How to install RTAI in UBUNTU/KUBUNTU

Tested with: ***Kubuntu 6.10 | Linux kernel 2.6.17 | RTAI 3.4***

All operations are done in Super User mode. To be a SU in ubuntu system you must do

```
>sudo -s
```

Before you begin the installation of Real Time Aplication Interface for Linux you must make sure you have the following packages, build-essential, kernel-package, gcc, libncurses5, libncurses5-dev, libqt3-mt-dev. To install these packages you must have the ubuntu/kubuntu CD and a internet connection. Do this:

```
#apt-get update
#apt-get install build-essential
#apt-get install kernel-package
#apt-get install gcc
#apt-get install libncurses5
#apt-get install libncurses5-dev
#apt-get install libqt3-mt-dev
```

Unpack the kernel source:

```
# cd /usr/src
# tar -xvjf linux-2.6.17.tar.bz2
# ln -s linux-2.6.17 linux
```

Unpack or copy RTAI to /usr/src/rtai-4xxxx:

```
#tar -xvjf rtaixxxx
#ln rtai-4xxxx rtai
```

Patch the kernel:

```
#cd /usr/src/linux
#patch -p1 < ../rtai/base/arch/i386/patches/hal6c1-2.6.17.patch
```

Copy the existing kernel config file to /usr/src/linux

```
#cp /boot/config-2.6.xxxx /usr/src/linux/.config
```

Configure the kernel:

```
# make menuconfig
```

Make sure that:

- "Loadable module support -> Module versioning support" is disabled
- "Kernel hacking -> Compile the kernel with frame pointers" is disabled
- "Processor type and features -> Use reg. arguments" is disabled
- "Processor type and features -> Interrupt pipelining" is enabled

Before the compilation of kernel you have to edit the Makefile

```
#vim Makefile
```

In the Makefile you must add "-fno-common -fno-stack-protector" in CFLAGS

```
CFLAGS          := -Wall -Wundef -Wstrict-prototypes -Wno-trigraphs \
                   -fno-strict-aliasing -fno-common -fno-stack-protector
```

Compile the kernel:

```
#make
#make modules_install
#cp arch/i386/boot/bzimg /boot/bzimg-2.6.17rtai
#cp System.map /boot/System.map-2.6.17rtai
#cd /boot
#ln -s System.map-2.6.17rtai System.map
#cd /usr/src/linux
#mkinitramfs -o initrd-2.6.17rtai.img 2.6.17
```

After kernel is successful compiled and installed, make a new entry for grub:
#vim /boot/grub/menu.lst
In the file add this boot entry:

```
title           Real Time Aplication Interface for Linux, kernel 2.6.17-rtai (or any other name)
root            (hd0,1) (shoud be different on your system see the old kernel entry)
kernel          /boot/bzimg-2.6.17rtai root=/dev/sda2 (shoud be different on your system see the old kernel entry) ro quiet splash
initrd          /boot/initrd.img-2.6.17rtai
quiet
savedefault
boot
```

Reboot now and boot from the kernel you compiled.

#cd /urs/src/rtai
Before you compile RTAI you must do this:
    In file /usr/src/rtai/base/include/rtai_posix.h
    You must comment or delete the function:
        int pthread_condattr_getclock(pthread_condattr_t *condattr, clockid_t *clockid);
Now you can compile RTAI:
#make
#make install

Reboot now.

#cd /usr/realtime/testsuite/kern/latency
# ./run

It will occur an error “trap: 250: SIGINT: badtrap”

To resolve this problem
# cd /usr/realtime/bin
And change the first line of file rtai_load from #!/bin/sh to #!/bin/bash
Then when you run the latency program will occur another error: “error opening /dev/rtfs”
To resolve this problem create a file called “ldmod”with the next content:

```
#!/bin/bash

if test \! -c /dev/rtai_shm; then
        mknod -m 666 /dev/rtai_shm c 10 254
fi
for n in `seq 0 9`; do
        f=/dev/rtf$n
        if test \! -c $f; then
                mknod -m 666 $f c 150 $n
        fi
done

prefix=`rtai-config --prefix`
arch=`rtai-config --arch`
insmod=/sbin/insmod

if [ "$prefix" == "" ]; then
echo "ERROR: please set your PATH variable to <rtai-install>/bin"
exit
fi

MODULES=$prefix/modules

#sync paranoia might be useful

sync
if (`rtai-config --linux-version | grep -q 2.6`);
then

$insmod $MODULES/rtai_hal.ko IsolCpusMask=0;
sync
if [ "$arch" = "i386" ]; then
```

```
#$insmod $MODULES/rtai_lxrt.ko;
$insmod $MODULES/rtai_ksched.ko;
else
$insmod $MODULES/rtai_up.ko;
fi
sync
$insmod $MODULES/rtai_sem.ko;
sync
$insmod $MODULES/rtai_mbx.ko;
sync
$insmod $MODULES/rtai_msg.ko;
sync
$insmod $MODULES/rtai_fifos.ko;
sync
$insmod $MODULES/rtai_tbx.ko;
sync
$insmod $MODULES/rtai_bits.ko;
sync
$insmod $MODULES/rtai_mq.ko;
sync
$insmod $MODULES/rtai_shm.ko;

else

$insmod $MODULES/rtai_hal.o
sync
if [ "$arch" = "i386" ]; then
$insmod $MODULES/rtai_lxrt.o;
#$insmod $MODULES/rtai_up.o;
#$insmod $MODULES/rtai_smp.o;
#$insmod $MODULES/rtai_mup.o;
else
$insmod $MODULES/rtai_up.o;
fi
sync
$insmod $MODULES/rtai_sem.o;
sync
$insmod $MODULES/rtai_mbx.o;
sync
$insmod $MODULES/rtai_msg.o;
sync
$insmod $MODULES/rtai_fifos.o;
sync
$insmod $MODULES/rtai_tbx.o;
sync
$insmod $MODULES/rtai_bits.o;
sync
$insmod $MODULES/rtai_mq.o;
sync
$insmod $MODULES/rtai_shm.o;

fi
sync
```

NOTE: you must execute the file you created (ldmod) every time you run the kernel with RTAI.

You have to put the correct PATH to:

```
#export PATH=$PATH:/usr/realtime/bin
#./ldmod
```

You are on your own. I will not assume any responsibility if anything goes wrong. Good luck.

Hélder Cabrita | December 2006 © all rights reserved
e-mail: a27470@alunos.det.ua.pt