

Installing RPCEmu 0.9.4 to Linux Mint

This guide, which I first used to get an earlier installation up and running, is based very closely on an original by the late Vic Norman of the Midlands User Group

This installation was carried out on a machine running 64-bit Linux Mint 20.1 but should apply to any Debian-sourced distribution. I chose to use the command line because all the correct commands may be copied and pasted from the RPCEmu web pages. However, be aware that the web pages use “<version>” instead of “0.9.4”, presumably so that the instructions would be applicable to newer versions. Make sure you use “0.9.4” (or whatever newer version appears in due course).

The RPCEmu home page, <http://www.marutan.net/rpcemu/index.php>, has links to all the files needed to complete this installation.

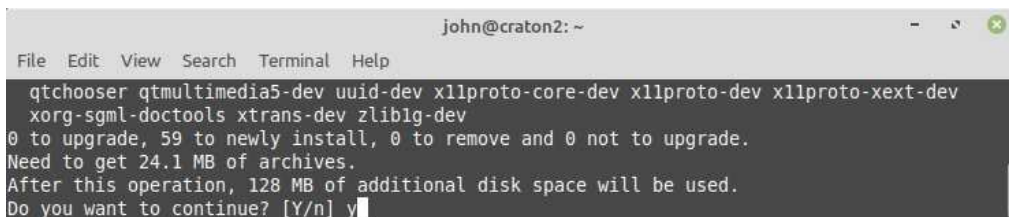
Install the Qt5 Development Package

From the homepage, follow the link, [How to compile from source](#) - Linux. Copy and paste the command:

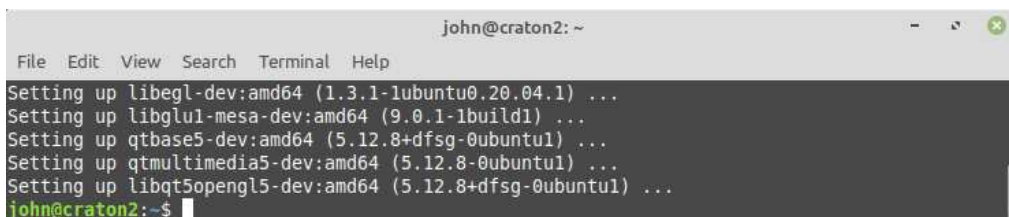
```
sudo apt-get install build-essential qtbase5-dev qtmultimedia5-dev libqt5multimedia5-plugins<enter>*
```

A terminal window titled 'john@craton2: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'sudo apt-get install build-essential qtbase5-dev qtmultimedia5-dev libqt5multimedia5-plugins' has been entered. The prompt '[sudo] password for john:' is shown with a masked password '*****'.

into whichever terminal program you use and enter your password. After a short while, it will tell you that so much disc space is needed and asks if you want to continue.

A terminal window titled 'john@craton2: ~' showing the output of the installation command. It lists several packages to be installed: qtchooser, qtmultimedia5-dev, uuid-dev, x11proto-core-dev, x11proto-dev, x11proto-xext-dev, xorg-sgml-doctools, xtrans-dev, zlib1g-dev. It states '0 to upgrade, 59 to newly install, 0 to remove and 0 not to upgrade.' and 'Need to get 24.1 MB of archives.' and 'After this operation, 128 MB of additional disk space will be used.' The prompt 'Do you want to continue? [Y/n] y' is shown with 'y' entered.

In the terminal, respond with **y<enter>** and, after a further short interval, the installation will be complete, returning you to the prompt.

A terminal window titled 'john@craton2: ~' showing the completion of the installation. It lists the packages being set up: libegl-dev:amd64 (1.3.1-1ubuntu0.20.04.1), libglu1-mesa-dev:amd64 (9.0.1-1build1), qtbase5-dev:amd64 (5.12.8+dfsg-0ubuntu1), qtmultimedia5-dev:amd64 (5.12.8-0ubuntu1), and libqt5opengl5-dev:amd64 (5.12.8+dfsg-0ubuntu1). The prompt 'john@craton2:~\$' is shown.

The Source Code

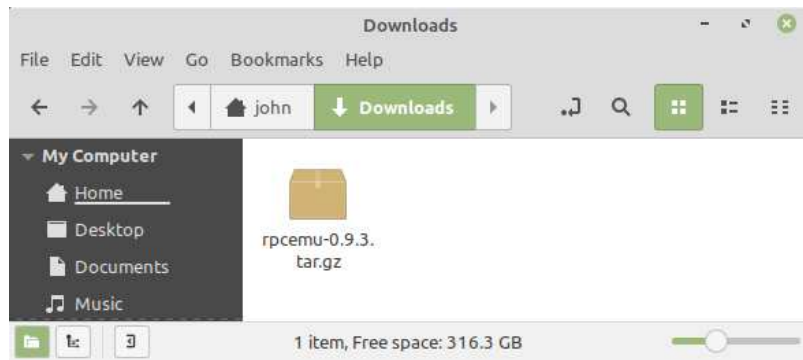
Download the source for RPCEmu from the Source Code link in the Downloads section of the home page linked to above. Tell the filer to save the file *rpcemu-0.9.4.tar.gz* rather than open it.

Before uncompressing the source code, set the directory where the file was saved as the current directory. It will usually be *Downloads*. Assuming you are already in your own (Home) directory, issue the command in the terminal:

```
cd Downloads<enter>
```

Note that the command is case-sensitive. The new current directory will show the following:

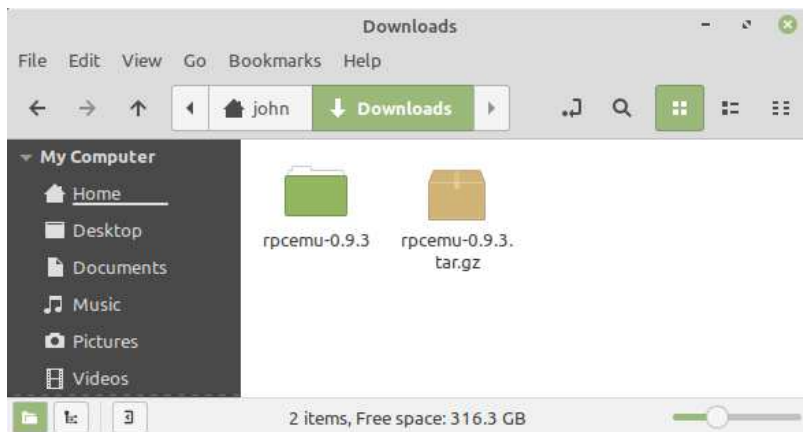
* This command is shown at reduced size in order to force it onto a single line. This avoids a newline if copying into the terminal.



Issue the following command to uncompress the source code file:

```
tar xfz rpcemu-0.9.4.tar.gz<enter>*
```

A directory called *rpcemu-0.9.4* will appear alongside the source code file as seen below.



It contains the following directories:

hostfs
netroms
poduleroms
riscos-progs
ROMs
src

and the following files:

cmos.ram
COPYING
readme.txt
rpc.cfg

Copy or move (drag) *rpcemu-0.9.4* to your name directory. Mine is */home/john* as shown below.

Ignore the fact that *Pictures* is shown as the last directory accessed in the image below; that's where the screen-shots had been saved to.

* See the notes at the end of this guide for an alternative way to extract the contents.



Yours will be */home/your-user-name*.

Compilation

To generate the dynamic recompiler involves the following steps before compiling.

In the desktop, navigate to *rpcemu-0.9.4/src/qt5/rpcemu.pro*, and load it into a text editor. Modify the file so that the line near the top of the file:

```
CONFIG += debug_and_release
```

becomes:

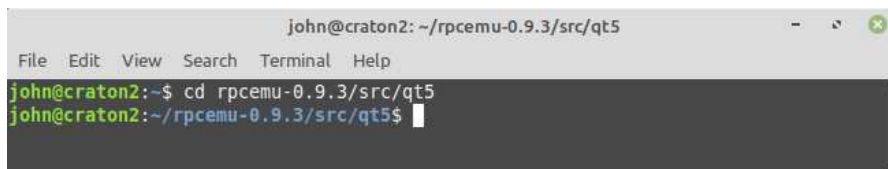
```
CONFIG += debug_and_release dynarec
```

Save the file.

Change the current directory back to your home directory with **cd ..<enter>** to go up one level from *Downloads*.

Change the current directory to be that containing the build scripts by copying the following into the terminal:

```
cd rpcemu-0.9.4/src/qt5<enter>
```

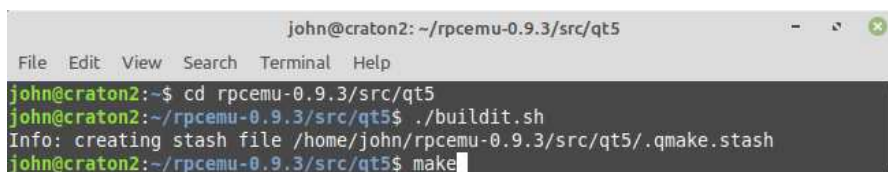


To prepare the Makefile, copy into the terminal the following:

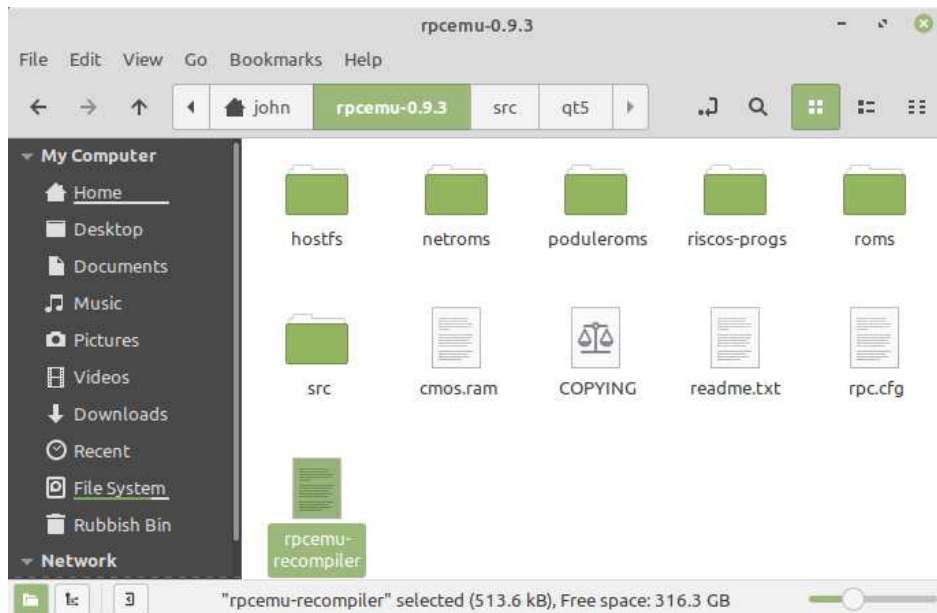
```
./buildit.sh<enter>
```

Follow that with the command:

```
make<enter>
```



After a short while, a file called *rpcemu-recompiler* should appear alongside all the other directories and files in *rpcemu-0.9.4* as shown below.



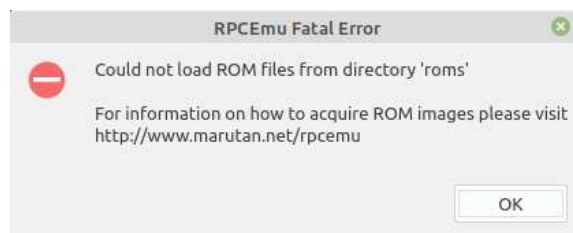
Issue the command:

```
cd ../../<enter>
```

to go back to your home directory from where RPCEmu can be run with the command:

```
./rpcemu-recompiler<enter>*
```

but, as RISC OS hasn't been installed, it will complain, both in the RPCEmu window and the terminal as shown below.



Installing RISC OS

Return to the RPCEmu homepage at <http://www.marutan.net/rpcemu/index.php>, and follow the link to [Using ROOL RISC OS 5](#) - All platforms. Here there are two links to follow.

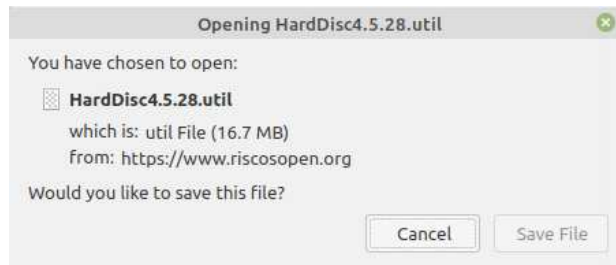
Firstly, the one for the [RISC OS ROM Image File](#). Currently marked as being for RISC OS 5.24, the link takes you to the page for the current stable version, 5.28. Download *IOMD 5.28 stable (softload)*. In the desktop, open the *Downloads* directory and double-click on the ZIP file and navigate to *soft/!Boot/Choices/Boot/PreDesk/!!SoftLoad*. When *!!SoftLoad* is reached, it will contain a file *riscos* which is the ROM.

Drag this out of the ZIP file to a convenient temporary location; *Downloads* is as good as anywhere. The download *IOMD 5.28 stable (softload)* is no longer needed but could be kept as insurance in case of finger trouble later in the process.

Move the ROM image, *riscos*, into the directory *ROMs* in *rpcemu-0.9.4*.

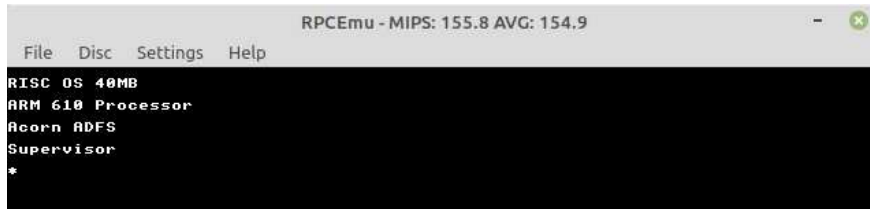
The second link, [self extracting hard disc](#) file, takes you to the Miscellaneous downloads page on the ROOL website where, towards the bottom of the page, you can download the file *HardDisc4 (self-extracting)*. When downloading it, Linux Mint asks you whether or not to save it rather than asking about opening/saving it.

* See the notes at the end of this guide for alternative ways of starting RPCEmu.



Save it and then move it from *Downloads* into the *hostfs* directory within *rpcemu-0.9.4*.

Run the file *rpcemu-recompiler* again and the top of the resulting window should look like this:



At the * prompt type the following:

Desktop<enter>

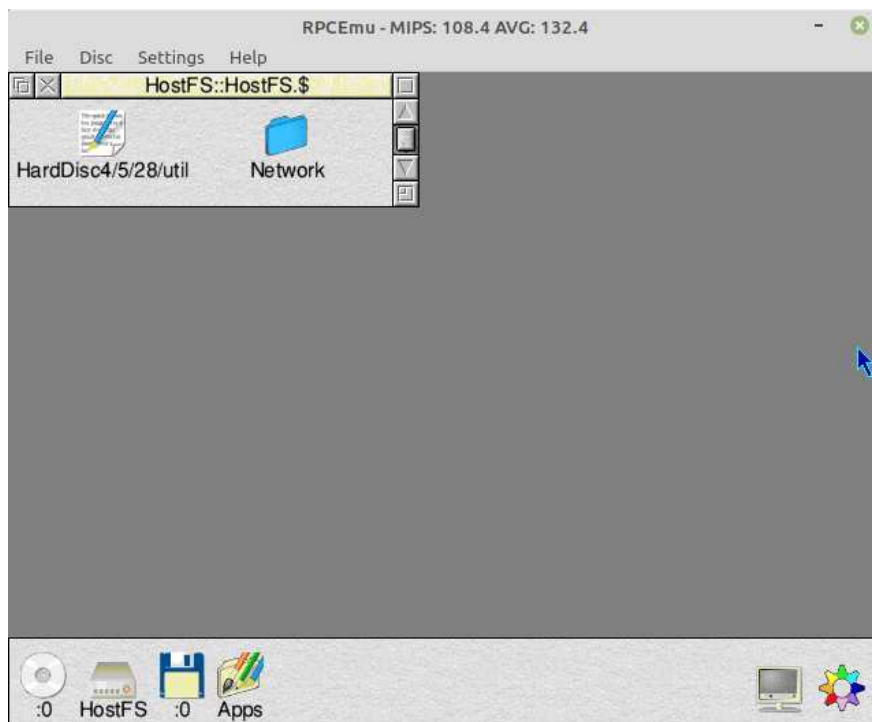
and the screen should look like this:



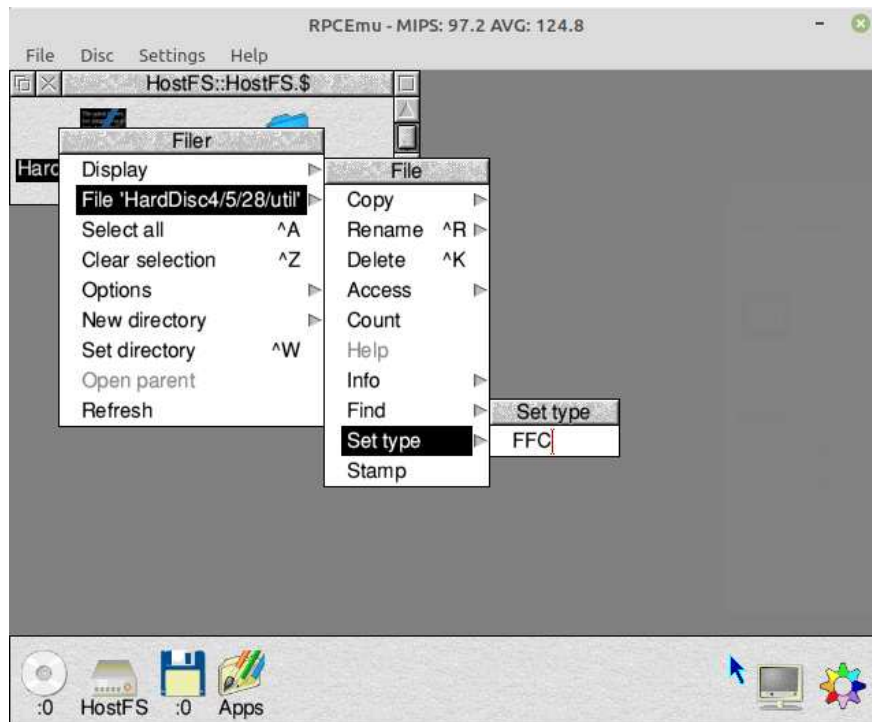
Click on **Cancel** in the middle of the error box and the screen should look like this:




Click on HostFS on the icon bar and the screen should look like this:

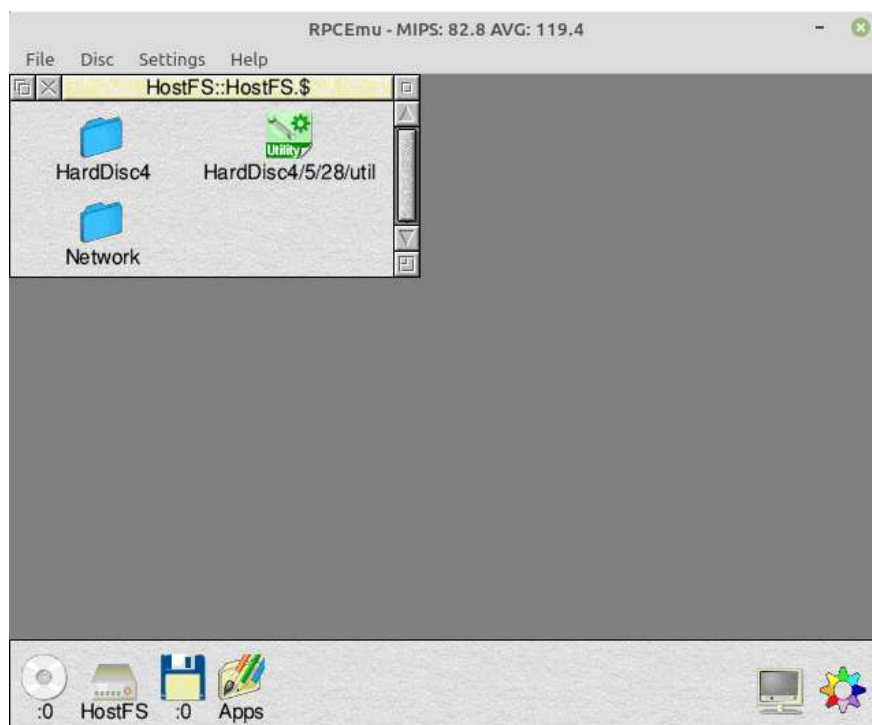


The next step is to change the file type of *HardDisc4/5/22/util* from FFF (Text) to FFC (Utility) by clicking middle button of the mouse over the *HardDisc4/5/22/util* icon to open the filer menu and moving down the menu to set type and changing Text to Utility or FFC as shown below:

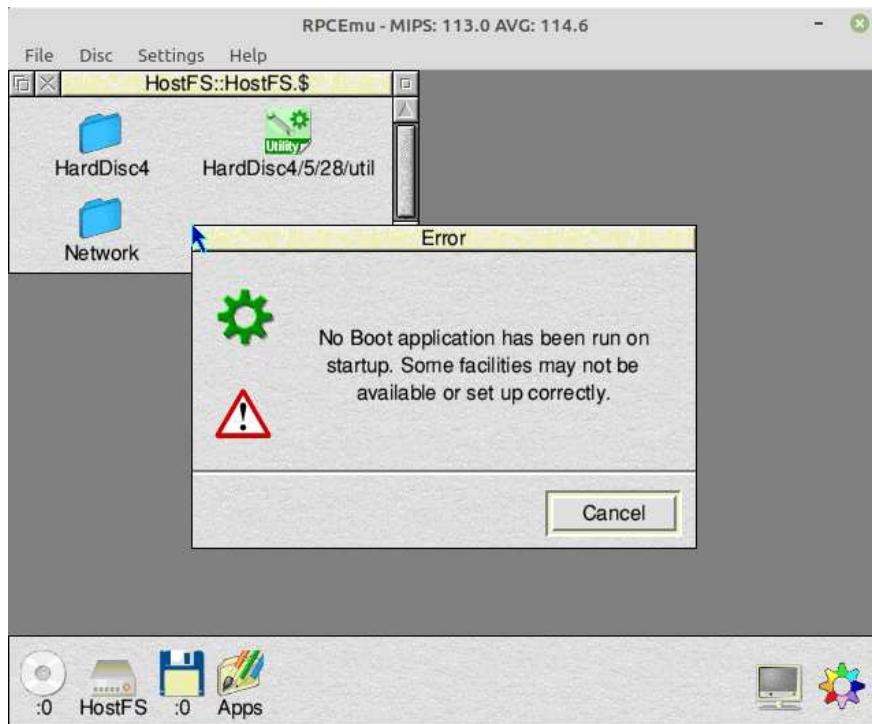


The file icon should change to a green utility icon like this: 

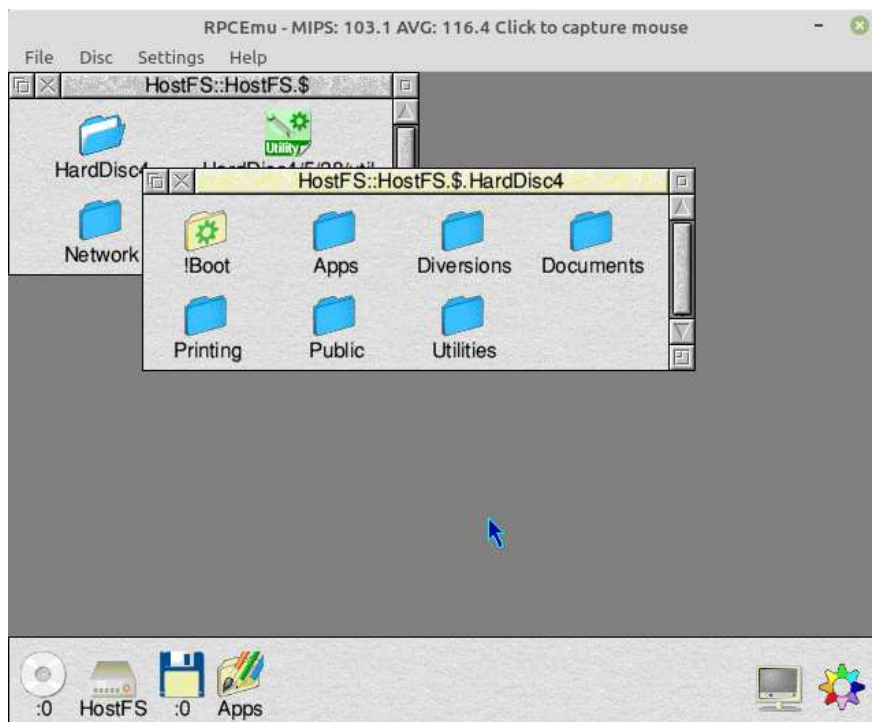
Double click on the *HardDisc4/5/22-util* icon and the machine will extract the HardDisc4 image and the screen should look like this:



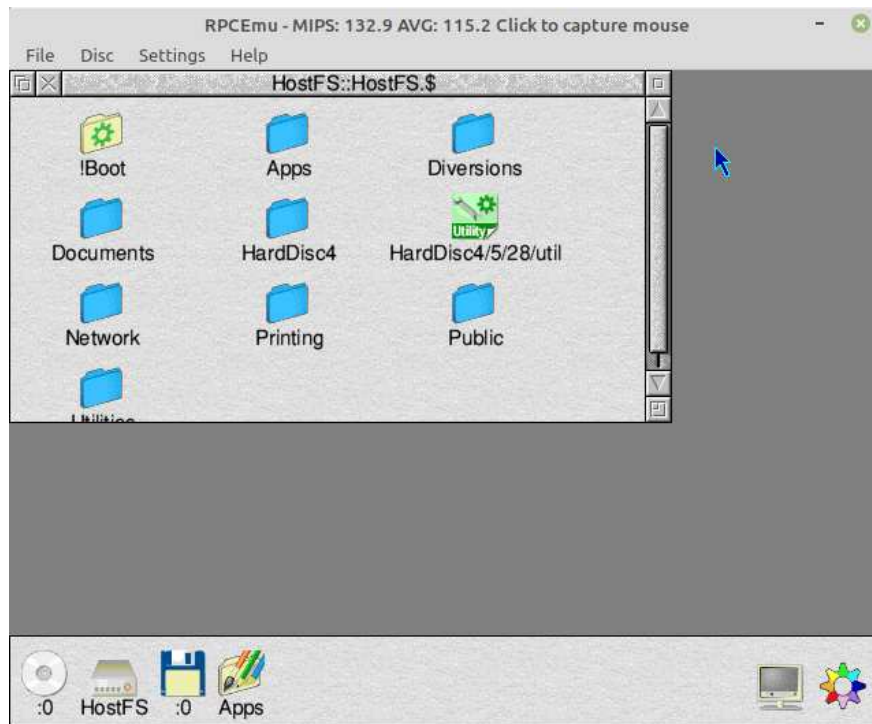
Double click on the *HardDisc4* icon and you will get an error message as shown below:



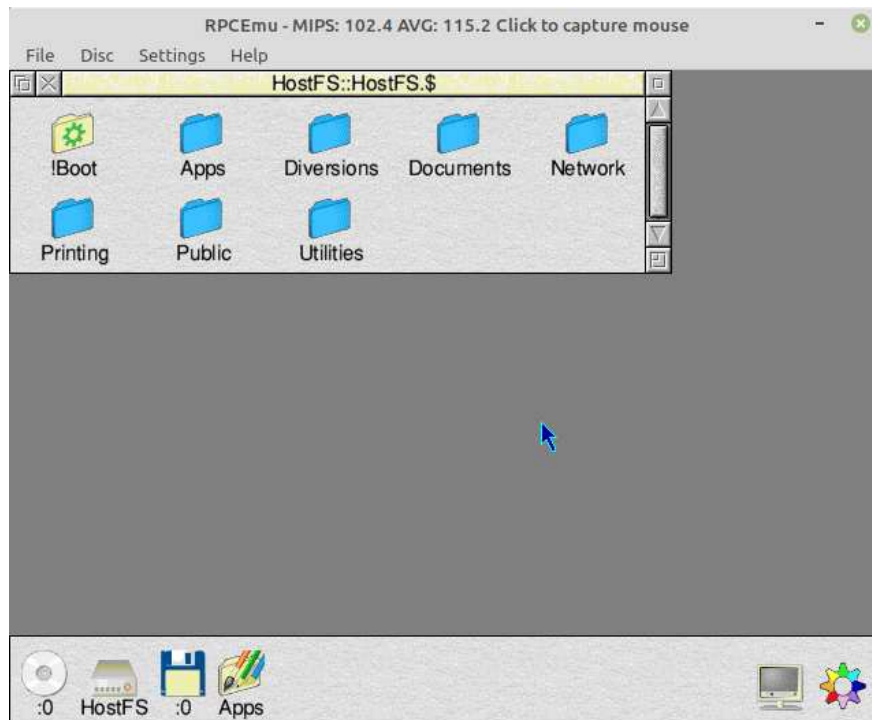
Press Cancel and the screen should show the following:



Select all the files in the *HardDisc4* directory and, with the shift key held down, move them up a level into the *HostFS* directory and the screen should look like this:



Now delete the directory *HardDisc4* and the file *HardDisc4/5/22-util* and the screen should look like this:



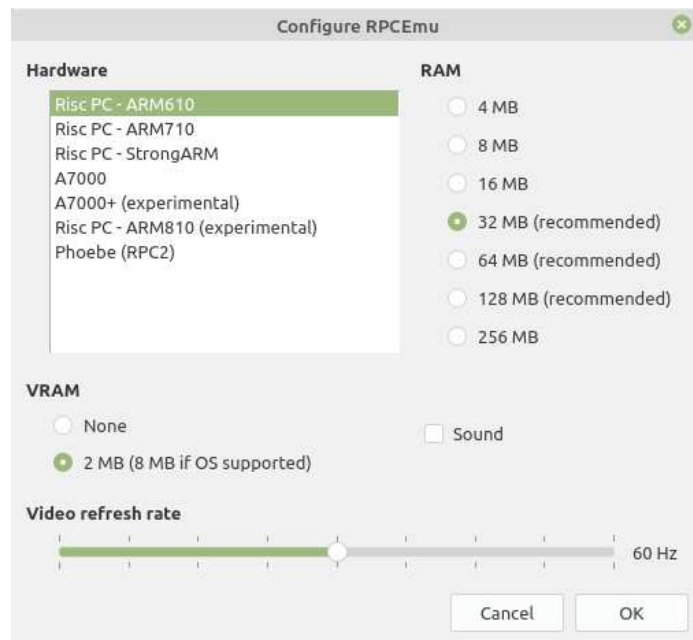
Press F12 and a command line will appear at the bottom of the screen with a * prompt. Type the following commands to make the machine boot from the HostFS drive:

```
configure filesystem hostfs<enter>
```

```
configure boot<enter>
```

Press enter again to exit the command prompt.

Now click on Settings in the RPCEmu menu bar and select Configure... The following configuration options will appear:



The image above used to show a configuration window with Risc PC - StrongARM already selected. I've replaced it with a proper default window.

Click on Risc PC - StrongARM and change the RAM to 256 MB and then click on OK. A request to reset RPCEmu will appear and accepting it will change the configuration. The only apparent change will be an increase in the size of the RPCEmu window.

The range of MDFs is limiting - there is no 1920×1080, for example and a suitable MDF will have to be sourced from elsewhere, eg, a RaspberryPi installation.

In the Settings menu, select Networking... and, unless you are very brave (or know what you are doing), check Network Address Translation (NAT) and click OK. At this point you will have a network connection to the internet even though there isn't a browser installed to show it.

You can now set up the system in accordance with your preferences. Applications and data can be transferred to the **parent** HostFS in Linux by copying them from Virtual RiscPC directly using some form of external drive (Linux understands FAT32 and NTFS) or via Dropbox or pCloud. From a native RISC OS machine, they should be zipped to preserve file types and the zip(s) transferred to the Linux file system. Only unzip them from within RPCEmu otherwise the filetypes will be lost.

John McCartney
Dec 2021

Extracting *rpcemu-0.9.4*

This is trivially easy, certainly on Mint 20.2 and, I'm pretty sure it's the same on most Linux distributions. Can you guess what it is yet?

Just double-click on *rpcemu-0.9.4.tar.gz* and, lo and behold, a new filer window opens showing the sole contents as *rpcemu-0.9.4*. Drag it out to the *Downloads* directory. It couldn't be easier.

This obviates the need to change the current directory to *Downloads* so there's no need to use `cd ..<enter>` to go up a level to the home directory.

Instead go straight to the command for changing to the directory containing the build scripts, namely:

```
cd rpcemu-0.9.4/src/qt5<enter>
```

Running RPCEmu

For those who grew up with the command line, running RPCEmu with a quick fiddle in a terminal might seem the proper way to launch it. We humans, on the other hand, might find it more convenient to double-click on something.

My something is a script file containing the following lines:

```
#!/bin/bash
cd /home/john/rpcemu-0.9.4
./rpcemu-recompiler
```

I have it saved as *Start RISC OS* inside the *rpcemu-0.9.4* directory. It's Properties are set with the Execute option set in Permissions.

In order to run it without tunnelling down to where it's saved, I created a link which sits on the desktop. Whatever else I'm engaged in doing, the desktop (and anything saved on it) is accessible with <logo key>D which toggles the view of the desktop. However, see over the page for using Linux's workspaces

By default, the filer in Mint doesn't show the option to make a link until it has been set in the filer preferences. Also in the filer preferences, permission to execute without asking needs to be set. This saves you being asked for permission every time you try to run RPCEmu. No doubt similar arrangements need to be made in other distributions/desktops.

The script appears in my default text editor with the colouring (shown above) applied automatically.

Using Linux's Workspaces

I discovered that there might be a problem when using an MDF where the screen resolution (in particular, the height) is the same as the natural resolution of the display being used; in my case, 1600×900. When entering full-screen mode, the RPCEmu window would stay in its default position but lose its title bar resulting in the icon bar being invisible, somewhere off the bottom of the screen. As with a properly behaved full-screen display, the window is immovable, just in the wrong position.

I discovered that, without invoking the full-screen mode, the Linux feature of <alt>click-and-hold (anywhere in the window) allows the window to be dragged into such a position so that it *appears* to be in full-screen mode. This necessitates setting the panel (the Linux equivalent of the icon bar) to autohide itself. This can be done by right-clicking in the panel and selecting the preferences. The drawback here is that any accidental movement of the mouse to the bottom of the screen causes the panel to pop up and obscure the icon bar.

Discussion on the RPCEmu mailing list* has led to a couple of very useful coping strategies.

Firstly, selecting full screen mode *before* RPCEmu reaches the desktop allows it to enter full-screen mode correctly. Thank you to Frank de Bruijn for this. Frank has pointed out that leaving full-screen mode after successfully invoking it prevents it being invoked again without the previously described problem.

Secondly, both Frank and David Gee suggested using Linux's ability to use different workspaces to run RPCEmu in full-screen mode. To change the workspace use <ctrl><alt>right arrow or <ctrl><alt>left arrow to move through the four available workspaces. These key shortcuts apply to Linux Mint and may not hold for other distributions however there will be appropriate key shortcuts for whichever distribution you are using.

Combining both of these strategies works really well. Selecting, say, workspace 2[†] before running RPCEmu and selecting full-screen mode before it gets to the desktop sets up RPCEmu in full-screen mode in workspace 2. It doesn't need to come out of full-screen to access the Linux desktop because workspace 1 can be viewed in its entirety by using <ctrl><alt>left arrow. From my point of view, this is a better arrangement than <alt><enter> as used by VirtualRPC.

This is particularly convenient when transferring files into RPCEmu. Plugging in an external drive (or using Dropbox or pCloud) and copying files into the relevant directory within HostFS from workspace 1 makes them immediately available to RPCEmu in workspace 2—though a directory may have to be closed and reopened in order that RISC OS can 'see' them.

* The list can be joined by going to: <http://www.riscos.info/cgi-bin/mailman/listinfo/rpcemu>

† Of course, RPCEmu could be opened within workspace 1 and Linux accessed from workspace 2 (or 3 or 4) but, as RPCEmu is a 'guest' on the system, it would seem only polite to make it run in a non-default workspace.