# POLITECHNIKA ŚWIĘTOKRZYSKA Wydział Elektrotechniki, Automatyki i Informatyki

Katedra Urządzeń Elektrycznych i Automatyki

Systemy operacyjne czasu rzeczywistego

## Instalacja systemu Xenomai

Instrukcja laboratoryjna

Paweł Strączyński

### 1 Instrukcja instalacji Xenomai 2.6.5 na Ubuntu 16.04

#### 1.1 Pobieranie plików źródłowych

Pracę zaczynamy od pobrania systemu Xenomai w wersji 2.6.5:

```
$ wget http://xenomai.org/downloads/xenomai/stable/xenomai-2.6.5.tar.bz2
$ tar xfvj xenomai-2.6.5.tar.bz2
```

Następnie pobieramy kernela (jądro linuksa) w wersji 3.18.20 - jest to najnowsza wersja kernela współpracująca z Xenomai'em 2.6.5

```
$ wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.18.20.tar.gz
$ tar xfv linux-3.18.20.tar.gz
```

#### 1.2 Przygotowanie kernela

Przechodzimy do instalcji pakietu potrzebnego do dalszerj pracy:

```
$ sudo apt install kernel-package
```

Następnie uruchamiamy skrypt do patchowania kernela.

```
$ cd linux-3.18.20
$ ../xenomai-2.6.5/scripts/prepare-kernel.sh
```

Wciśnięcie klawisza ENTER spowoduje wczytanie domyslnej konfiguracji. Kolejnym krokiem będzie konfiguracja jądra linuksa. Aby przejść do konfiguracji kernela w trybie graficznym należy skorzystać z komendy:

\$ make xconfig

Opcjonalnie można skorzystać z narzędzia do konfiguracji kernela bez 'GUI:

```
1 $ sudo apt install libncurses5-dev
2 $ make menuconfig
```

Poniżej przedstawiono kluczowe elementy konfiguracji jądra linuksa dla systemu Xenomai:

```
1 Recommended options:
2 
3 * General setup
4 --> Local version - append to kernel release: -xenomai-2.6.5
5 --> Timers subsystem
6 --> High Resolution Timer Support (Enable)
7 * Real-time sub-system
```

```
--> Xenomai (Enable)
8
    --> Nucleus (Enable)
9
         --> Pervasive real-time support in user-space (Enable)
10
  * Power management and ACPI options
11
    --> Run-time PM core functionality (Disable)
12
    --> ACPI (Advanced Configuration and Power Interface) Support
13
         --> Processor (Disable)
14
    --> CPU Frequency scaling
15
        --> CPU Frequency scaling (Disable)
16
    --> CPU idle
17
        --> CPU idle PM support (Disable)
18
19 * Pocessor type and features
    --> Processor family
20
         --> Core 2/newer Xeon
21
_{22}| * Power management and ACPI options
    --> Memory power savings
23
         --> Intel chipset idle memory power saving driver (Disable)
24
25 * Real-time sub-system
    --> Number of registry slots
26
         --> 4096
27
    --> Size of the system heap
^{28}
         --> 2048 Kb
29
    --> Size of the private stack pool
30
         --> 1024 Kb
31
    --> Size of private semaphores heap
32
         --> 48 Kb
33
    --> Size of global semaphores heap
34
         --> 48 Kb
35
```

#### 1.3 Kompilacja i instalacja jądra linuksa RT

Po wykonaniu konfiguracji jądra można przejść do jego kompilacji:

```
1 $ CONCURRENCY_LEVEL=$(nproc) make-kpkg --rootcmd fakeroot --initrd kernel_image
2 kernel_headers
```

Po pomyslnym przebiegu procesu kompilacji można przejść do instalacji kernela:

```
1 $ cd ..
2 $ sudo dpkg -i
3 linux-headers-3.18.20-xenomai-2.6.5_3.18.20-xenomai-2.6.5-10.00.Custom_amd64.deb
4 linux-image-3.18.20-xenomai-2.6.5_3.18.20-xenomai-2.6.5-10.00.Custom_amd64.deb
```

#### 1.4 Przygotowanie systemu

Po poprawnej instalacji jądra czasu rzeczywistego należy odpowiednio przygotować system operacyjny Linux do współpracy z systemem Xenomai. Przygotowanie należy zacząć od dodania użytkownika do grupy o odpowiednim poziomie uprawnień:

```
    sudo addgroup xenomai --gid 1234
    sudo addgroup root xenomai
    sudo usermod -a -G xenomai $USER
```

Kolejnym krokiem jest taka konfiguracja GRUB'a, aby uprzednio zainstalowane jądro linuksa uruchamiało się domyślnie. Ponadto konfigurujemu odpowiedni poziom uprawnień dla stworzonej wcześniej grupy użytkowników. Otwieramy plik konfiguracyjny dowolnym edytorem (np. nano) jak niżej:

```
$ sudo nano /etc/default/grub
```

A następnie dokonujemy konfiguracji:

```
1 GRUB_DEFAULT="Advanced_options_for_Ubuntu>Ubuntu,_with_Linux_3.18.20-xenomai-2.6.5"
2 #GRUB_DEFAULT=saved
3 #GRUB_SAVEDEFAULT=true
4 #GRUB_HIDDEN_TIMEOUT=0
5 #GRUB_HIDDEN_TIMEOUT_QUIET=true
6 GRUB_TIMEOUT=5
7 GRUB_CMDLINE_LINUX_DEFAULT="quiet_splash_xeno_nucleus.xenomai_gid=1234"
8 GRUB_CMDLINE_LINUX=""
```

Po skonfigurowaniu GRUB'a należy zrobić jego uaktualnienie i uruchomic system ponownie.

```
    $ sudo update-grub
    $ sudo reboot
```

#### 1.5 Instalacja Xenomai'a

Po poprawnym uruchomieniu wcześniej przygotowanego kernela można przejść do instalcji Xenomai'a:

```
1 $ cd xenomai-2.6.5/
2 $ ./configure
3 $ make -j$(nproc)
4 $ sudo make install
```

Warto rónież dodać do pliku .bashrc odpowiednie zmienne środowiskowe:

```
1 echo '
2 ####_Xenomai
3 export_XENOMAI_ROOT_DIR=/usr/xenomai
4 export_XENOMAI_PATH=/usr/xenomai
5 export_YATH=$PATH:$XENOMAI_PATH/bin
6 export_PKG_CONFIG_PATH=$PKG_CONFIG_PATH:$XENOMAI_PATH/lib/pkgconfig
7 export_LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$XENOMAI_PATH/lib
8 ' >> ~/.bashrc
```

Test instalacji można przeprowadzić za pomocą polecenia xeno latency.

## 2 Zadania do wykonania

- 1. Zainstalować system Xenomai zgodnie z instrukcją przedstawioną powyżej.
- 2. Zapoznać się z informacjami o zasobach systemu w katalogu /proc/xenomai.
- 3. Uruchomić program testowy wskazany przez prowadzącego, a następnie sprawdzić zawartość plików sched i stat.
- 4. Obciążyć działanie systemu przy pomocy polecenia **stress**. Sprawdzić jaki wpływ obciążenie systemu ma na zasoby systemu czasu rzeczywistego.
- 5. Za pomocą polecenia dmesg wyswietliść informację o utworzonych i usuniętych obiektach przez Xenomai. Jeśli pojawiają się błędy sprawdzić i przeanalizować ich kody.

```
/usr/include/asm-generic/errno-base.h
2/usr/include/asm-generic/errno.h
```