

# RTAI



Under construction. This page needs a bit of research on my part to complete. Please obtain example code from the [RTAI](#) website for the moment.

## Introduction

RTAI is a system that allows your Linux operating system to run so-called 'hard' realtime tasks. By 'hard' we mean that it isn't just 'best effort' it really will do something regularly, to a schedule dictated by you, no matter how heavily loaded the system. RTAI comes in two parts: a set of patches for your kernel and a library you can link your programs against which allows them to become realtime tasks. It may feel like you're running your program under Linux just like any other normal user-mode program, however under the hood you are taking full control of the hardware, and any other processes will only run as the remaining CPU time permits.



The above is a gross over-simplification of what is actually happening and is simply a lay-person's introduction. Please investigate the RTAI documentation for discussion of things like jitter, latency and so on

## Getting RTAI

You can download the latest version of RTAI from the website:

```
# wget https://www.rtai.org/userfiles/downloads/RTAI/rtai-5.1.tar.bz2
# tar xvf rtai-5.1.tar.bz2
# cd rtai-5.1
```

## Identifying the HAL patch

RTAI requires the application of kernel patches, but carries patches only for certain kernel versions. You can find out which ones like this:

```
# find ./ -name hal\*.patch
./base/arch/x86/patches/hal-linux-3.10.32-x86-8.patch
./base/arch/x86/patches/hal-linux-3.18.20-x86-6.patch
./base/arch/x86/patches/hal-linux-4.4.71-x86-10.patch
./base/arch/x86/patches/hal-linux-4.9.80-x86-4.patch
./base/arch/x86/patches/hal-linux-4.4.115-x86-10.patch
./base/arch/x86/patches/hal-linux-4.1.18-x86-9.patch
./base/arch/x86/patches/hal-linux-3.16.7-x86-5.patch
```

```
./base/arch/x86/patches/hal-linux-3.14.44-x86-12.patch  
./base/arch/x86/patches/hal-linux-4.9.51-x86-4.patch
```

Now check your own kernel version:

```
# uname -r  
4.4.14
```

We can see that the closest kernel with a HAL patch after this is 4.4.71, so that's the one to aim for. For now just copy the patch to the kernel source location:

```
# cp base/arch/x86/patches/hal-linux-4.4.71-x86-10.patch /usr/src
```

## Switching Kernel Version

We need to get ourselves on a kernel of the correct version (above) before we do anything else. Start by downloading and uncompressing the new version:

```
# cd /usr/src  
# wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.71.tar.xz  
# tar xvf linux-4.4.71.tar.xz  
# cd linux-4.4.71
```

Now use the existing configuration as a base and 'make oldconfig'. There may be some new config options that get prompted for, it's likely you can accept the defaults though.

```
# zcat /proc/config.gz > .config  
# make oldconfig
```

If you have a more obscure filesystem for your root partition you may now want to 'make menuconfig' to ensure it's compiled into your kernel [\*] instead of a module [m], or you may want to investigate using initrds. For anyone using the default EXT4, you're good to just make now:

```
# make
```

And then copy the results to install:

```
# make modules_install  
# cp arch/x86_64/boot/bzImage /boot/vmlinuz-rtai  
# cd /boot
```

Edit the lilo config:

```
# vi /etc/lilo.conf
```

And ensure there's a section added like this:

```
image = /boot/vmlinuz-rtai
```

```
root = /dev/sda1
label = Linux-4.4.71
read-only
```

Leave the other kernel image section in there as a second option in case something goes wrong with the boot, but you want to make the new one the default (the first entry).

## Patching the Kernel

When you're sure that new kernel works and your system boots and functions, now's the time to add the RTAI patches:

```
# cd /usr/src/linux-4.4.71
# patch -p1 < /usr/src/hal-linux-4.4.71-x86-10.patch
```

And now you can make the kernel again. This may take some time:

```
# make
```

If there are any prompted questions about the RTAI it's safe to leave them as defaults. Next install the new kernel.

```
# make modules_install
# cp arch/x86_64/boot/bzImage /boot/vmlinuz-rtai
```

You've already configured LILO, so just re-run it to reference the newly copied kernel:

```
# lilo
```

Reboot the machine to ensure that the new kernel works.

## Compiling/Installing RTAI

Before compiling RTAI itself we must ensure `/usr/src/linux` points to the new RTAI kernel.

```
# cd /usr/src
# rm linux
# ln -s linux-4.4.71 linux
```

Next just compile in the normal way:

```
# cd rtai-5.1
# ./configure
# make
# make install
```

At this point, some kernel modules should have arrived in `/usr/realtime/modules/`. For now you can load them manually, however in the future you can add them to `/etc/rc.d/rc.local`:

```
# /sbin/insmod /usr/realtime/modules/rtai_hal.ko
# /sbin/insmod /usr/realtime/modules/rtai_sched.ko
# /sbin/insmod /usr/realtime/modules/rtai_fifos.ko
```

Check they are there:

```
# lsmod | grep rtai
rtai_fifos          39941  0
rtai_sched         118855  1 rtai_fifos
rtai_hal           1762779  2 rtai_fifos,rtai_sched
```

Now we need to ensure we can link to the libraries:

```
vim /etc/ld.so.conf
```

Add the line:

```
/usr/realtime/lib
```

and then run (as root):

```
# ldconfig
```

Finally, we are ready to write our first RTAI program.

## Example Program

Here's an example Makefile to compile a really simple realtime program. It uses the `rtai-config` utility to set everything up:

```
CC = $(shell /usr/realtime/bin/rtai-config --cc)
LXRT_CFLAGS = $(shell /usr/realtime/bin/rtai-config --lxrt-cflags)
LXRT_LDFLAGS = $(shell /usr/realtime/bin/rtai-config --lxrt-ldflags)

all:: rtai-example

rtai-example: rtai-example.c
    $(CC) $(LXRT_CFLAGS) -o $@ $< $(LXRT_LDFLAGS) -llxrt
clean::
    $(RM) -f rtai-example *~
.PHONY: clean
```

And here is the demo program, `rtai-example.c`:

```
TODO
```

## Dedication

This page is dedicated to the memory of captain@captain.at whos pages have been a great source of information in kernel and realtime programming. You can still find his stuff in various archives, e.g. [here](#).

RIP Captain.

## Sources

\* Originally written by [User bifferos](#)

[howtos](#), [rtai](#), [realtime](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/es:howtos:software:rtai>

Last update: **2019/02/28 01:52 (UTC)**

