MaRTE OS FAT16 Filesystem

Daniel Sangorrin

daniel.sangorrin@gmail.com

2009-3-4

Revision History

Revision 0.1	2009-2-26	Revised by: dsl
first draft		
Revision 0.2	2009-3-4	Revised by: dsl
added Jean-Louis suggestions		

This document describes how to configure and use the FAT16 filesystem driver that comes with MaRTE OS.

Table of Contents

1. Preface

This document is intented for those who want to know how to configure and use the FAT16 filesystem driver on MaRTE OS. Note that this manual is aimed at the x86 architecture of MaRTE OS.

1.1. Feedback

Feedback is most certainly welcome for this document. Send your additions, comments and criticisms to the following email address : <`daniel.sangorrin@gmail.com`>.

2. The MaRTE OS FAT16 filesystem driver

Although it is not required by the POSIX minimal profile, MaRTE OS contains a basic implementation of a FAT16 filesystem driver. The driver provides the means to read, write, create and delete files from a FAT16 filesystem. The source code is contained at `marte/x86_arch/fat` and it follows the Microsoft (C) FAT16 specification. In its current implementation there are several limitations:

1. It does not provide concurrency facilities (ie, protected objects or mutexes). In a typical architecture there will be only one "Logger" task writing to the disk so we do not want to introduce an extra overhead. If you want to use the driver from different tasks you need to provide mutual exclusion access to the functions by yourself.
2. Only supports short filenames (XXXXXXXX.XXX) without extension, that is, EIGHT characters maximum (if you put extension it is ignored so file.txt is the same as file.dat and you open it by using the path "/fat/file"). The reason is that for Long filenames there are some Microsoft (R) software patent issues.
3. It can not mount more than one FAT16 partition at a time. This is is to keep the code simple.

In MaRTE OS the FAT16 driver can be used in three ways:

1. Through the Ada.Text_Io package for Ada applications.
2. Through a POSIX interface (open, read, write) mainly for C/C++ applications but also for Ada applications by using the POSIX5 bindings
3. As a non-standard library with our own interface. In this case you don't need to install the driver in the MaRTE OS driver filesystem, just use it as a static library that you link your application with. This is the option with least overhead but it makes your application non-portable so it is not recommended.

3. Preparing your FAT16 partition

MaRTE OS FAT16 filesystem driver can read-write files to a FAT16 filesystem but the filesystem must be formatted before with a partition and formatting tool. Here we describe how to partition and format a CompactFlash card (or any other IDE disk) using GNU/Linux. For a QEMU image disk (to try it on the QEMU emulator) you can check the MaRTE OS tutorial for QEMU ([1]).

1. Unmount any filesystem mounted on the drive if there is one (i.e.: **umount /media/usbdisk**)
2. Create a FAT16 partition. (i.e.: with **cfdisk** or **fdisk**). For the FAT16 filesystem you have to select the 06 code. For the partition size use one that fits your needs and is lower than 2GB (the maximum for FAT16). For example, 20-100 MB. Do not be too greedy with the size because the driver is in experimental phase and big sizes are less tested. If you want to boot from the same partition toggle the bootable flag as well.
3. Format the partition: **mkdosfs -F 16 /dev/sda1**. Make sure that you are formatting the partition (sda1), not the whole disk (sda).
4. After the previous steps you should have your drive partitioned and formatted so you can mount it with **mount /dev/sda1 /mnt** and create a test file: **vi /mnt/toto**

If you want to boot from the drive you can install GRUB in the same partition:

```
# mkdir -p /mnt/boot/grub
# cp /boot/grub/* /mnt/boot/grub/
# vi /mnt/boot/grub/menu.lst
# umount mnt/
# grub
>> root (hd1,0)
>> setup (hd1)
>> quit
```

4. Installation

In order to use the FAT16 filesystem driver you need to install it in MaRTE OS (except if you want to use it as a library without a standard interface). The FAT16 driver is installed in the same way as any other driver in MaRTE OS, which is described in the MaRTE OS user's guide. The main steps you have to follow are:

1. In `marte/x86_arch/arch_dependent_files/marte-kernel-devices_table.ads` uncomment the FAT driver related lines. Do not modify the name of the device file because the driver is expecting it the way it is, just uncomment lines so it looks like this:

```
...
with Fat_Driver_Functions;
...
9 => (Name   => "FAT Driver      ",
      Create => Fat_Driver_Functions.Create'Access,
      Remove => null,
      Open   => Fat_Driver_Functions.Open'Access,
      Close  => null,
      Read   => Fat_Driver_Functions.Read'Access,
      Write  => Fat_Driver_Functions.Write’Access,
      Ioctl  => null,
      Delete => Fat_Driver_Functions.Delete’Access,
      Lseek  => Fat_Driver_Functions.Lseek’Access),
...
13 => ("/fat/           ", 9, 0, True, Mount_Point, False, 0),
```

2. Configure the FAT16 disk and partition number (remember that the driver can only mount one FAT16 partition at a time). This step is very important and you need to know the disk and partition number that you want to mount. The variables are in `marte/x86_arch/drivers/fat/fat_driver_functions.ads`

   ```
   Disk : constant IDE.Drive := IDE.hdc;
   Part_Num : constant IDE.Partitions.Partition_Number := 1;
   ```

3. Recompile MaRTE OS kernel with **mkmarte**
4. Test the installation with some examples in `marte/x86_arch/drivers/fat/`

5. Troubleshooting

1. At initialization the FAT16 driver fails

Check that the FAT module is configured for the appropiate device (hda, hdb, hdc, hdd) and partition number

2. IDE problems

If you have problems reading the sectors check that the IDE module is working ok (x86_arch/drivers/ide). The IDE driver has some delays inside (of 400ns). In MaRTE OS check that your "Minimum_Suspension_Time" parameter at marte-configuration_parameters.ads is bellow 400ns.

3. File ’xxxx’ not found but it exists!

Maybe you tried to open without the complete path. The FAT16 filesystem is mounted on `/fat/` directory and therefore all files are open using a path like `/fat/xxxx`. The filename can be up to eight characters and can’t have extension (check the limitations of the driver in Section 2)

6. Further Information

1. Hello MaRTE OS using an emulator http://marte.unican.es/documentation/tutorials/hello-marte-howto.pdf (http://marte.unican.es/documentation/tutorials/hello-marte-howto.pdf)

7. Legal section

7.1. Copyright and License

Linux is a registered trademark of Linus Torvalds.

7.2. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.