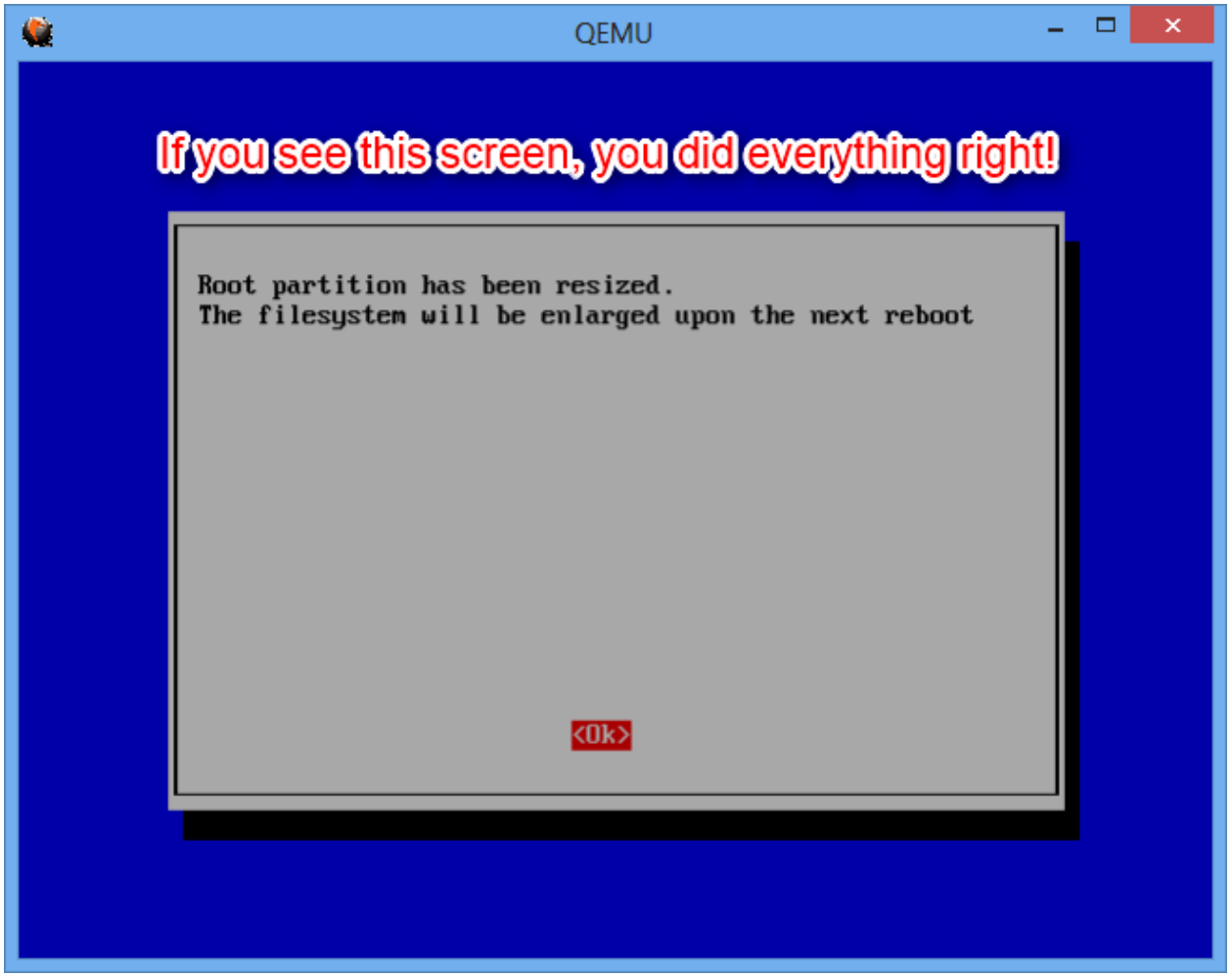
```
sudo raspi-config
```


and select

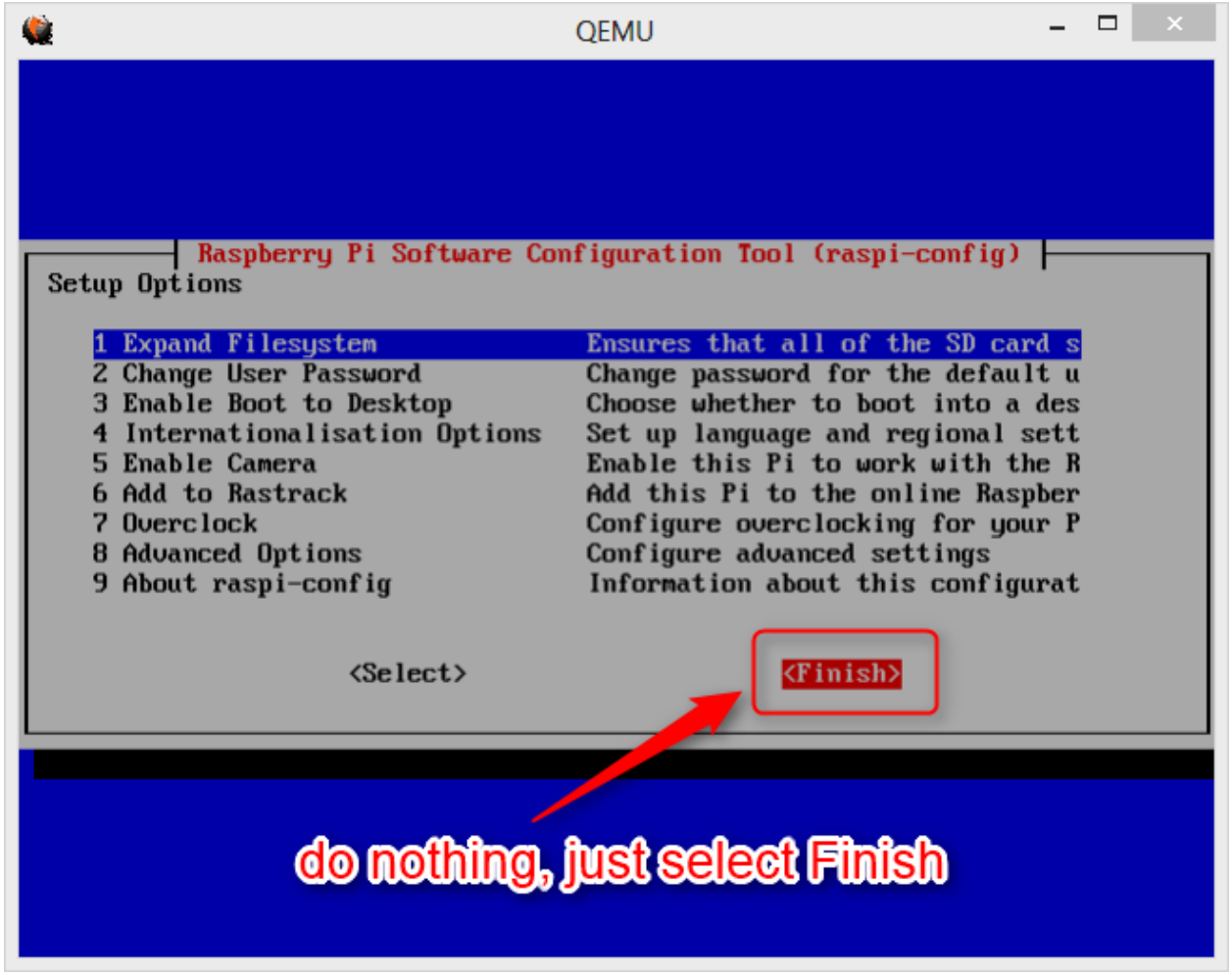
```
1 Expand Filesystem            Ensures that all of the SD card s...
```




The following screen should tell you that everything went fine.




Accept the message by pressing ENTER and exit raspi-config afterwards by selecting FINISH.



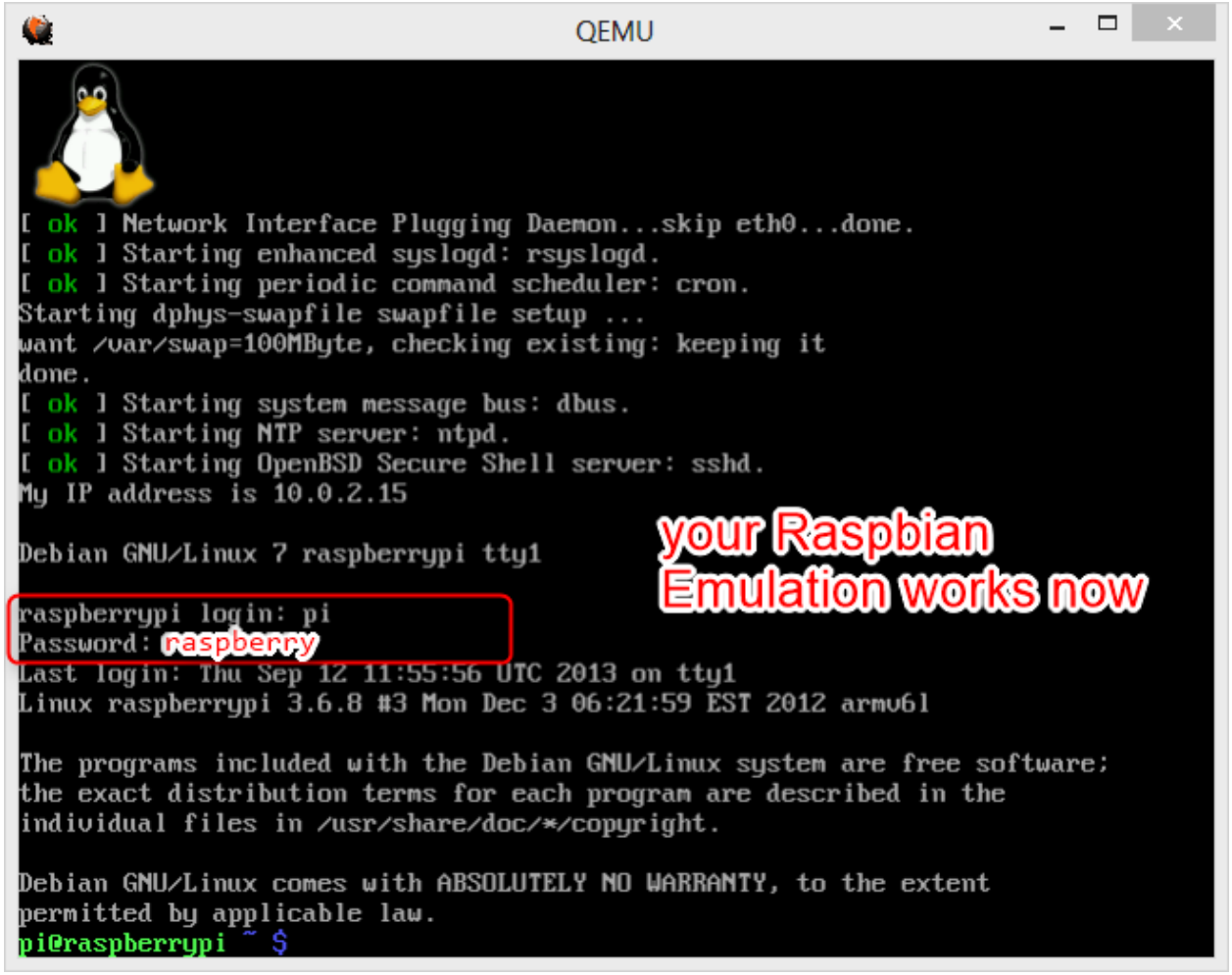

You will get asked if you like to reboot. Select YES to initiate the reboot.


During the reboot the disk well be resized and at the end you should arrive at the login prompt.

You can then login with the following credentials:

```
username: pi
password: raspberry
```

Please keep in mind, that the keyboard layout is english by default, make sure you type the **y** correctly.




## Fixes inside the emulation

### /etc/X11/xorg.conf

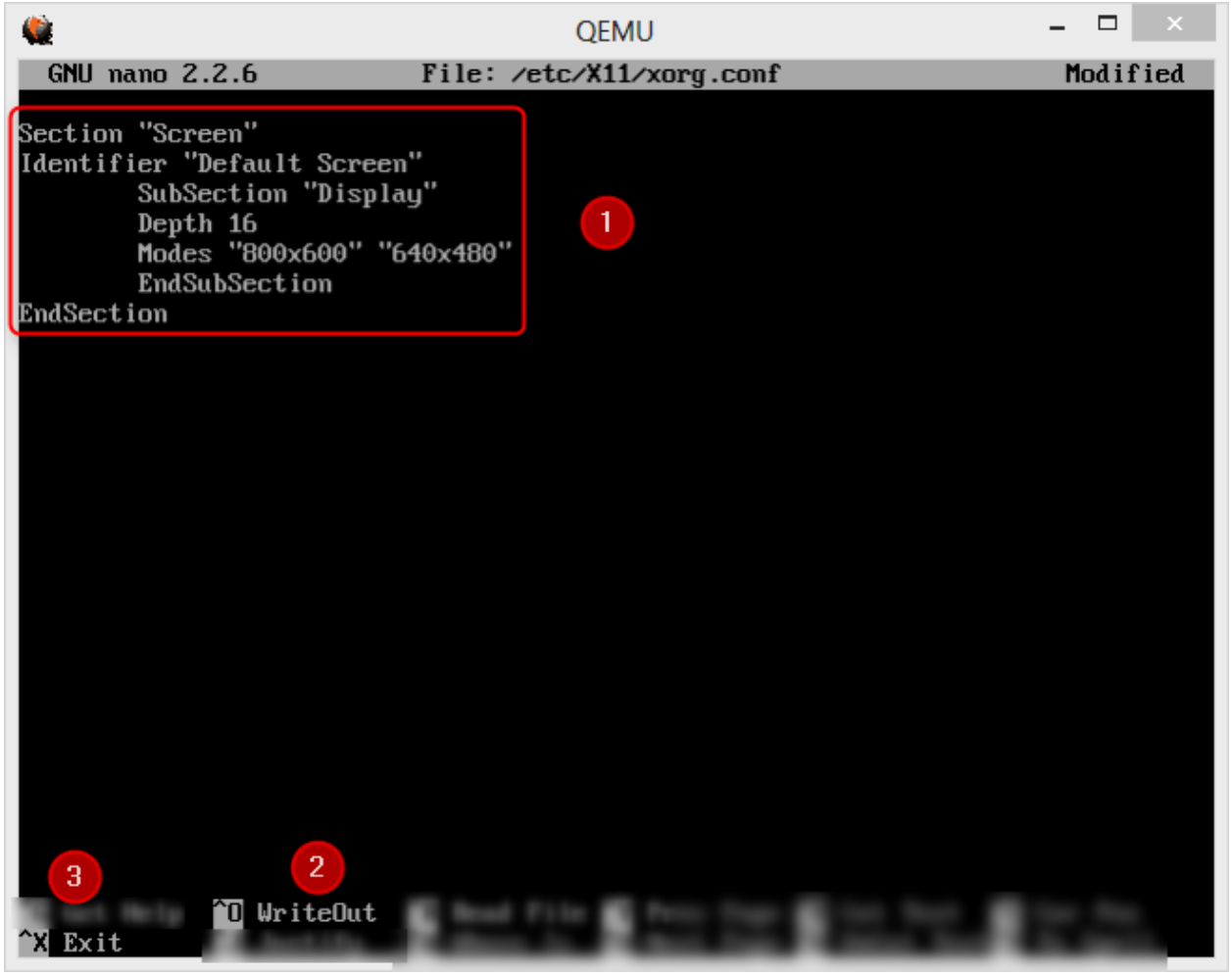
To be able to run X with 800×600 we need to manually create an xorg.conf by executing

```
sudo nano /etc/X11/xorg.conf
```


Write the following into the file

```
Section "Screen"
Identifier "Default Screen"
 SubSection "Display"
  Depth 16
  Modes "800x600" "640x480"
 EndSubSection
EndSection
```

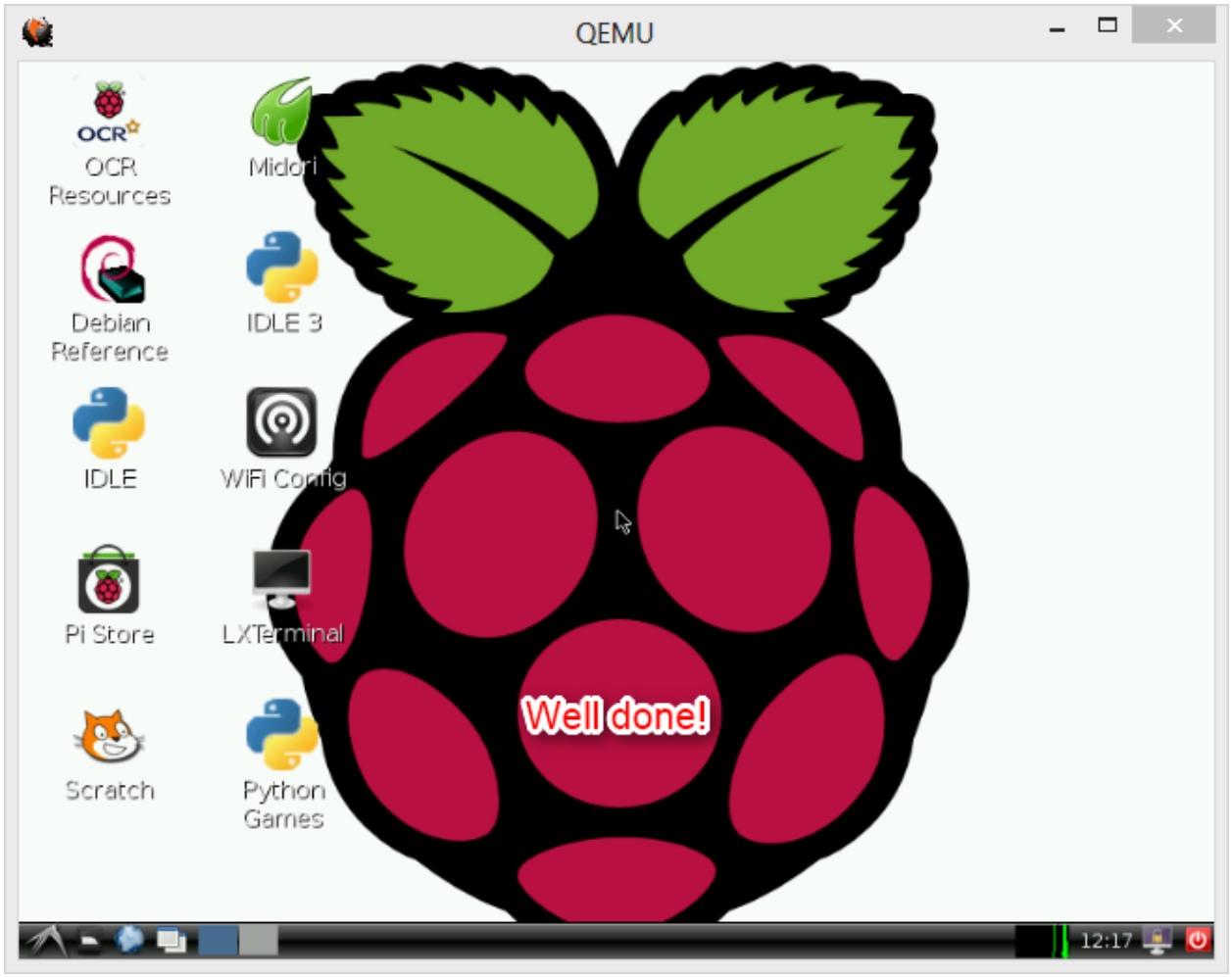
and then WriteOut (Ctrl+Shift+o) and Exit (Ctrl+Shift+X) in nano.




Just for testing you can now start the X server with

```
startx
```

It should present you the desktop with a resolution of 800×600 pixels as shown in this image.




## Additonal Notes

### Memory

Don' try to use more than 256MB memory within the qemu command line. The value is hard-coded in the emulated arm-chip, so you cannot change it!

### Graphics

At the moment it seems, that the maximum resolution you can emulate is 800×600 with a depth of 16bit.