

How-to Install Xenomai in Ubuntu 10.04

Jorge Azevedo

jorge.amado.azevedo@gmail.com

v1.0 – September 8, 2012

Abstract

This *how-to* describes one way to install Xenomai 2.5.5.2 on Ubuntu (or *Anybuntu*) 10.04.3 (Lucid Lynx) with Linux kernel 2.6.32.

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 Xenomai in an Ubuntu based Linux system. As it specifically provides commands to this distribution and version, the process should be quite similar on other Debian based distros. This Ubuntu version comes with 2.6.32.x kernel version, so a 2.6.32 kernel should be used.

This guide is a variant of "How-to Install RTAI in Ubuntu Hardy" by Cristóvão Sousa, who was kind enough to release his original work under a Creative Commons license and send me his source files.

2 Xenomai Libraries

The first step in installing Xenomai is installing its libraries.

- All the necessary packages for compiling, building and installing the Xenomai libraries can be installed via the following commands

```
sudo apt-get install devscripts debhelper dh-kpatches
```

- Got to Downloads folder

```
cd ~/Downloads
```

- Download and untar Xenomai

```
wget -O - http://download.gna.org/xenomai/stable/xenomai-2.5.5.2.tar.bz2 | tar -jxf -
```

- Go to its folder

```
cd xenomai-2.5.5.2
```

- Now we're ready to build the packages. Debian packages need at least our personal information and a note about versioning. So first we define that

```
DEBEMAIL="your@email" DEBFULLNAME="Your Name" debchange -v 2.5.5.2 Release 2.5.5.2
```

- Now everything is in place, and all we have to do is build the packages

```
debuild -uc -us
```

- The resulting packages will be produced in the parent folder, so you can install them by doing

```
sudo dpkg -i ../*.deb
```

- Now we can check which kernel version are supported by our Xenomai version

```
ls -1 /usr/src/kernel-patches/diffs/xenomai
```

- Typically, the output will look something like this

```
adeos-ipipe-2.6.30-arm-1.15-03.patch.gz  
adeos-ipipe-2.6.31-arm-1.16-02.patch.gz  
adeos-ipipe-2.6.32.20-x86-2.7-03.patch.gz  
adeos-ipipe-2.6.33-arm-1.18-00.patch.gz  
adeos-ipipe-2.6.34.4-powerpc-2.10-05.patch.gz  
adeos-ipipe-2.6.34.5-x86-2.7-04.patch.gz  
adeos-ipipe-2.6.35.7-powerpc-2.11-02.patch.gz  
adeos-ipipe-2.6.35.7-x86-2.7-04.patch.gz
```

- Since our version of Ubuntu ships with kernel 2.6.32, we're gonna choose version 2.6.32.20. Adapt accordingly.

3 Kernel

- These packages will prepare your system for building a custom kernel package (these will take up almost 300mb of disk space, so consider yourself warned).

```
sudo apt-get build-dep --no-install-recommends linux-image-2.6.32-21-generic  
sudo apt-get install libncurses5-dev kernel-package
```

- Linux source code is usually located in `/usr/src`, so let's start by going there

```
cd /usr/src
```

- Download and untar kernel source for version 2.6.32.20

```
sudo wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.20.tar.bz2
```

```
sudo tar -jxf linux-2.6.32.20.tar.bz2
```

- Enter the source folder

```
cd linux-2.6.32.20
```

- Apply the patch

```
sudo /usr/src/kernel-patches/i386/apply/xenomai
```

- Now our ready for configuration. Instead of starting a configuration from scratch, our approach is to copy the original Ubuntu kernel `.config` to our folder and start configuration from there. This way we know for sure we have a kernel configuration that boots in our computer.

```
sudo cp ../linux-headers-2.6.32-33-generic/.config .config
```

- Since there's a slight mismatch in versions between the Ubuntu kernel from which we got the `.config` and our own kernel, we need to account for the possible difference in configuration options. We do this by running the following. It is usually safe to just press enter and leave the different configurations at their default values.

```
sudo make oldconfig
```

- We're now ready to configure our new Xenomai kernel.

```
sudo make menuconfig
```

- This is where the main difficulty of installing Xenomai resides. The Linux kernel has many different options and some of them are not compatible with Xenomai. Which options are problematic is something that varies between systems. The basic recommended setup is the following

```
– Processor type and features
  Processor family = choose yours
  [ ] Enable -fstack-protector buffer overflow detection
– Power management and ACPI options
  [ ] Cpu Frequency scaling
– - ACPI (Advanced Configuration and Power Interface) Support
  - < > Processor
```

- Processor family

Here you should choose your system's processor family. It is not recommended that you leave the default i586 option since, although it is generic, it may present problems to Xenomai. (ref: xenomai.org)

- Enable -fstack-protector buffer overflow detection

This is an Ubuntu kernel specificity. Without this option disabled, compilation will fail do to an incompatibility between our options and the Ubuntu default options.

- < > Processor

- Cpu Frequency scaling

These options try to make the processor run as predicatibily as possible. By deactivating the processor ACPI support, the processor isn't thrown into deep sleep states when inactive. By deactivating frequency scaling support, the clock frequency does not vary as a function of system load.

After all changes have been done, exit and say yes to save the configuration.

At this point the kernel source is ready for compilation. The next command will compile the kernel and generate debian packages. You should adjust concurrency level (number of paralell jobs) to your machine. Some suggest that it should be the number of cores plus one for maximum cpu usage. You should also adjust the `--append-to-version` parameter to fit your needs. In this case, the generated kernel version will be `2.6.32.20-xenomai-2.5.5.2`.

```
sudo CONCURRENCY_LEVEL=5 CLEAN_SOURCE=no fakeroot make-kpkg --initrd \
--append-to-version -xenomai-2.5.5.2 --revision 1.0 kernel_image kernel_headers
```

This can take from 30 minutes to 3 hours depending on the machine you are using. It can also consume up to 4 GB of disk space. You will see many warning messages, don't worry.

- Two deb packages are generated in `/usr/src/`, kernel image and source headers. Installation is as per usual

```
sudo dpkg -i ../*.deb
```

- Finally, we need to manually generate an initramfs or otherwise the system won't boot. This is a specificity of Ubuntu 10.04 and beyond. Previous versions of Ubuntu (and Debian) worked in a different manner.

```
sudo update-initramfs -c -k 2.6.32.20-xenomai-2.5.5.2 && sudo update-grub
```

4 Xenomai Test

If everything goes smoothly in the above steps Xenomai is installed. To test it, reboot the computer and choose the new Xenomai kernel. If it boots properly, then execute the latency test:

```
cd /usr/share/libxenomai-dev/examples/native/  
sudo make  
sudo ./trivial-periodic
```

Press **Ctrl-C** to stop. A typical output from a successful installation looks like

```
Time since last turn: 1000.145987ms  
Time since last turn: 1000.123452ms  
Time since last turn: 999.123545ms  
Time since last turn: 1000.123432ms  
Time since last turn: 999.12332ms  
Time since last turn: 999.87643ms
```

trivial-periodic is a simple application with a period of one second. On each activation, the time elapsed since the last activation is displayed on screen. In essence, what we see is the variation of the application's period. This is a gross estimation of the scheduling jitter, but it's a sufficient test to assert the validity of our installation. If there are wild variations on these values, something is wrong.

There's a small trick to allow non root user access to Xenomai. During installation, a user group called "xenomai" was added to the system with group id number 125 (you can check it in `/etc/group`). First we'll add ourselves to that group

```
sudo usermod -a -G xenomai username
```

Where "username" should be changed accordingly.

Now we need to pass the group id as a special boot parameter to the system. This way, all users belonging to the "xenomai" group will have access to real-time performance. The boot parameter is passed through grub, so we edit its default behavior

```
sudo gedit /etc/default/grub
```

And we edit the line that says

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

To

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash xeno_nucleus.xenomai_gid=125"
```

Then we update grub

```
sudo update-grub
```

On restart, trivial-periodic should now run without sudo.

If your kernel fails to boot or refuses to work in a normal, useful fashion, please refer to [for help](#). One common workaround is disabling MSI (under Bus Options) and other more advanced options.

5 Conclusion

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

6 Further reading

Here is a list of helpful documents for further help on Xenomai.

- [Xenomai- Building Debian Packages](#) (on which this guide was based) contains the same content but under a different lens.
- [Xenomai - Configuring x86 kernels](#) is the "go to" resource on kernel compilations. Some of the more obscure kernel options are mentioned in this document.
- [Xenomai FAQ](#) also contains further information on kernel configuration, under the section "Tips and tricks".