

RTLinux Installation Instructions

FSM Labs, Inc.*
<http://www.fsmlabs.com>

April 20, 2001

Abstract

This document is intended to guide the user through the installation steps needed to compile and install RTLinux from the Web.

Contents

1	Preparing for Installation	1
1.1	Downloading Linux Kernel	1
1.2	Checking Packages	3
2	RTLinux Installation	3
3	Post Installation and Running RTLinux Programs	6
4	Special Notes	7
5	Documentation and Sources of Help	7

1 Preparing for Installation

1.1 Downloading Linux Kernel

In order to compile the RTLinux kernel, you first need to download the kernel for which RTLinux was built. To do so, note that there are patches in the

*business@fsmlabs.com

top-level directory by the name `kernel_patch*`. For RTLinux 3.0-pre8, these files are:

- `kernel_patch-2.2`, and
- `kernel_patch-2.4`

Where:

- `kernel_patch-2.2` is for kernel 2.2.18 which can be found at <http://ftp.kernel.org/pub/linux/kernel/v2.2/linux-2.2.18.tar.gz>
- `kernel_patch-2.4` is for Linux kernel 2.4.0-test1 which can be found at <http://ftp.kernel.org/pub/linux/kernel/v2.4/test-kernels/linux-2.4.0-test1.tar.gz>

In general, to find out which Linux kernels are necessary for a given RTLinux distribution, simply type the following at the toplevel RTLinux source code directory:

```
grep -A 2 "^W*VERSION" kernel_patch*
```

For example, by typing the above command on an RTLinux 3.0pre8 release, I obtained the following output:

```
kernel_patch-2.2: VERSION = 2
kernel_patch-2.2- PATCHLEVEL = 2
kernel_patch-2.2- SUBLEVEL = 17
```

```
kernel_patch-2.4: VERSION = 2
kernel_patch-2.4- PATCHLEVEL = 4
kernel_patch-2.4- SUBLEVEL = 0
```

meaning that I have two options: I can either use the file `kernel_patch-2.2` on Linux kernel version 2.2.17, or `kernel_patch-2.4` on a Linux kernel version 2.4.0. Both of these Linux kernels can be obtained from <http://ftp.kernel.org/pub/linux/kernel>

Thus, it is up to you to choose which of the two Linux kernels would be most appropriate for your given hardware configuration.

1.2 Checking Packages

If you use kernels 2.2.x on x86 platforms, please make sure that you have gcc 2.7.2.3 or egcs-1.1.2 or egcs-2.91 installed. You can verify that with

```
gcc -v
```

On Debian, it is enough to install the "gcc272" package. On RedHat systems, one needs to install the "kgcc" RPM. You may be able to use other compiler versions, but this is not recommended.

2 RTLinux Installation

If you have downloaded the RTLinux distribution with a prepatched kernel, skip steps 1 and 2. Quick check: if your kernel contains file arch/i386/kernel/rtlinux.c, you do not need to patch the kernel.

1. put a fresh copy of the Linux kernel in the /usr/src/linux directory:
 - **cd /usr/src**
 - **tar xzf linux-2.2.18.tar.gz**
 - **cd linux**
2. If you haven't done so already, make sure that you remove old directories associated with older versions of RTLinux:
 - **rm -fr /usr/include/rtlinux**
 - **rm -fr /usr/rtlinux-***
 - **rm -fr /usr/rtlinux**
 - **rm -fr /usr/src/rtlinux**
3. If you haven't done so already, put a fresh copy of the RTLinux kernel in the /usr/src/rtlinux directory:
 - **cd /usr/src**
 - **tar xzf rtlinux.tar.gz**

4. Create a symbolic link from within the `rtlinux` directory to the `linux` directory:

- **`cd /usr/src/rtlinux`**
- **`ln -sf /usr/src/linux ./linux`**

5. Patch the kernel with the RTLinux patch:

- **`cd /usr/src/linux`**
- **`patch -p1 < /usr/src/rtlinux/kernel_patch-2.2`**

OR, if you're using a 2.4.xx kernel:

- **`patch -p1 < /usr/src/rtlinux/kernel_patch-2.4`**

6. Now, clean all old `.o` files and stale dependencies:

- **`make mrproper`**

7. Now, configure the Linux kernel:

- **`cd /usr/src/linux`**
- **`make config`**
OR **`make menuconfig`**
OR **`make xconfig`**

Enabling APM support is not recommended. APM BIOS calls may have unpredictable effect on real-time performance.

On Alpha machines, you need to enable RTLinux Support (`CONFIG_RT LINUX`). On i386 and PPC, this is done automatically.

Please make sure to specify the correct CPU type for the target machine.

8. After you are finished configuring the Linux kernel, type:

- **`make dep`**

Steps 9 through 11 are x86-specific.

9. Compile the Linux kernel and modules:

- **make bzImage**
- **make modules**

Install the Linux modules:

- **make modules_install**
- **cp arch/i386/boot/bzImage /boot/rtzImage**

10. Configure LILO. To do so, edit `/etc/lilo.conf` to contain the following piece (you only need to do this once):

```
image=/boot/rtzImage
    label=rtlinux
    read-only
    root=/dev/hda1
```

WARNING: replace `root=/dev/hda1` in the above with your root filesystem. The easiest way to find out which filesystem it should be, take a look at the existing entry in your `/etc/lilo.conf` for `"root="`. Alternatively, type `"df"`, and look for the line for `"/"` in the `"mounted on"` column. The corresponding entry in the `"Filesystem"` column is your root filesystem.

11. Install LILO. To do so, type:

- **/sbin/lilo**

Restart the computer:

- **/sbin/shutdown -r now**

Load the RTLinux kernel: At the LILO: prompt, press `"Shift"` or `"Tab"`. This will give you a listing of the available kernels. Enter:

- **rtlinux**

RTLinux should boot.

12. Configure RTLinux:

- **cd /usr/src/rtlinux**
- **make config**
OR **make menuconfig**
OR **make xconfig**

13. Compile RTLinux:

- **make**
- **make devices**
- **make install**

The last step will create the directory:

```
/usr/rtlinux-xx (xx denotes the version)
```

which contains the default installation directory for RTLinux which is needed to create and compile user programs (that is, it contains the include files, utilities, and documentation). It will also create a symbolic link:

```
/usr/rtlinux
```

which points to `/usr/rtlinux-xx`. In order to maintain future compatibility, please make sure that all of your own RTLinux programs use `/usr/rtlinux` as its default path.

3 Post Installation and Running RTLinux Programs

To be able to run any programs, you must first load the `rtlinux` modules. To do so, type:

- **/usr/rtlinux/bin/rtlinux start < programname >**

where the optional `< programname >` is the name of the `rtlinux` program/module you want to run.

You can also try running the examples. To do so, simply go to the appropriate directory under `/usr/rtlinux/examples` and type:

- **make test**

For example:

- **cd /usr/rtlinux/examples/sound**
- **make test**

4 Special Notes

If you change any Linux kernel options, please don't forget to do:

- **cd /usr/src/rtlinux**
- **make clean**
- **make**
- **make install**

5 Documentation and Sources of Help

The `docs/html/GettingStarted` document contains a brief introduction to RTLinux. Additional documents in `docs/html` also provide information about other aspects to RTLinux such as web installation, CD installation, FAQ, and RTiC-Lab.

In case of problems, please consult the FAQ first, available in the `docs/` directory.

If all of the above fails, you can obtain help from your peers via the `rtl@rtlinux.org` mailing list for which you can un/subscribe to via `http://www.rtlinux.org/mailling_lists.html` .

FSM Labs further provides commercial support, development, and training. Please contact FSM Labs at

`business@fsm labs.com`

or visit their website at

`http://www.fsm labs.com`

for additional information.