Linux for the MIPS® Malta™ Development Platform User's Guide

Document Number: MD00646
Revision 01.03
October 30, 2008

MIPS Technologies, Inc.
955 East Arques Avenue
Sunnyvale, CA 94085-4521

Contents

1 Introduction

This document describes how to download, install, and boot the distribution of the Linux kernel and toolchain for the MIPS® Malta™ Development Platform. This version of the Linux operating system is customized to support the development of embedded applications for MIPS Technologies' 4KEc® and 4KSd™ CPU cores, and the 24K®, 24KE™, 34K® and 74K™ core families which support Release 2 of the MIPS32® Instruction Set Architecture (ISA). Release 1 processors are now supported, although so far only the 4Kc® processor has been tested.

2 Distribution Contents

Linux distribution deliverables are available on MIPS Technologies' public website as a downloadable tar archive and on the hard disk supplied with the Malta board. The distribution includes the following:

1. Linux kernel, based on MIPS Technologies' stable 2.6.23 version, which contains support for all current MIPS cores, Instruction Set Architectures (ISAs), and Application Specific Extensions (ASEs). The complete source code for the kernel is included in order to simplify the reconfiguration and rebuilding of the kernel for the MALTA Development Platform if required.

2. Cross-development tools precompiled to run on the host machine and produce executables for use on the Malta. The tools are based on CodeSourcery G++ toolchain, which is derived from gcc-4.2, binutils-2.17.50, and includes support for all MIPS ISAs and ASEs. Both both big and little endian targets are supported.

3. A root filesystem (RFS) for the target board, consisting of precompiled development packages and utilities for the MIPS target architecture.

3 Host System Requirements

The recommended host system configuration for installation is:

- an Intel Pentium or compatible processor
- 1 GByte RAM or more
- 40 - 60 GBytes available disk space
- A recent distribution of Linux. The `libxml2` Python bindings, in the package `python-libxml2`, must be installed at the same time that Python is installed.

4 Installation

This section describes the steps required to download and install the toolchain and kernel on your host machine.

To begin, first create and then move to the directory that will contain the downloaded files. For example:

```
$ cd ~
~$ mkdir CodeSourcery
~$ cd CodeSourcery
```

Now set up an ftp connection to MIPS Technologies' website:

```
mipstech$ ftp ftp.mips.com
```

and use "anonymous" in response to the name prompt from the ftp server:

```
ftp> anonymous
```

Use your email address as the password:

```
ftp> <your.email.address>
```

Move to the directory containing the files (the directory contents are invisible):

```
ftp> cd outgoing
```

Enable binary transfer mode:

```
ftp> binary                    -
```

Now you can choose between the little endian version:

```
ftp> get mips-mipsisa32r2el-malta-fc5s-25-install.tgz
```

or the big endian version of the toolchain:

```
ftp> get mips-mipsisa32r2-malta-fc5s-29-install.tgz
```

and then get the Linux kernel:

```
ftp> get kernel_2.6.23.tar.bz2
ftp> bye
```

Untar the toolchain tarball:

```
mipstech$ tar xzvf mips-mipsisa32r2el-malta-fc5s-25-install.tgz
```

Move to the toolchain directory:

```
mipstech$ cd mips-mipsisa32r2el-malta-fc5s-25-install
```

In supervisor mode, install the toolchain:

```
mips-mipsisa32r2el-malta-fc5s-25-install$ sudo ./install
```

Press <space bar> enough times and enter *Y* in the agreement.

Now follow the directions until the components to be installed are displayed.

Note: A known issue in the installation is that you must not "relocate" any of the components. The default location is `/opt/timesys/toolchain...` The toolchain will not function correctly if this location is changed.

Include the path to the toolchain:

```
mips-mipsisa32r2el-malta-fc5s-25-install$ PATH=/opt/timesys/toolchain:$PATH
mips-mipsisa32r2el-malta-fc5s-25-install$ cd ..
```

Untar the kernel:

```
mipstech$ tar xzvf kernel_2.6.23.tar.bz2
mipstech$ cd kernel_2.6.23
```

Config files are contained in the `arch/mips/configs` directory. Of particular interest are the files `malta_aprpconfig`, `malta_smtcconfig`, `malta_smvpconfig`, and `malta_upconfig`. For this example, `malta_upconfig` is used.

Copy the appropriate config file:

```
kernel_2.6.23$ cp arch/mips/configs/malta_upconfig .config
```

You can modify configuration settings by opening menuconfig:

```
kernel_2.6.23$ make menuconfig
```

When finished making your modifications, exit and then build the kernel:

```
kernel_2.6.23$ make CROSS_COMPILE=mipsisa32r2el-linux-
```

When completed, the kernel file (`vmlinux.srec`) will be located in the directory `arch/mips/boot`.

5 Booting Linux Kernel on Malta

The Linux kernel can be booted on the Malta board using the root file system on the Malta hard disk, using the root file system with NFS, or by cloning the Malta root file system.

5.1 With Root File System on Hard Disk

Make the kernel available to tftp:

```
kernel_2.6.23$ cp arch/mips/boot/vmlinux.srec /tftpboot/vmlinux_up_el.srec
```

Check if the `tftp-server` and `xinetd` are installed:

```
kernel_2.6.23$ rpm -q tftp-server
kernel_2.6.23$ rpm -q xinetd
```

If they are not installed, install and then enable them:

```
kernel_2.6.23$ sudo /sbin/chkconfig tftp on # enable tftp server
kernel_2.6.23$ sudo /etc/init.d/xinetd restart # make xinetd aware of tftp server
```

Make sure the endian switch setting (S5) on the Malta is consistent with the endianess of the kernel to be used. Then restart the Malta board and enter the following command at the Yamon prompt:

```
YAMON> load tftp://<hostIPaddress>/vmlinux_up_el.srec
```

Transfer control to the kernel:

```
YAMON> go . root=/dev/hda3 # /dev/hda3 is little endian file system
```

In the above command, if the kernel was built using big endian, "`root=/dev/hda3`" is replaced by "`root=/dev/hda2`".

After the kernel is loaded, a prompt is displayed. At this point, there is no password for the root account.

5.2 With Root File System Over 'NFS'

First add the directory to be exported by opening the exports file:

```
$ sudo <editor> /etc/exports
```

and adding the following line:

```
/opt/timesys/linux/6.2/mipsisa32r2el-std/rfs *(no_root_squash,anongid=0,rw,sync)
```

The above command assumes the RFS is in `/opt/timesys/linux/6.2/mipsisa32r2el-std/rfs`.

Export all directories in `/etc/exports`:

```
$ sudo /usr/sbin/exportfs –a
```

Restart NFS:

```
$ sudo /etc/init.d/nfs restart
```

Assign the correct values to the variables:

```
YAMON> set bootserver 192.168.20.194
YAMON> set ipaddr <Malta board IP address>
YAMON> set gateway <gateway IP address>
YAMON> set subnetmask <subnet mask>
```

Load the Linux kernel using the following commands:

```
YAMON> set nfs1 "nfsroot=$bootserver:/opt/timesys/linux/6.2/"
YAMON> set nfs2 "mipsisa32r2el-std/rfs,timeo=20 "
YAMON> set nfs3 "ip=$ipaddr:$bootserver:$gateway:$subnetmask"

YAMON> go . $nfs1$nfs2 $nfs3
```

5.3 Cloning the MALTA Root File System

All Malta boards include a disk that contains a root file system (RFS) with 16 versions of the Linux kernels

- little/big endian
- uclibc/glibc

- aprp/smtc/smvp/up

To boot any of the available kernels, use the following Yamon commands:

```
YAMON> disk read hda 3f ff 800d0000
YAMON> go 800d0000
```

During the boot process, you will choose which kernel to boot and which partitions to be mounted.

To clone the root file system, first create on the host a gzipped tar file of the root file system.

```
$ cd /opt/timesys/linux/6.2/mipsisa32r2el-std/rfs
rfs$ sudo tar cvzf rfs.tar.gzip
```

Now on the Malta, login as root (no password at this time) and using the IP addresses used in the previous examples, mount the drive, move to that directory, and then do a secure copy of the gzipped tar file containing the root file system that we generated on the host to the current directory (`/mnt`).

```
# mount /dev/hda8 /mnt
# cd /mnt
mnt # scp 192.168.20.194:/opt/timesys/linux/6.2/mipsisa32r2el-std/rfs/rfs.tar.gzip .
mnt # tar xvzf rfs. tar.gzip
```

The last step untars the root file system on `/mnt`, effectively putting the new file system onto the `/dev/hda8` partition,

The above example assumes that the RFS was installed in: `/opt/timesys/linux/6.2/mipsisa32r2el-std/rfs`.

Note: You should never overwrite the partition from which the system was booted.

6 Toolchains

The host cross-toolchains are prefixed by the target names. For little endian, this is "`mipsisa32r2el-linux`", for example, “`mipsisa32r2el-linux-gcc`”. This name reflects the fact that both the native/target and cross/host compilers are configured to generate code which is, by default, compatible with either Release 1 or Release 2 of the MIPS32 ISA (i.e. `-march=mips32r1` or `-march=mips32r2`). You can instruct the compiler to generate code for other ISAs and cores.

By default the code generated by the compiler is optimized for the 4Kc/24K/34K pipeline. Normally this will provide excellent performance on all MIPS32 Release 1 or 2 cores; however, if you know that your code will run only on a specific CPU core, you may obtain more optimal code by specifying an explicit CPU type, for example, `-march=24k`, `-march=34k`, `-march=4kec`, `-march=4ksd` or `-march=74kc`. Note that this may enable the use of instructions which are not available, or are sub-optimal, on other cores.

To enable the `builtin` compiler intrinsics for the DSP ASE, available on the 24KE and 34K families, you must specify one of `-mdsp`, `-march=24ke`, or `-march=34k`.

To enable use of the SmartMIPS ASE, a feature of the 4KSd core, specify either `-msmartmips` or `-march=4ksd`.

Use of the multi-threading ASE is automatic when a program uses `pthread_create()` etc, so long as you are running the SMTC or SMVP versions of the kernel on an MT-capable core. Both VPEs will be used to execute threads within your program in parallel. No extra action is required by the programmer.

The non-standard `-mallow-branch-to-undefined` option may be used when building certain boot loaders which need this feature.

It is not possible to build and run applications on Linux that have been compiled using the MIPS16 ASE.

7 Linux Kernel Known Issues

No known issues.

8 Support

MIPS Technologies provides support for the Linux distribution through the following channels:

- WWW documentation pages at `http://www.mips.com`. This should be your first call when looking for the answer to any problems or queries you may have. There may be updated versions of the documents available on the site.
- Support for the Timesys release can be purchased from Timesys.
- Issues specific to the Malta Development Platform can be sent to support@mips.com.

9 References

1. MIPS Malta™ User's Manual
   MIPS Document: MD00048

2. Malta™ Developer's Kit Getting Started
   MIPS Document: MD00051

3. Yamon™ Users Manual
   MIPS Document MD00008

10 Revision History

Revision	Date	Description
01.00	October 30, 2008	Initial version.
01.01	XXX	Input from Chris R & Jaidev
01.02	XXX	Input from Hieu, Chrisr & Terrence
01.03	October 30, 2008	Input from Terrence & Chris D

Template: nW1.03, Built with tags: 2B