

Readme File: Version 1.0 – January 27, 2001

1 Overview

This CodeKit software contains a Linux patch file for the 2.2.18 kernel. After this patch file is applied, the resulting kernel binary will support boot-from-Flash from the NetSC520 board's 16-Mbyte flash bank, using the Journaling Flash File System (JFFS) to read and write files to and from Flash memory. No hard drive is needed.

This patch file is based on an early version of the Memory Technology Device (MTD) driver subsystem included in the 2.4.0 kernel. Further information on MTD and JFFS is available at www.linux-mtd.infradead.org.

For support: e-mail epd.support@amd.com or call 1-800-222-9323
--

1.1 Tool and System Requirements

The following target hardware and tools are required to build and execute this CodeKit software.

ITEM	DESCRIPTION
Target Hardware	NetSC520 Demonstration Board, Revision 1.0 or later
Development System Hardware	NetSC520 Demonstration Board with laptop hard drive, plus second computer (any suitable type) to be used for serial and telnet connections; <i>or</i> NetSC520 Demonstration Board, plus Linux development system, plus additional Windows-based system with Macraigor Raven-AMD device.
Monitor	None
Development System's Operating Systems	Linux kernel version 2.2.18; (optional) Windows® NT 4.0, Windows 95, or Windows 98
Tool Suite	GCC 2.73 or EGCS 1.3
Compiler*	GCC 2.73 or EGCS 1.3
Assembler**	Included with compiler
Linker*	Included with compiler
Other CodeKits	Remon CodeKit, CK002803

There are several ways to build this CodeKit. The simplest, but slowest way, is to use a single NetSC520 demonstration board with Linux on a hard drive to build the system and copy the kernel and file system image into Flash memory. A faster but more complicated way is to use a faster PC running Linux to compile the kernel, and then use a third, Windows machine to run Remon with the Macraigor Raven device to copy the images into the NetSC520 board's Flash memory. (While it is possible for a single PC to run Windows and Linux in a dual boot configuration, that is not recommended for the system used to build and download this CodeKit.)

Both recommended approaches to building this CodeKit are discussed below.

1.2 Installing the CodeKit Software

This CodeKit software is delivered as a self-extracting executable file. In Windows, double click on the EXE file to create a directory called C:\AMD\CK005900. Some of the files will need to be copied to the Linux development system.

Under Linux, type `unzip ck005900.exe`. This will uncompress the file and create a CK005900 directory in the current directory.

1.3 Files and Directories

The EXE file contains the following files.

FILE	DESCRIPTION
2.2.18-mtd.patch	Patch file for a 2.2.18 kernel
Flash.img	A minimal JFFS file system image based on Embedix 1.0
mtd.tar.gz	MTD code and utilities as a separate file
zFlashImage	An MTD-enabled kernel binary image
License.pdf	Licensing agreement for the software contained in the CodeKit.
Readme.pdf	This document in Adobe Acrobat PDF.

1.4 Setting Up the Build Environment

A Linux installation, either on a fast workstation or on a hard-drive equipped NetSC520 board, is necessary to build the kernel image. The distribution should not matter, except that the kernel may have problems if it is built by gcc-2.96, which is included with Red Hat 7.0. All other distributions that include a 2.2 kernel should work fine.

Whichever distribution is used, the necessary components to build a kernel must be installed. Because this can vary from distribution to distribution, it is impossible to give more details.

1.5 Setting Up the Target Hardware

Attach two serial cables to the serial ports on the NetSC520 board, and then to a computer that is running Hyperterm or some other serial interface program on each of its two ports. Configure the computer's serial ports for 9600 baud, 8 bit, no parity, 1 stop bit operation. The system log from the NetSC520 will be piped out along these cables.

If the Flash kernel is to be built on the NetSC520 board, you must first attach a hard drive that already contains a suitable Linux installation. This will be your Linux development system. Login to the board's Linux console from another computer, via Telnet over an Ethernet connection. This second computer can be the same one to which the board's serial cables are connected.

If the Flash kernel is to be built on a separate development system, you need two additional computers besides the NetSC520 board. The separate Linux development system is used to build the Flash kernel. Then a Windows-based system is used to program the NetSC520 board's Flash memory via REMON software and a Macraigor Raven-AMD device. (Either computer can be used to display the board's serial port output.) The Raven device connects the Windows system to the NetSC520 board's AMDebug port. Position the connector on the AMDebug port so that pin 1 is away from the edge of the board. Set up the Raven device as described in the REMON CodeKit, CK002803.

1.6 Building the CodeKit Software

Copy the mtd.tar.gz file into a directory owned by the root user of the Linux development system. Using a command shell, untar it by typing `tar xzf mtd.tar.gz`. This will create a new directory mtd with the MTD source and utilities files in it. Change directory into it by typing `cd mtd/utilities` and run the `MAKEDEV` script. This will create the MTD devices in the workstation's /dev directory, which is necessary for building the final kernel image.

Download the 2.2.18 kernel from <http://www.kernel.org/pub/linux/kernel/v2.2/linux-2.2.18.tar.gz> and move it into the /usr/src directory on the Linux development system. Delete the symbolic link named "linux" in that directory. This will prevent the new kernel source files from overwriting the old kernel source files.

Using a command shell, untar the 2.2.18 package by typing `tar xzf linux-2.2.18.tar.gz`. This will create a new directory named "linux" with the 2.2.18 source files in it. Rename this directory by typing `mv linux linux-mtd-2.2.18`, and then create a symbolic link to it by typing `lnk -s linux-mtd-2.2.18 linux`. Copy the 2.2.18-mtd.patch file from this CodeKit into the /usr/src/linux directory and change directory into it.

Patch the kernel source files by typing `patch -p1 < 2.2.18-mtd.patch`. This will change a few source files and add the MTD and JFFS source files. It will also create a ".config" file especially set up for the NetSC520. Create a symbolic link to the asm directory by typing `lnk -s include/asm-i386 include/asm`. Build the flash kernel by typing `make clean dep zImage`. This will create a zImage in /usr/src/linux/arch/i386/boot that will boot from the hard drive, but will not boot from Flash memory because it has the wrong root device. Type `./mzfi` to run the mzfi script, which will create a zFlashImage file in your /usr/src/linux directory. This file needs to be placed at the base of the Flash on the NetSC520.

If you are programming the Flash memory from a hard drive attached to the NetSC520, you need to have built the zImage as described above, loaded it onto the NetSC520 board's hard drive and, run the MTD MAKEDEV script. Configure LILO to boot from that kernel (consult the KERNEL-HOWTO at www.linuxdoc.org for more information on this subject) and reboot the machine. The new kernel will understand how to write to Flash memory. To install the kernel into Flash, type `cat zFlashImage > /dev/mtd0`. To install the file system image, type `cat Flash.img > /dev/mtd2`.

If you are programming the Flash memory via the Macraigor Raven device, transfer the zFlashImage binary file to a computer with the Macraigor wiggler. Run the REMON remote monitor (available as a CodeKit CK002803 from AMD) and execute the following commands:

```
z
yn
ns zFlashImage 0
ns Flash.img 100000
z
```

These instructions will copy the Flash kernel and JFFS file system image into Flash memory.

1.7 Running the CodeKit Software

Reset the NetSC520 board and reboot it. During the BIOS start-up, you must set up the Flash boot sequence if it is not set up already. Use the Tab key to select the Custom Configuration menu item and press Enter. Tab over to the Linux Flash Boot Loader item and set it to Enabled. Hit Esc to return to the Main Menu, and then select Write to CMOS and Exit.

The NetSC520 BIOS will finish executing and begin running the Linux Flash boot loader. After a few seconds, a standard Linux login prompt will appear on one of the consoles. Login as `root` with password `password`. To make your system secure, you should change the root password immediately. You may also telnet to the NetSC520 with a default IP address of 163.181.222.136.

The sample file system included with this CodeKit is a very stripped down version of Linux, missing many standard commands. It is based on the Embedix distribution maintained by Lineo, and they will provide limited support for users of this CodeKit.

The IP address for the NetSC520 is located in `/etc/rc.d/init.d/S20network`. If you plan to use the NetSC520 in a different network, you will need to change this file. The netmask, broadcast, and gateway lines will need to be changed to conform with the new IP address.

1.8 Building a new file system

It is possible, though difficult, to build a new file system image. The simplest way to do this is to get Linux running from a hard drive on the NetSC520 board using the Flash-enabled kernel and MTD MAKEDEV script as described above. Then create a hard drive partition sized under 14 Mbytes. Install a Linux system onto this hard drive, which will be very difficult because most desktop installations will not fit into 14 Mbytes. AMD recommends the use of the Embedix SDK for this procedure.

When a working Linux system has been created on the partition, use the “`mkfs.jffs`” command in `mtd/util/` to create a `Flash.img` binary image from the small partition. This file can then be written to Flash as described above.

2 About the Software

The software in this CodeKit is based on the Memory Technology Devices (MTD) for Linux code developed by David Woodhouse and others, the Journaling Flash File System (JFFS) by Axis Technologies and others, and the Embedix Software Development Kit by Lineo. As with all Linux projects, these projects are constantly under development and more mature versions are probably available. The source code in this CodeKit is based on the January 11th snapshots of the MTD and JFFS code and on Embedix 1.0.

Although some attempt will be made to keep this CodeKit up-to-date with its parent code bases, the rapid pace of Linux development means that it will rarely be completely up-to-date. Users are warned that syncing up to the latest Linux kernel, MTD or JFFS versions may cause the system to fail to work, perhaps unpredictably.

In order to provide a greater understanding of the CodeKit, there is a brief summary of the important files below. Files labeled (HOST) are on the host or build Linux system, while files labeled (TARGET) are in the Flash drive of the NetSC520.

2.1 (HOST) /usr/src/linux/drivers/mtd/e520.c

This is the mapping file for the NetSC520 board. It tells the rest of the MTD system that the NetSC520 uses physically mapped memory, divided into four partitions. One of these partitions is intended for the Flash kernel, two are set aside for BIOS, and the remaining partition is used by the file system. This file should not change much as MTD evolves.

If this CodeKit is being used on the Élan™SC520 microcontroller customer development platform (CDP) or on another NetSC520 board with a different flash configuration, this file will need to be modified so that the `WINDOW_ADDR` value points to the absolute, physical address of the Flash memory.

2.2 (HOST) /usr/src/linux/drivers/mtd/cfi_probe.c

This file contains instructions on how to probe for Common Flash Interface (CFI) compliant devices that have been mapped into memory by a mapping file, such as `e520.c`. It should also be stable as MTD evolves, but if a user attempts to sync up to the current MTD snapshot, discrepancies in assumptions may appear here.

2.3 (HOST) /usr/src/linux/drivers/mtd/cfi_cmdset_0002.c

This file contains the instructions to unlock and program Flash chips that conform to the standard AMD Flash algorithms. It may disappear from the MTD code base in the future or be merged with cfi_cmdset_0001.c. As it evolves, discrepancies may appear between it and e520.c.

2.4 (HOST) /usr/src/linux/arch/i386/boot/bootsect.S

This file contains the instructions for booting the Linux kernel. It performs some set-up operations and moves the kernel boot code into the proper location. Since this code is ignored by LILO, it is safe to use a kernel compiled with this file on a machine with a hard drive or a machine booting from Flash. It does not currently support kernel images larger than 450 Kbytes.

2.5 (TARGET) /etc/rc.d/init.d/

This directory contains the start-up files for various services. S20network is the most important, since it controls the IP addresses for the network system. S90thttpd controls the web server source location.

2.6 (TARGET) /home/httpd/

This directory is the default location for HTML files. Files can be added or changed in order to change the default web page's appearance. The default location of the HTML files can be changed by editing S90thttpd, as mentioned above.

Trademarks

© 2001 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, combinations thereof, AMDebug, and Élan are trademarks of Advanced Micro Devices, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.