***********************Draft***********************

QEMU Ultibo Bare Metal
FileHandling
07/22/21

***********************Draft***********************

Note:
https://ultibo.org/forum/viewtopic.php?f=13&t=1303&p=11632#p11632

By Ultibo Wed Jul 21, 2021 9:01 pm

I suspect the version of QEMU that you have on the RPI3B+ is later than the one on the RPi4, try doing qemu-system-arm -version on each one.

We recently discovered that the Ultibo SD card driver was not compatible with the latest versions of QEMU, a fix for this is included in the release from today (Ultibo core 2.1.079) so if you update your RTL to the latest either using the RTL Builder or by rerunning the ultiboinstaller script then it should work now.
https://en.m.wikipedia.org/wiki/QEMU. On the pi400-1 I ran ./ultiboinstaller.sh on pi400-1.

https://en.m.wikipedia.org/wiki/QEMU

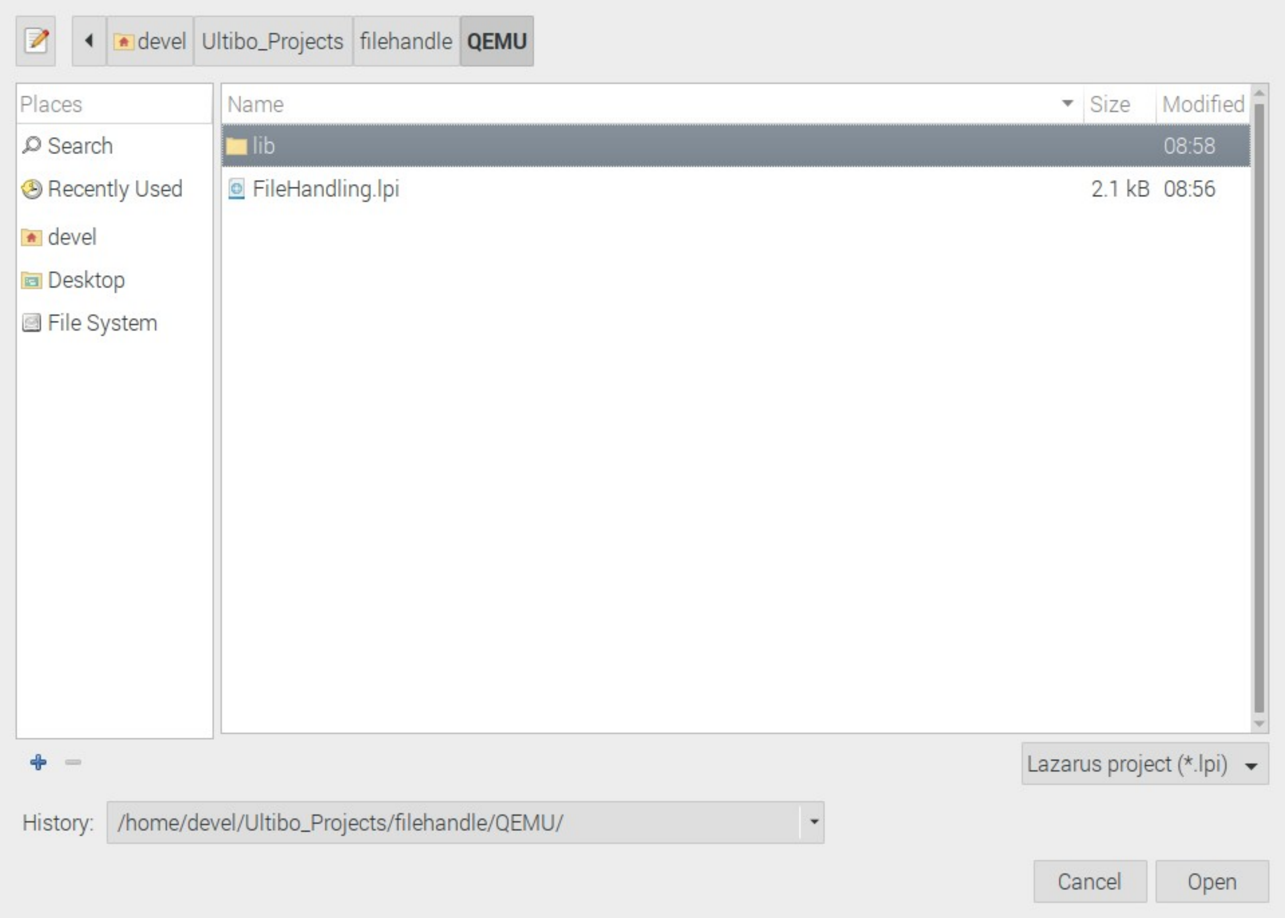
QEMU is a hosted virtual machine monitor: it emulates the machine's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems. It also can be used with Kernel-based Virtual Machine (KVM) to run virtual machines at near-native speed (by taking advantage of hardware extensions such as Intel VT-x). QEMU can also do emulation for user-level processes, allowing applications compiled for one architecture to run on another.[3]

Note : Additional software is needed to run QEMU “sudo apt-get install qemu-system-arm”. The following programs are added.

/usr/bin/qemu-img /usr/bin/qemu-nbd /usr/bin/qemu-system-aarch64
/usr/bin/qemu-io /usr/bin/qemu-pr-helper /usr/bin/qemu-system-arm

The command line for starting **Lazarus IDE (Ultibo Edition) “~/ultibo/core/lazarus.sh”**

Project/Project Open

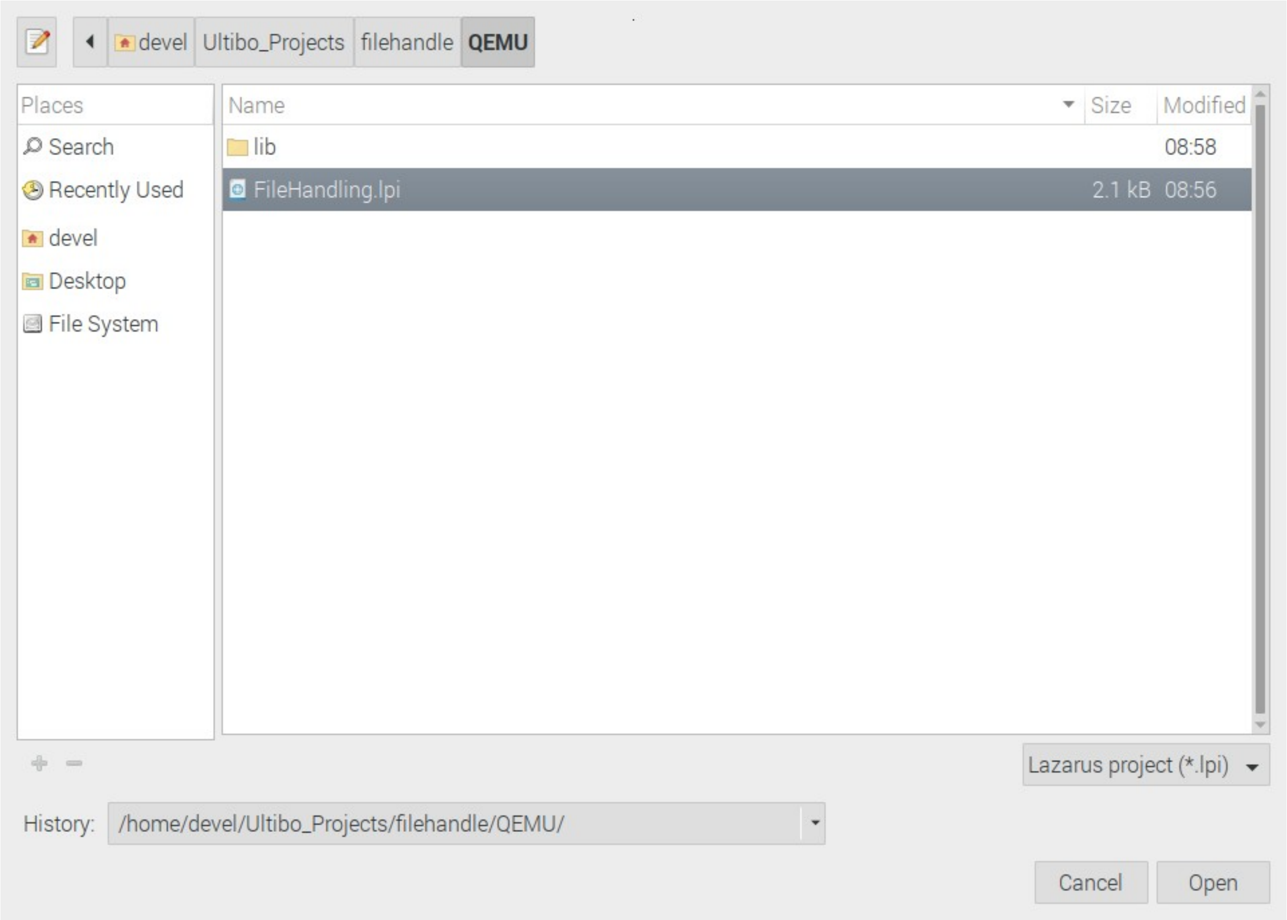


Name	Size	Modified
lib		08:58
FileHandling.lpi	2.1 kB	08:56

Lazarus project (*.lpi)

History: /home/devel/Ultibo_Projects/filehandle/QEMU/

Select FileHanding.lpi

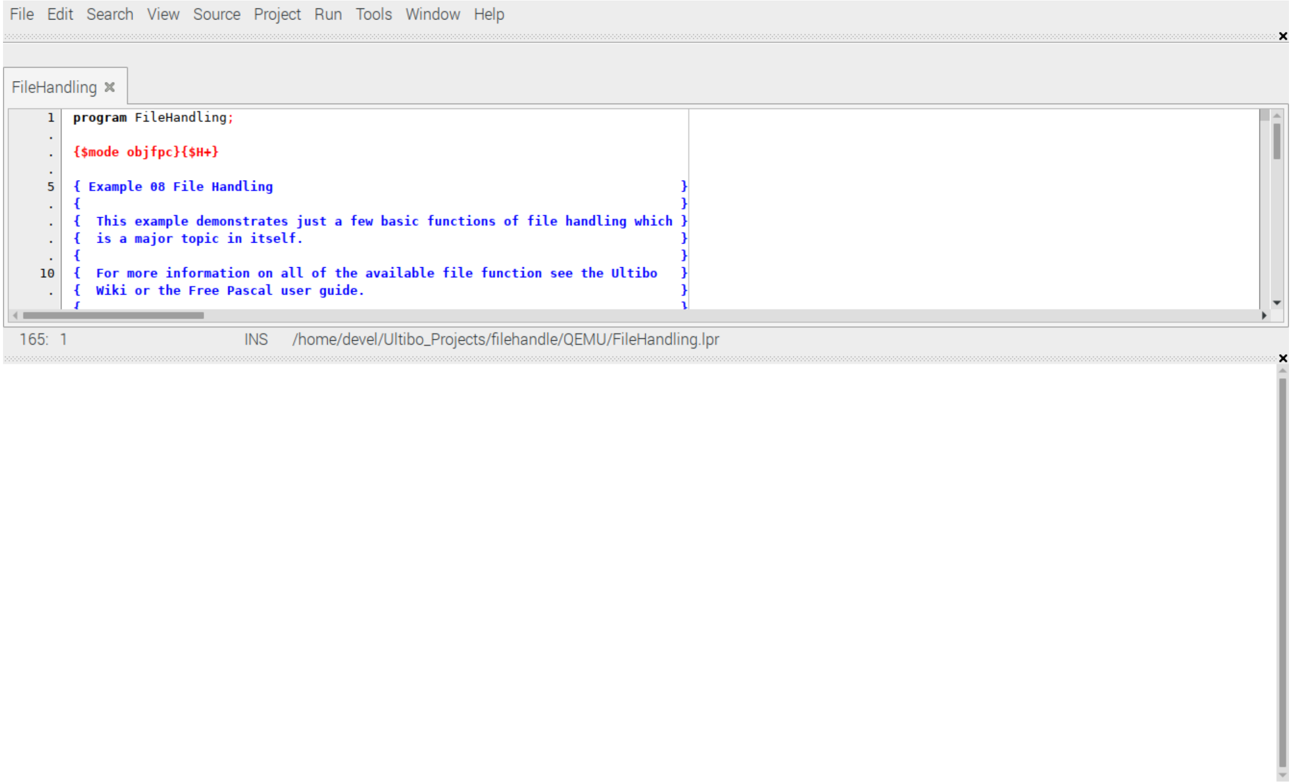


Lazarus project (*.lpi)

History: /home/devel/Ultibo_Projects/filehandle/QEMU/

Depress Open

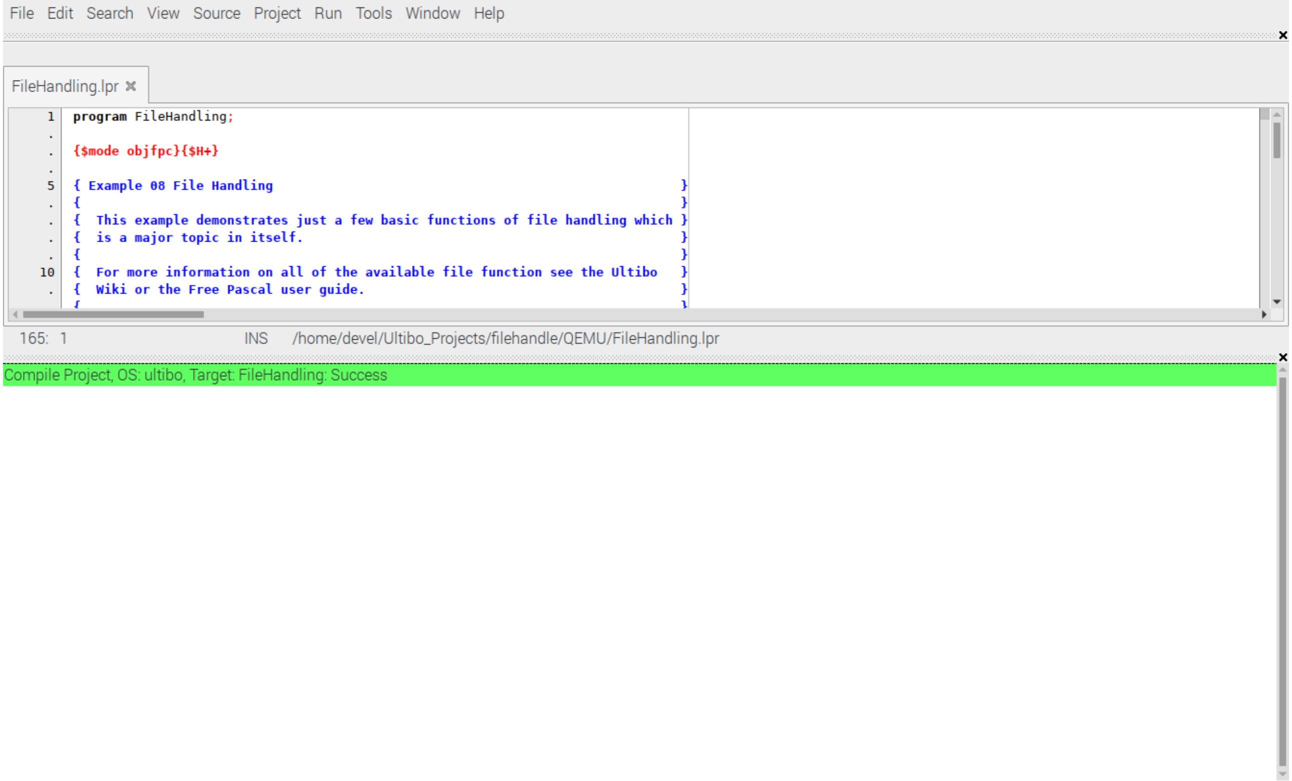
Using Lazarus IDE (Ultibo Edition)



```
program FileHandling;

{$mode objfpc}{$H+}

{ Example 08 File Handling                                   }
{                                                            }
{  This example demonstrates just a few basic functions of file handling which }
{  is a major topic in itself.                               }
{                                                            }
{  For more information on all of the available file function see the Ultibo  }
{  Wiki or the Free Pascal user guide.                       }
```

165: 1   INS   /home/devel/Ultibo_Projects/filehandle/QEMU/FileHandling.lpr

RUN/Compile The kernel.bin is created when the Grean bar appers.

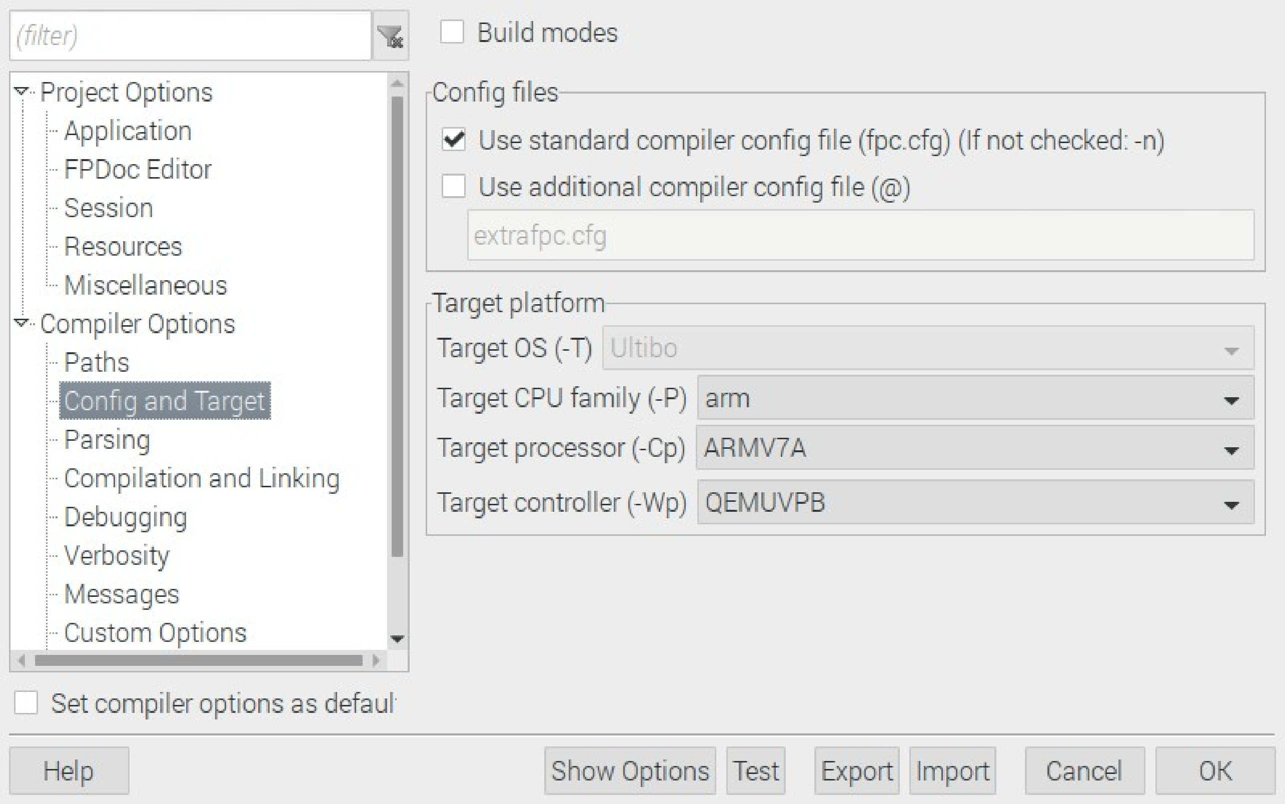


```
program FileHandling;

{$mode objfpc}{$H+}

{ Example 08 File Handling                                                    }
{                                                                             }
{  This example demonstrates just a few basic functions of file handling which }
{  is a major topic in itself.                                                }
{                                                                             }
{  For more information on all of the available file function see the Ultibo  }
{  Wiki or the Free Pascal user guide.                                        }
```

165: 1   INS   /home/devel/Ultibo_Projects/filehandle/QEMU/FileHandling.lpr

Compile Project, OS: ultibo, Target: FileHandling: Success

Project/Project Options/Config and Target



□ Build modes

Config files

☑ Use standard compiler config file (fpc.cfg) (If not checked: -n)

□ Use additional compiler config file (@)

extrafpc.cfg

Target platform

Target OS (-T) Ultibo

Target CPU family (-P) arm

Target processor (-Cp) ARMV7A

Target controller (-Wp) QEMUVPB

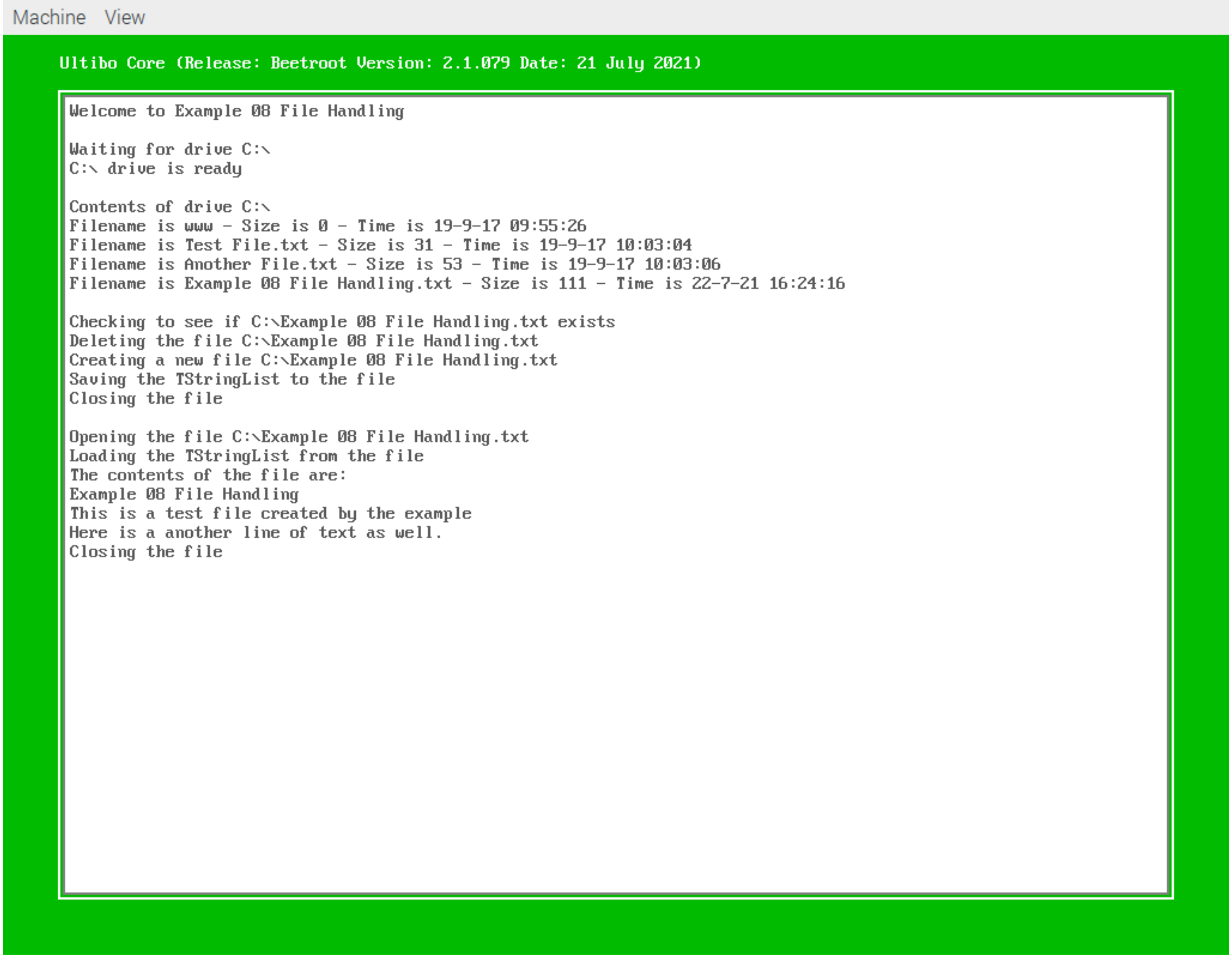
□ Set compiler options as defaul

~~Note: Currently this is not working correctly.~~

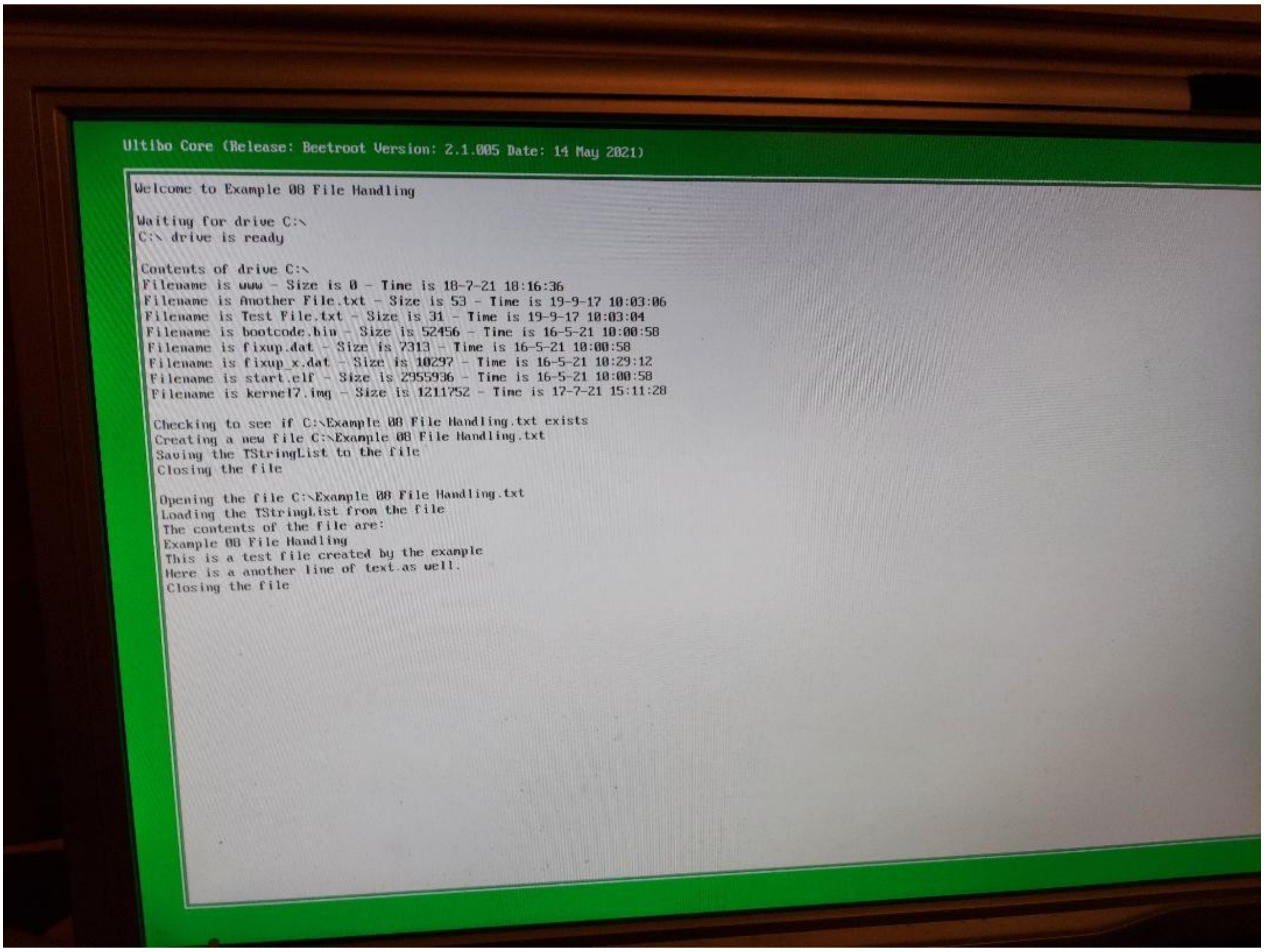
Following the upgrade using ./ultiboinstaller.sh

```
{============================================================================}
{Global constants}
const
 {Version constants}
 ULTIBO_RELEASE_DATE             = '21 July 2021';
 ULTIBO_RELEASE_NAME             = 'Beetroot';
 ULTIBO_RELEASE_VERSION          = '2.1.079';
 ULTIBO_RELEASE_VERSION_MAJOR    = 2;
 ULTIBO_RELEASE_VERSION_MINOR    = 1;
 ULTIBO_RELEASE_VERSION_REVISION = 079;

{============================================================================}
```

QEMU version now works like hardware



```
Ultibo Core (Release: Beetroot Version: 2.1.079 Date: 21 July 2021)

Welcome to Example 08 File Handling

Waiting for drive C:\
C:\ drive is ready

Contents of drive C:\
Filename is www - Size is 0 - Time is 19-9-17 09:55:26
Filename is Test File.txt - Size is 31 - Time is 19-9-17 10:03:04
Filename is Another File.txt - Size is 53 - Time is 19-9-17 10:03:06
Filename is Example 08 File Handling.txt - Size is 111 - Time is 22-7-21 16:24:16

Checking to see if C:\Example 08 File Handling.txt exists
Deleting the file C:\Example 08 File Handling.txt
Creating a new file C:\Example 08 File Handling.txt
Saving the TStringList to the file
Closing the file

Opening the file C:\Example 08 File Handling.txt
Loading the TStringList from the file
The contents of the file are:
Example 08 File Handling
This is a test file created by the example
Here is a another line of text as well.
Closing the file
```



```
Ultibo Core (Release: Beetroot Version: 2.1.005 Date: 14 May 2021)

Welcome to Example 08 File Handling

Waiting for drive C:\
C:\ drive is ready

Contents of drive C:\
Filename is www - Size is 0 - Time is 18-7-21 18:16:36
Filename is Another File.txt - Size is 53 - Time is 19-9-17 10:03:06
Filename is Test File.txt - Size is 31 - Time is 19-9-17 10:03:04
Filename is bootcode.bin - Size is 52456 - Time is 16-5-21 10:00:58
Filename is fixup.dat - Size is 7313 - Time is 16-5-21 10:00:58
Filename is fixup_x.dat - Size is 10297 - Time is 16-5-21 10:29:12
Filename is start.elf - Size is 2955936 - Time is 16-5-21 10:00:58
Filename is kernel7.img - Size is 1211752 - Time is 17-7-21 15:11:28

Checking to see if C:\Example 08 File Handling.txt exists
Creating a new file C:\Example 08 File Handling.txt
Saving the TStringList to the file
Closing the file

Opening the file C:\Example 08 File Handling.txt
Loading the TStringList from the file
The contents of the file are:
Example 08 File Handling
This is a test file created by the example
Here is a another line of text as well.
Closing the file
```

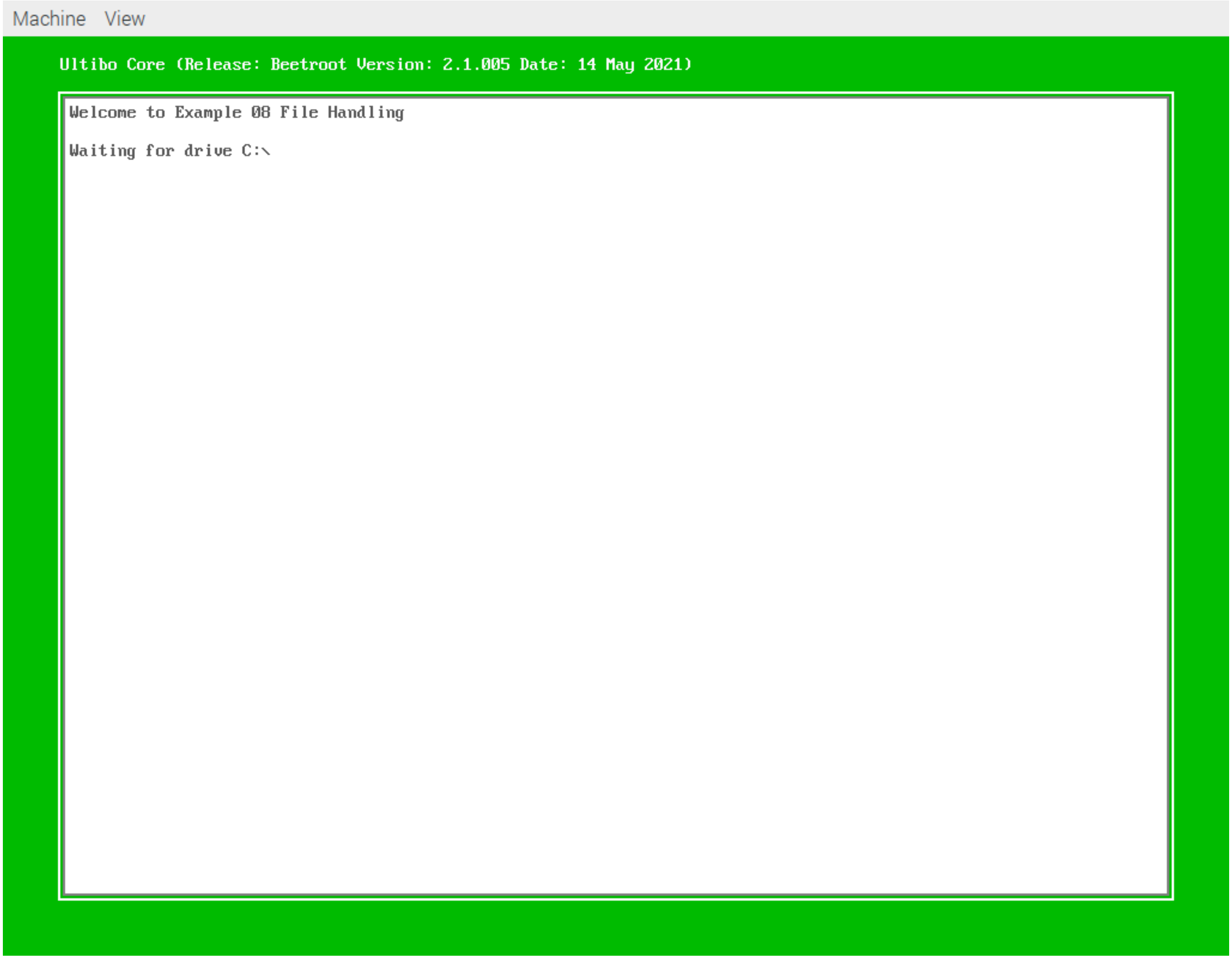
This is an 8Gb

```
Disk /dev/sda: 7.3 GiB, 7864320000 bytes, 15360000 sectors
Disk model: Storage Device
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xa46788b6

Device     Boot Start     End Sectors  Size Id Type
/dev/sda1        2048 2047999 2045952  999M  b W95 FAT32
```

These are the files needed on a micro sd.

```
'Another File.txt'               fixup.dat     start.elf
 bootcode.bin                    fixup_x.dat  'Test File.txt'
'Example 08 File Handling.txt'   kernel7.img   www
```

```
devel@mypi3-20:~/Ultibo_Projects/filehandle/QEMU $ ./startqemu.sh
```



```
Ultibo Core (Release: Beetroot Version: 2.1.005 Date: 14 May 2021)

Welcome to Example 08 File Handling

Waiting for drive C:\
```