Using Flash as the Root File System

Application Note AN-449

Notes

By Pallathu Sadik

Introduction

IDT provides a royalty free, license-fee free Linux operating system environment for its Interprise™ family of integrated communications processors. In addition to the basic operating system and device drivers, IDT provides higher level functions common in gateways and other applications. This application note describes how to use the file systems residing in flash devices as root file systems.

For more details on IDT Linux, please refer to: http://www.idt.com/products/docs/Linux_ds.pdf.

Overview

The majority of IDT evaluation boards, as well as end products deployed in the field by users of IDT processors, have one or more flash devices on the system. These flash devices can be used to store Linux kernel images, Linux applications or both. IDT Linux supports a variety of flash devices and file systems suitable for flash devices. It also provides a feature which allows the file system residing within the flash to be mounted as the "root" file system. This feature is very useful in systems that have limited main memory (typically SDRAM or SRAM). An alternative would be to use "ramdisk" as the "root" file system. However, "ramdisk" consumes main memory and leaves very little space for executing kernel and applications. For example, on systems that have 16 MB of main memory, if 8 MB is used for "ramdisk", less than 8 MB will be available for running kernel and applications.

Flash Devices on IDT's 79EB438 Evaluation Board

IDT's EB438 evaluation board has two flash memory devices on the board. This document explains how to create Journaling Flash File System, version 2 (JFFS2) on a flash device and mount it under Linux as the "root" file system. This document uses the EB438 evaluation board as an example. However, the concepts described here are applicable to any system which uses an IDT processor running IDT Linux.

The EB438 evaluation board has two flash memory devices on the board as shown in Table 1.

Flash	Size (MB)	Port Width (bits)	Physical Address (hex)
AM29DL163DT	2	16	08000000
AM29DL323DT	4	16	11000000

Table 1 EB438 Flash Memory Devices

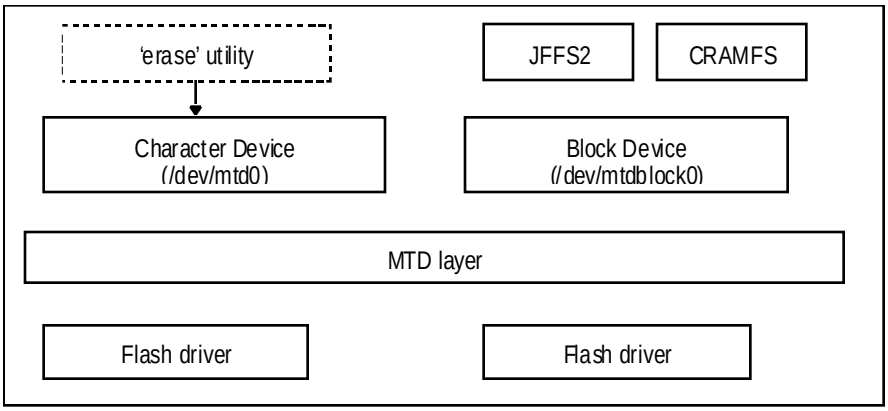
The board also has an EEPROM located at address 0x1FC00000 - the power reset address. Therefore, boot code such as IDT/sim or IdtBoot resides within this EEPROM. In this document, the 4 MB flash will be used for storing the file system

Memory Technology Devices (MTD) Overview

IDT Linux provides device drivers for a variety of flashes. It also provides an intermediate software layer which offers an interface through which upper layers of software such as file systems can interact with device drivers at the lower level. This intermediate layer is called MTD (Memory Technology Devices). The

Notes

advantage of this approach is that the upper layer does not need to know the specifications of the device used to store data/files and the device driver can remain completely unaware of the needs of the application using the data/files. A simplified view of the software layers is shown in the figure below.



Journaling Flash File System, Version 2 (JFFS2)

JFFS2 is a file system suitable for embedded devices and is the one we will discuss in this document. It is a compressed file system like CRAMFS (Compressed ROM File System). Unlike CRAMFS, which is read-only, JFFS2 is a read-write file system. For more details on JFFS2, please refer to http://sources.redhat.com/jffs2/.

Kernel Configuration for MTD and JFFS2

The remainder of this document assumes that the reader is reasonably familiar with IDT Linux and has hands-on experience with it. Many routine steps in the build and test procedure have been skipped here in order to simplify this document and to focus on the issues of interest to system designers. For a detailed description of the skipped steps, please refer to the IDT Linux release documents you received with the Linux product.

While configuring the kernel for the EB438 processor, select the "Memory Technology Devices (MTD)" option from the main menu to display the screen shown below. Enable the MTD options on this screen as indicated by the asterisks.

Memory Technology Devices (MTD)
<> Memory Technology Device (MTD) support*
[ ] Debugging
< > MTD partitioning support
< > RedBoot partition table parsing
--- User Modules And Translation Layers
<> Direct char device access to MTD devices*
<> Caching block device access to MTD devices*
< > FTL (Flash Translation Layer) support
< > NFTL (NAND Flash Translation Layer) support
RAM/ROM/Flash chip drivers
Mapping drivers for chip access
Self-contained MTD device drivers
NAND Flash Device Drivers

Screen 1

Notes

Now select "RAM/ROM/Flash chip drivers" option in Screen 1 to display the Screen 2, shown below. Enable the options on this screen as indicated by the asterisks.

```
<*>  Detect flash chips by Common Flash Interface (CFI) probe
< >  Detect non-CFI AMD/JEDEC-compatible flash chips
[ ]  Flash chip driver advanced configuration options
< >  Support for Intel/Sharp flash chips
<*>  Support for AMD/Fujitsu flash chips
< >  Support for RAM chips in bus mapping
< >  Support for ROM chips in bus mapping
< >  Support for absent chips in bus mapping
[ ]  Older (theoretically obsoleted now) drivers for non-CFI chips
```

Screen 2

Return to Screen 1 and select the "Mapping drivers for chip access" option to display the following screen. Enter the values that are underlined.

```
<*>          CFI Flash device in physical memory map
(11000000[1])  Physical start address of flash mapping
(400000[2])    Physical length of flash mapping
(2[3])         Bus width in octets
```

Screen 3

Notes 1, 2, 3 above: The values in the example shown above are specific to EB438 evaluation board.

[1] This address will depend on board design and configuration. 11000000 shown above is for the location of the 4MB flash on the EB438.

[2, 3] This will change depending on the size and type of the flash used.

Now enable support for JFFS2 file system in kernel under "File Systems" menu, shown in the screen below.

```
File Systems
.....
.....
[*] Journalling Flash File System v2 (JFFS2) support
.....
.....
```

Screen 4

Since JFFS2 on flash will be used as the root file system, the ramdisk support may be disabled by un-checking the "Block Devices -> Ramdisk" option. Save the configuration. We are ready to build the kernel.

Notes

Building a JFFS2 File System

In order to build a JFFS2 file system, a utility called "mkfs.jffs2" is required. This utility is generally available in the public domain, possibly on your desktop system. It is also provided by IDT in releases after March 17, 2004.

The procedure for building JFFS2 file system is given below.

1. On your development system host, create a new directory and copy the files that you wish to have within your flash root system into that directory. Let's call this directory "stage".
2. Run the mkfs.jffs2 utility. The syntax for mkfs.jffs2 is as follows:

   mkfs.jffs2 -r <directory> -o <outfile> [-l | -b] [-p] [-e]

   where:

   directory: *The directory created in step 1 above*

   outfile: *The file that can be programmed to flash*

   -l or -b: *Little or Big Endian respectively*

   -p: *pad to end of the block*

   -e: *erase block size.*

Issue the command "mkfs.jffs2 -help" for details on available options.

In our case, since we are going to implement the root file system on the flash, the command can be used as shown below:

mkfs.jffs2 -r stage -o root.jffs2 -p

This will generate root.jffs2 under the current directory. Make sure that the "stage" directory contains all the required files for the root file system.

Programming Flash

Using IDT/sim or IdtBoot, the file created above can be programmed into the flash on the board using the *fb* command as shown below:

fb -f 0xb1000000 root.jffs2

This command will program the file 'root.jffs2' which has the JFFS2 file system into the second flash (4 MB) on board.

Booting Linux

First, the kernel needs to be made aware that the flash is to be used as the root file system. This can be achieved by setting boot parameters under IDT/sim or IdtBoot as shown below:

<IDT>set bootparm1 "root=/dev/mtdblock0 rootfstype=jffs2"

Next, the Linux image can be booted. After linux is booted, typing the *mount* command will show the devices that are mounted. The result should be as shown below:

/dev/mtdblock0 on / type jffs2 (rw)

/proc on /proc type proc (rw)

The first line shows that the root device is mounted on */dev/mtdblock0*. You may also run the *free* command to see how much memory remains available in the system. It will approximate the actual physical memory available.

Notes

Conclusion

By following the steps described in this document, any system using an IDT processor running IDT Linux will be able to use file systems residing in the flash devices as the root file system, allowing system developers to fully utilize the main memory for running kernel and applications.

References

http://www.linux-mtd.infradead.org/

http://sources.redhat.com/jffs2/

IDT 79EB438 Evaluation Board Manual

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.