Installing Transputer MINIX

Stuart Menefy

10 June 1992

1 Introduction

Transputer MINIX can be installed in a number of different ways, and which you choose will normally depend on the hardware you have available. In general, you will need a 32 bit transputer with at least 1Mb RAM, and some means of providing disk drives (a hard disk is recommended, but not compulsory), and a terminal. However this can be done in a number of different ways, including:

- If the transputer is hosted by a conventional computer, as it is for the majority of transputer tools, this can provide both the terminal and disk drives. In this case it communicates with the transputer through a normal IServer, and MINIX file systems are represented by large files on the host's file system.
- If the transputer card has an RS232 port, this can be used to connect a serial terminal.
- A B005 disk controller board, or similar hardware using an M212 can be connected to provide disk storage directly for MINIX.

It is possible to 'mix and match' these, and possibly other, options to get a working system, however there are two common configurations, 'stand alone' or 'hosted':

	Stand alone	Hosted
Terminal	RS-232 ports	Host computer
Disk drives	B005	Host computer

These are the two options which are described in this document. It should be possible to get one of these systems running in order to generate the desired system. As an alternative it is possible to use another computer and normal transputer tools to cross compile MINIX, and use this to bootstrap Transputer MINIX. This is also discussed briefly.

2 Stand alone

The stand alone operation of Transputer MINIX is the most useful for those people who want to use the Transputer as their main computer.

2.1 Requirements

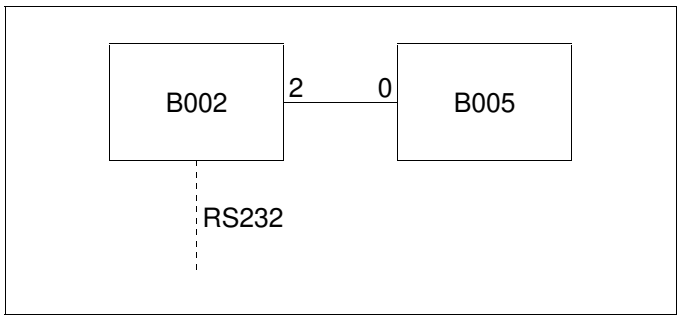
The following are required for installing MINIX in this way:

- A 32 bit Transputer with at least 1Mb memory, and at least one serial port (currently a B002 or B016). Note MINIX is not designed to work with T414A.
- A B005 or similar disk controller using an M212 and providing a hard and floppy disk.

- A terminal.
- Disks:
    - Boot disk.
    - Minimal root filing system (copied to RAM disk).
    - Sources to kernel, MM and FS.
    - Compiler and associated tools.
- About an hour.

2.2 Hardware Setup

This configuration requires that the hardware is set up as shown below, that is, with link 2 of the B002 connected to link 0 of the B005.



The terminal should be configured to run with 8 bits, 1 stop bit, no parity at 9600 baud.

2.3 Booting MINIX

To boot MINIX, the following procedure should be followed:

1 After checking all the connections are as shown above, set the link speeds of both boards to be the same, set the B002 to boot from link, and the B005 to boot from ROM, and auto boot the floppy drive.

2 Turn the power on. The B005 should try to boot from disk, and after a while will set the error flag.

3 Place the boot disk in the floppy drive and hit reset.

4 After a short while the message

```
Booting MINIX 1.5. Copyright 1990 Prentice-Hall, Inc.
Insert ROOT diskette and hit RETURN (or specify bootdev).
```

should appear on the terminal. There may be other messages relating to the hardware which you are running on. These can be ignored for the moment.

5 Eject the boot disk and insert the root filing system, disk, and enter **3,32**, followed by RETURN. The root filing system will be loaded into the RAM disk, printing a message of the form:

```
Memory size = 2048K  MINIX = 156K  RAM disk = 720K Available =  1172K
Loading RAM disk.                                      Loaded 0K
```

(the numbers will vary depending on the system being used). As the disk is loaded the numbers will change, eventually to be replaced by:

```
RAM disk loaded.
```

6 MINIX will then start running the **/etc/rc** file. This may (amongst other things) ask for the time to be entered. Finally you will be prompted to login. When the system is set up, only a limited number of users are set up. For all system administration you must be logged in as root, the password is "Geheim".

Congratulations, you have just booted MINIX.

2.4 Preparing your hard disk

Setting up a hard disk under MINIX is not easy. It requires knowing about certain system administration issues which most people will not be familiar with. This section should make the process as painless as possible, but take some time over it, and make sure that there isn't any data on any disks attached to the B005 that isn't backed up.

Geometry

Before you can prepare your hard disk, you need to know the disk geometry. This is the disk's size, and dictates how much data can be stored on it. In particular:

- the number of cylinders, heads and sectors per track.
- the reduced write current (RWC) and write pre-compensation (WPC) cylinders.
- the interleave and skew values. These are used to optimise disk access times, but may require some experimentation to find the best values for your hardware. An interleave of 6 works well, and both skew values can be set to 0.
- the parking track (also called the landing zone) should be set if your disk requires it, otherwise give a value of 0.

If you are unsure of any of these values, either consult the documentation that came with the drive[1] or one of the freely available hard disk information files.

Partitioning

All hard disks under MINIX must be partitioned. This is a means whereby a single disk can be made to look like several smaller disks, and has several advantages:

- A disk can have different operating systems or different types of filing systems on different partitions.
- If a filing systems becomes corrupted, the scope of the damage should be limited to a single partition.
- On a large disk, it can make some operations more manageable if it is logically divided into several smaller disks.

The decision on now to partition a disk must be taken before any data is placed on it, as this cannot be changed later without losing all the data on the disk. Generally, if your disk is quite small, just make the whole disk one partition, otherwise divide up the disk into a system and user area.

[1] Assuming that you had any of course!

Device files

Each device under MINIX is identified by a special file in the **/dev** directory. The names given to these files in Transputer MINIX are fairly standard. Each name consists of two letters identifying the device type, and a number to identify a particular device of that type. So for example **fd0** is the first floppy disk. In addition, hard disks have a final letter in the name. This identifies the partition within the disk, with **a** representing the whole disk, **b** the first partition, **c** the second, and so on. Finally, some devices are available in both a block and character organisation, in particular disks are always available in both forms. These are distinguished by placing an **r** prefix onto the drive name for the character based device.

These files are used by the system to obtain the major and minor device number of the device which is being accessed. The major number is used to decide which device driver will be used when a device is accessed, and the minor number is passed to the device driver to determine which device should be accessed.

For the B005, the major number is always 3. The minor number gives the drive and partition number, with the drive number in bits 4 and 5, and the partition number in the lower four bits. For the first hard disk, the drive number is 0, and the floppy is device 2. This is why the number entered when booting is **3,32**. For the hard disk, partition 0 is the whole disk, 1 the first partition, 2 the second and so on. The floppy is not partitioned, so this value is ignored. To check the values set up for the files either use **ls -l** or **inodes**.

A summary of the device numbers is shown in the table below:

Major	Minor	Device	Description
1	0	**ram**	RAM disk
1	1	**mem**	User memory
1	2	**kmem**	Kernel memory
1	3	**null**	NULL device
2	*x*	**id***x*	IServer filing systems
3	0	**hd0a**	Whole of winny 0
3	1	**hd0b**	Partition 0 of winny 0
3	2	**hd0c**	Partition 1 of winny 0
3	3	**hd0d**	Partition 2 of winny 0
3	16	**hd1a**	Whole of winny 1
3	17	**hd1b**	Partition 0 of winny 1
3	32	**fd0**	Floppy disk 0
4	0	**tty0**	Terminal 0
4	1	**tty1**	Terminal 1
5		**tty**	Controlling terminal
6		**lp**	Parallel printer port
7	0	**link0**	Link 0
7	1	**link1**	Link 1
7	2	**link2**	Link 2
7	3	**link3**	Link 3

Instructions

1 Boot MINIX as described above.

2 Format the hard disk. To do this use the **format** command, with parameters describing the disk, for example:

```
format -w -dc 615 -dh 4 -ds 17 -rwc 615 -wpc 0 -i 6 -hs 0 -ts 0
   -park 0 /dev/rhd0a
```

(This should be typed as one line, it is shown split over two lines simply to fit it on the page.) These values are correct for the ST-125 commonly found on B005's. The **-w** means that the disk being

formatted is a Winchester disk. The next 9 parameters give the number of cylinders, heads, sectors, RWC, WPC, interleave, head skew, track skew and parking cylinder respectively. Finally the device name for the hard disk is given. This will format the disk, one cylinder at a time, and then verify it. This may take some time.

3 Until now, the device files which have been used have been the default ones provided on the minimal filing system. These are set up correctly for the standard B005 Winchester, partitioned into two equal 10Mbyte partitions. However this may not be suitable for the disk being installed, and so have to be changed. How the devices are currently set up can be seen by examining the file **MAKEDEV** in **/dev**. This gives the **mknod** commands which were used to create the devices, and should be updated if you make any changes. In particular check that the correct number of device files are made for the number of partitions you require (both block and character special files), and that the size given for block special files is correct (the final argument to **mknod** in Kbytes). If any of these values are incorrect, the device files must be removed, and re-made using **mknod**.

4 Next the hard disk must be partitioned. To do this, the **fdisk** command is used. This is an interactive program which will display the current partition table, and allow changes to be made to it. **fdisk** will try and guess the number of head and sectors on the disk, as it does not have access to them directly, however it can be told using the **-s** and **-h** options:

```
fdisk -h4 -s17 /dev/hd0a
```

Frequently **fdisk** will flag possible problems with the current partitioning, where there was a problem with early versions of MINIX. These can be ignored for Transputer MINIX.

5 Once the disk is partitioned, a filing system must be made in each partition. This is done using **mkfs**, which simply takes the name of the partition and the size. So for example:

```
mkfs /dev/hd0b 20908
```

Take care if copying partition sizes from **fdisk**. **fdisk** partition information is displayed in terms of blocks, which are conventionally 512 bytes, however **mkfs** requires the disk size in zones, which are 1024 bytes.

6 Finally, the new filing system can be mounted off the root filing system. This simply means specifying the device which is to be mounted, and where it should appear on the root filing system (which must be the name of a directory which already exists). For example:

```
mkdir /user
mount /dev/hd0c /user
```

It should now be possible to access this directory like any other, and the hard disk will be used for storing the files.

2.5 Recompiling MINIX

Once a hard disk has been set up for Transputer MINIX, it is useful to be able to make the hard disk the root filing system. This overcomes having to load the root filing system from floppy, hence freeing up memory, and ensuring that any changes to the root file system are written directly to disk without having to specifically mount the root file system image and modify this.

However, this change requires modifying the MINIX kernel. This is a simple process, and ensures that the kernel sources are available. The file which has to be modified is the standard include file **minix/boot.h**, and so is located off **/usr/include**. This contains a definition of **DROOTDEV**, which will be set to **DEV_RAM**. Simply change this to be **DEV_HD0**, and save the file.

Next move to the source directory, **/usr/src**. This contains several directories of source for different parts

of MINIX, in particular the kernel is made from three parts, **kernel**, **fs** and **mm**, and various other bits and pieces in **tools**.

minix/boot.h is used by the kernel file **system.c**, however it is not specified as a dependency in the makefile, and so recompilation of this file must be forced by hand by removing the file **kernel/system.o**.

To build a new boot disk, place a blank floppy into the drive, move into the **tools** subdirectory, and type:

```
make disk "FORMAT=-f"
```

This will first compile a new version of **kernel/system.o**, re-link the kernel, and rebuild it. Then the disk will be formatted, and the new kernel written onto it.

Finally, before trying to boot off this this disk, set up the hard disk so that there is enough of the utilities MINIX requires to boot. In particular the **/etc** and all the files in it, and the files in **/bin**, in particular **login**, **sh** and any other commands used by **/etc/rc** or alternative shells.

3 Hosted

The hosted operation of Transputer MINIX is the most useful for those people who use Transputers hosted by their normal computer, and want to run MINIX as a normal transputer application.

3.1 Requirements

The following are required for installing MINIX in this way:

- A 32 bit Transputer with at least 1Mb memory, connected to the host through a link.
- A host computer with hard and floppy disks.
- An interface between the host and the transputer.
- An IServer capable of talking to the transputer.
- Disks containing:
    - MINIX boot file.
    - MINIX root filing system.
    - Sources to kernel, MM and FS.
    - Compiler and associated tools.
- About an hour.

3.2 Installation

Before MINIX can be run on a hosted system, the following files must be placed in the current directory:

```
minix.btl
0
```

minix.btl is the MINIX executable which will be booted onto the transputer. **0** is the root filing system which is used by MINIX, and may be supplied compacted in some way, as the file is at least 1Mbyte, possibly larger.

3.3 Booting MINIX

To boot MINIX, simply type:

```
iserver /sb minix.btl /se
```

or whatever command is used to start the IServer and boot the file **minix.btl** onto the transputer. This will then display:

```
Booting MINIX 1.5. Copyright 1990 Prentice-Hall, Inc.
Memory size = 2048K  MINIX = 156K  RAM disk = 100K Available =  1792K
```

(the numbers will vary depending on the system being used). This may then be followed by some information about the IServer which MINIX thinks you are using. If you have problems getting MINIX to run check that these values are correct, as MINIX tries to optimise server usage by using the features provided by later IServers where available.

MINIX will then start running the **/etc/rc** file. This will (amongst other things) ask for the time to be entered. Do so. Finally you will be prompted to login. The root password is “Geheim”.

Congratulations, you have just booted MINIX.

3.4 Creating new filing systems

As the hosted version of MINIX uses host files for filing systems, some of the problems associated with setting them up are reduced. However it is still worth reading the section on using a hard disk (section 2.4) for details on major and minor devices etc.

MINIX file systems are stored in simple binary files, with the name being a single digit. This corresponds to the minor device number in the **/dev/** file, thus the file **/dev/host0** is stored in **0** (and is the root filing system), **/dev/host1** is stored in **1** and so on. (The generic name **host** is used for all file systems on the host, although in the long term will probably change to **id**, as they are ‘drives’ accessed through the IServer.) Currently the number of hosted file systems is limited to 4, however this can be easily changed by re-compiling the host file system device driver in MINIX. Files are only ever read from the current directory, so it is possible to have different directories with different filing systems in them.

When MINIX creates a new filing system, any data in the file will be lost, and so you should check that there are no files with the same name which can be overwritten. The file must already exist for MINIX to use it, so it should be created outside of MINIX. This can be done using **touch** if available, or simply copy a small file to the current directory with the correct name. MINIX will make the file larger if required, but cannot shrink a file which is too large, and so the extra space which the file occupies will be wasted. Note that once a filing system has been made, it cannot be made any larger without making a new filing system, which will destroy any data in the file system.

To make the new filing system, follow the steps described in 2.4 Instructions step 3 for checking that the device files are correct. In particular check that the size given for the block special files is correct for the size of the file system you intend to make. Then simply make the file system by using **mkfs**. For example:

```
mkfs /dev/id1 1024
```

would make a 1Mbyte file system in the file **1**.