How-to Install RTAI in Ubuntu Hardy

Cristóvão Sousa
crisjss@gmail.com

v0.3 – October 2, 2008

Abstract

This *how-to* describes one way to install RTAI 3.6.1 on Ubuntu (or *Any*buntu) 8.04 (Hardy Heron) with Linux kernel 2.6.24.

Disclaimer

The author makes no representations or warranties with respect to the contents or use of this document. Use it at your own risk.

1 Introduction

This guide is intended to help installing RTAI in a Ubuntu based Linux system. As it specifically provides commands to this distribution and version, the process should be quite similar to other Debian based distros. This Ubuntu version comes with 2.6.24.x kernel version, so a 2.6.24 kernel should be used[1].

2 Preparation

- Download Linux kernel 2.6.24 (vanilla) – `linux-2.6.24.tar.bz2` @ `http://www.kernel.org`.
- Download RTAI 3.6.1 – `rtai-3.6.1.tar.bz2` @ `http://www.rtai.org`.
- Put this files into `/usr/src/`:

```
sudo cp linux-2.6.24.tar.bz2 /usr/src/
sudo cp rtai-3.6.1.tar.bz2 /usr/src/
```

- Unpack the files:

[1] Although there is no 2.6.24 kernel RTAI support for 64 bits, the 2.6.23 one can be used instead.

```
cd /usr/src/
sudo tar xfv linux-2.6.24.tar.bz2
sudo tar xfv rtai-3.6.1.tar.bz2
```

- Rename linux source directory to a more descriptive name (optional):

```
sudo mv linux-2.6.24 linux-build-2.6.24-rtai-3.6.1
```

- Create symbolic links to the two new folders (you can choose another names if you want):

```
sudo ln -snf linux-build-2.6.24-rtai-3.6.1 linux
sudo ln -snf rtai-3.6.1 rtai
```

- It is necessary to have installed all the packages associated with kernel compilation:

```
sudo apt-get install build-essential kernel-package ncurses-dev
```

3 Kernel Patching and Configuration

- Patch the kernel source with the correspondent RTAI patch:

```
cd /usr/src/linux
sudo patch -p1 -b < /usr/src/rtai/base/arch/x86/patches/hal-linux-2.6.24-x86-2.0-07.patch
```

- In order to make the new kernel configuration the most similar to the already installed one you can do the two next steps:

```
sudo cp /boot/config-`uname -r` ./.config
sudo make oldconfig   # then press Enter to all the prompts
```

  Press Enter to all the prompts of the last command.

- Now is time to configure the new kernel:

```
sudo make menuconfig
```

  In the menu there are many parameters you can change if you know what you are doing. The changes needed by RTAI are:

  - `Enable loadable module support` > `Module versioning support` = no

- `Processor type and features` > `Interrupt pipeline` = yes
- `Processor type and features` > `Subarchitecture Type` = PC-compatible

Because power management is a "latency killer" you should turn it off:

- `Power management options` > `Power Management support` = no
- `Power management options` > `CPU Frequency scaling` > `CPU Frequency scaling` = no

To have a new kernel with a understable name you should choose a good suffix to it, per example `-rtai-3.6.1`:

- `General setup` > `Local version - append to kernel release` = `-rtai-3.6.1`

If you know your machine processor type you can choose the right one:

- `Processor type and features` > `Processor family` = choose yours

If the chosen processor family have a TCS (all above Pentium I) then RTAI will use it and will have more time precision and latency. If you don't know which processor family is, a safe choice is 'Pentium I' (for not Pentium I inferior CPUs).

It is possible use multiple cores on multicore machines:

- `Processor type and features` > `Multi core scheduling` = no/yes

If your machine has a single core processor uncheck this option. If your machine has multicore you can choose to use all or only one[2].

After all changes exit and say yes to save the configuration.

- Optionally you can make a backup for the new configuration file:

```
cd /usr/src/linux ; sudo cp .config /boot/config-2.6.24-rtai-3.6.1
```

4 Kernel Compilation

- First prepare to a clean compilation

```
sudo make-kpkg clean
```

- At this point the kernel source is ready to compilation. The next command will compile the kernel and generate deb packages:

```
sudo make-kpkg --initrd kernel_image kernel_headers
```

[2] In case you want to simulate a single core machine.

This can take up to one hour depending in the machine you are using. It also can consume huge disk space, up to 2 GB. You will see many warning messages, don't worry. If this command ends with an error you should try to fix it and compile again (internet can be your friend).

- Two deb packages are generated in `/usr/src/`, kernel image and source headers. If you wish you can free the disk space used in the compilation with `sudo make-kpkg clean`.
- Go to packages directory:

```
cd /usr/src/
```

5 Kernel Installation

- Install the kernel image and source headers packages:

```
sudo dpkg -i linux-image-2.6.24-rtai-3.6.1_2.6.24-rtai-3.6.1-10.00.Custom_i386.deb
sudo dpkg -i linux-headers-2.6.24-rtai-3.6.1_2.6.24-rtai-3.6.1-10.00.Custom_i386.deb
```

  The package installation should create a new Grub entry.

- Due to a bug, the next commands need to be executed:

```
cd /usr/src/linux-headers-2.6.24-rtai-3.6.1/arch/x86/
sudo cp /usr/src/linux-headers-2.6.24-19/arch/x86/Makefile_32* ./
```

- Now, you must reboot. Choose the new RTAI patched kernel in Grub, at boot time.

6 RTAI Configuration and Installation

- Creating a build tree separated from the source tree of RTAI is advised by RTAI people, so do:

```
cd /usr/src/rtai
sudo mkdir build
cd build
```

- Configure and compile RTAI:

```
sudo make -f ../makefile menuconfig
```

  - `General` > `Linux source tree` = `/usr/src/linux-headers-2.6.24-rtai-3.6.1`

  - `Machine (x86)` > `Number of CPUs (SMP-only)` = the right number (if applicable)

  At the end exit and say yes to save the configuration. RTAI will be compiled.

- If there were no errors, install RTAI:

  ```
  sudo make install
  ```

- A backup of the RTAI devices files is needed. Do:

  ```
  sudo cp -a /dev/rtai_shm /lib/udev/devices/
  sudo cp -a /dev/rtf[0-9] /lib/udev/devices/
  ```

- Configure RTAI dynamic libraries to be wide available:

  ```
  sudo -s
  echo /usr/realtime/lib/ > /etc/ld.so.conf.d/rtai.conf
  exit
  sudo ldconfig
  ```

- The RTAI binaries directory can be added automatically to the `$PATH` variable. To do that, add the line

  `PATH="$PATH:/usr/realtime/bin"`

  to the end of `~/.profile` (user) and/or `/root/.profile` (root).

7 RTAI Test

If everything goes smooth in the above steps RTAI is readily installed. To be sure execute the latency test:

```
cd /usr/realtime/testsuite/kern/latency/
sudo ./run
```

Press `Ctrl-C` to stop.

8 Conclusion

That's all. Your machine is ready to real time. If some problem arise during the process you can try to solve it searching the internet for the error sentence.

References

[1] G. Racciu and P. Mantegazza, *RTAI 3.4 User Manual*, rev 0.3 ed., Oct. 2006. [Online]. Available: http://www.rtai.org

[2] J. Monteiro, *RTAI Installation Complete Guide*, Feb. 2008. [Online]. Available: https://www.rtai.org/RTAICONTRIB/RTAI_Installation_Guide.pdf