

RTLinux Installation Guide

Prof. Chang-Gun Lee

1 Overall steps for RTLinux installation

1. RedHat Linux Installation

If you have not done it before, make a separate disk partition and install RedHat linux on the partition. RedHat 8.0 (Psyche) is recommended since it is most compatible with the RT-Linux we will use in this course. Also, make sure to install kernel development tools, gcc compiler, and utilities such as patch, depmod, make, and bzip2.

Note: The following explanation is based on RedHat 8.0 with Linux kernel 2.4.18 and rtlinux-3.2-pre1. If you do want to use RedHat 9.0, please replace "linux-2.4.18.tar.bz2 + rtlinux-3.2-pre1.tar.bz2" with "linux-2.4.22.tar.bz2 + rtlinux-3.2-pre3.tar.bz2".

2. Get linux kernel

Download "linux-2.4.18.tar.bz2" from the course website.

3. Get rtlinux kernel and patches

Download "rtlinux-3.2-pre1.tar.bz2" from the course website.

4. Put a fresh copy of the rtlinux kernel in /usr/src/rtlinux-3.2-pre1

Use the following commands.

```
cd /usr/src
```

```
tar xjf rtlinux-3.2-pre1.tar.bz2
```

This will create "rtlinux-3.2-pre1" directory under /usr/src.

5. Put a fresh copy of the linux kernel in /usr/src/rtlinux-3.2-pre1/linux

Use the following commands.

```
cd /usr/src/rtlinux-3.2-pre1
```

```
tar xjf linux-2.4.18.tar.bz2
```

```
ln -fs linux-2.4.18/ linux
```

This will create "linux" directory symbolic linked to "linux-2.4.18" under /usr/src/rtlinux-3.2-pre1.

6. Patch the linux kernel with the rtlinux patch

```
cd /usr/src/rtlinux-3.2-pre1/patches
```

```
bzip2 -d kernel_patch-2.4.18-rtl3.2-pre1.bz2
```

```
cd /usr/src/rtlinux-3.2-pre1/linux
```

```
patch -p1 < /usr/src/rtlinux-3.2-pre1/patches/kernel_patch-2.4.18
```

7. Clean all “.o” files and stale dependencies

```
cd /usr/src/rtlinux-3.2-pre1/linux
make mrproper
```

8. Configure the linux kernel

```
cd /usr/src/rtlinux-3.2-pre1/linux
make config /* text mode */ or
make xconfig /* X mode */
See Section 3 for details about linux configuration.
```

9. Build a new linux kernel

Use the following commands in order.

```
cd /usr/src/rtlinux-3.2-pre1/linux
make dep /* dependency refresh */
make bzImage /* compile linux kernel */
make modules /* compile modules */
su /* become the root to do the followings */
make modules_install /* Install the compiled linux modules */
cp arch/i386/boot/bzImage /boot/rtzImage
```

10. Configure your boot loader

In the following explanation, we assume that the root file system “/” is mapped to /dev/hda3 and the boot file system “/boot” is mapped to /dev/hda2. You can check out your mapping using “df” command. If you are using LILO as your boot manager, add the following lines to the file “/etc/lilo.conf”.

```
image = /boot/rtzImage
    label = rtlinux
    read-only
    root = /dev/hda3
```

The /dev/hda3 should be the device on which your root file system has been installed. Then, use the following command.

```
/sbin/lilo
```

For more details for your specific setting, see <http://www.freeos.com/articles/2701/> and <http://www.linuxheadquarters.com/howto/basic/lilo.shtml>.

If you are using GRUB as your boot manager, add the following lines to the file “/etc/grub.conf”.

```
title rtlinux
    root (hd0, 1)
    kernel /rtzImage ro root = /dev/hda3
```

The /dev/hda3 should be the device on which your root file system has been installed. (hd0, 1) corresponds to the first (0) physical disk derive’s second (1) partition, which is /dev/hda2.

It should be the partition where the boot file system “/boot” resides. For more details for your specific setting, see <http://www.gnu.org/software/grub/>.

11. Reboot and select rlinux from boot image options

RTLinux should boot.

12. Configure RTLinux

```
cd /usr/src/rtlinux-3.2-pre1
make xconfig /* accept default - rtsock can be installed later with the one in rtlinux-3.2-pre3
*/
```

13. Compile RTLinux Use the following commands in order.

```
cd /usr/src/rtlinux-3.2-pre1
make dep /* dependency refresh */
make
su /* become the root to do the followings */
make devices
make install
```

14. Run regression test

```
./scripts/regression.sh
You should get [OK] for all tests.
```

2 Run RTLinux applications

First of all, you should be the **root** to do the followings.

In order to run rlinux applications given in “examples” directory, you first insert dynamic rlinux kernel modules like mbuf, rtl_fifo, rtl, rtl_posixio, rtl_sched, and rtl_time. You can insert each module with “insmod modulename.o”. Fortunately, the script “rtlinux” can do this for you at a single step. Try

```
rtlinux status /* look at which module has been loaded */
rtlinux start /* insert all rlinux modules */
rtlinux status /* check whether all rlinux modules are loaded successfully*/
```

If you can see all the modules loaded, you are done. Otherwise, you might see error messages while you were doing “rtlinux start”. This may be due to incorrect configuration of linux kernel. See Section 3 and change settings. Then, try to build linux kernel again - repeat from step 7.

If you were successful in inserting all rlinux modules, go to examples/hello directory and make “hello.o” file by typing make. Then, insert “hello.o” by

```
insmod hello.o .
```

Then, try

```
lsmod.
```

to see hello.o has been successfully inserted.

In order to see the messages from the hello module, you should do this in text mode. If you are in X mode, you can check the output message from hello.o by typing

```
dmesg.
```

Finally, you can remove the hello module by typing

```
rmmod hello.
```

Note that “.o” should not be used in rmmmod.

3 Notes on Linux Configuration

This is the most difficult part in making RTLinux run successfully. The following items should be taken care of when configuring linux with “make config” or “make xconfig”. Regard the followings just as suggestions. The setting may be different for your computer. So, the first try is to change only the following settings keeping others as DEFAULT. If it works (which means you can run the hello module in Section 2), you are very lucky. Even if it does not work at the first time, don't be depressed. Be patient and try several different settings if you were not successful with the following settings.

1. Code maturity level options: check YES
2. Loadable module support: check all of 1) Enable loadable module support, 2) set version info on all system module, and 3) kernel module loader.
3. Processor type: select your processor type. Refer “/proc/cpuinfo” for your processor information.
4. Network device support: choose the proper device depending on your network device. If you have a 3COM card, in Ethernet (10 or 100Mbit), check 3COM cards and mark M for 3c590/3c900 series. If you have a Davicom network card, in Ethernet (10 or 100Mbit), check Davicom DM9102, and so on. You can check it out by ”System Settings/Network Configuration tool”.

5. Check kernel hacking and also check Magic SysRq Key.
6. For all others, keep the default as the first try. If it fails, change settings of suspicious options incrementally. Note that you can save your settings into a file and use it as default at the next time. For this, you can store your settings into “myconfig”. Then, next time you want to change some options, you can copy the “myconfig” to “.config”. “.config” is the file that “make config” or “make xconfig” uses as the default. Also, note that “make mrproper” in Step 7 will delete the “.config” file. So, you have to copy “myconfig” to “.config” after “make mrproper”.

Good Luck!

4 RTLinux Manual Page Installation

For developing RTLinux applications, we need to be familiarized with RTLinux system calls. However, it is almost not possible to memory everything. Fortunately, RTLinux provides manual pages for all system calls. It is a good idea to refer the manual pages whenever needed.

To install the manual pages,

1. Download “rtldoc-3.0.tar.bz2” from the course website.
2. Copy it to /usr/src/rtlinux-3.2-pre1/doc.
3. Use the following command

```
cd /usr/src/rtlinux-3.2-pre1/doc
tar xjvf rtldoc-3.0.tar.bz2
```

This should create /usr/src/rtlinux-3.2-pre1/doc/rtldoc-3.0 directory.

4. Add the directory path to the MANUAL PAGE search path.
 - Method 1: Add the following line to /etc/man.config:
MANPATH /usr/src/rtlinux-3.2-pre1/doc/rtldoc-3.0/man.
 - Method 2: Add the following line to .bashrc in your home directory:
export MANPATH=\$MANPATH:/usr/src/rtlinux-3.2-pre1/doc/rtldoc-3.0/man.
5. Try “man rtf_create” or “man 3 rtf_create”.