



Systemy operacyjne czasu rzeczywistego

Wprowadzenie / Instalacja Xenomai



WEAil



PSk

Paweł Strączyński

pstraczynski@tu.kielce.pl

Katedra Urządzeń Elektrycznych i Automatyki

System operacyjny czasu rzeczywistego

System operacyjny czasu rzeczywistego to taki w którym można określić **maksymalny** czas wykonania danej operacji.

W systemach operacyjnych czasu rzeczywistego (*real time operating system* – RTOS) poprawność procesu obliczeniowego zależy nie tylko od poprawności wyniku, ale również od czasu w którym został on osiągnięty.

Zastosowania systemu operacyjnego czasu rzeczywistego

- ☐ automatyka, sterowanie maszynami,
- ☐ robotyka,
- ☐ lotnictwo,
- ☐ aparatura medyczna,
- ☐ systemy bezpieczeństwa np. ABS, ESP itp.

Zadania systemu operacyjnego czasu rzeczywistego

Do zadań systemu operacyjnego czasu rzeczywistego należą m.in.:

- ☐ zarządzanie zasobami systemu,
- ☐ dostarcza mechanizmów do synchronizacji zadań i komunikacji pomiędzy nimi,
- ☐ szeregowanie zadań,
- ☐ przydzielanie zasobów poszczególnym procesom.

Podział systemów czasu rzeczywistego

- ❑ twarde (*hard real time*) - przekroczenie czasu powoduje katastrofalne skutki takie jak zagrożenie życia lub uszkodzenie urządzenia.
- ❑ miękkie (*soft real time*) – przekroczenie czasu powoduje negatywne skutki, a ich stopień jest zależny od czasu trwania opóźnienia,
- ❑ mocne (*firm real time*) – przekroczenie czasu nie skutkuje katastrofą ale powoduje nieprzydatność wyniku.

Systemy operacyjne czasu rzeczywistego

- ❑ RTLinux,
- ❑ QNX,
- ❑ RTAI,
- ❑ Xenomai,
- ❑ vxWorks,
- ❑ freeRTOS.

Rozszerzenia czasu rzeczywistego Linux'a

Linux może spełniać wymagania czasu rzeczywistego na dwa sposoby.

Pierwszym jest doprowadzenie do pełnej wywłaszczalności jądra poprzez zmiany w nim samym. Takie rozwiązanie nie wymaga dodatkowego API, programy to zwykle procesy linuksa.

Takie rozwiązanie jest możliwe dzięki zastosowaniu łaty **RT_Preempt patch**

Drugi dotyczy modelu dodatkowego mikrojądra czasu rzeczywistego i wykonywanie jądra linuksa jako jednego z zadań mikrojądra.



Rozszerzenia czasu rzeczywistego Linux'a oparte o mikrojądro

- ❑ RTLinux - opracowany przez New Mexico Tech i obecnie utrzymywany przez Wind River Systems (komercyjny) – powstał w 1996 roku,
- ❑ RTAI (*Real-Time Application Interface*) - opracowany przez Politechnikę Mediolańską w 2000 roku
- ❑ Xenomai - wywodzący się z RTAI Linux, powstaje w roku 2001 i wprowadza tzw. *skins* (skórki) umożliwiające przenośność.

Historia systemu Xenomai

- ❑ Projekt Xenomai został uruchomiony w sierpniu 2001 roku.
- ❑ W roku 2003 połączył się z RTAI (Real-Time Application Interface) w celu uzyskania produktu czasu rzeczywistego opartego o platformy darmowe dla systemu GNU/Linux o nazwie RTAI/Fusion.
- ❑ W roku 2005 stał się niezależnym projektem o nazwie Xenomai.

System Xenomai

Xenomai jest oprogramowaniem obejmującym biblioteki czasu rzeczywistego współpracującym z jądrem Linux'a. Sterowniki te umożliwiają wykorzystanie twardych ograniczeń czasowych dla aplikacji użytkownika.

Xenomai bazuje na abstrakcyjnym **jądrze RTOS**, które nadaje się do budowy wszelakiego rodzaju interfejsów czasu rzeczywistego, przez jądro które eksportuje zestaw ogólnych usług czasu rzeczywistego. Podstawowa struktura Xenomai'a jest podobna do RTAI, jednak Xenomai wprowadza lepszą integrację z Linux'em.

Xenomai opiera się o technologie ADEOS (*ang. Adaptive Domain Environment for Operating Systems*). Adeos jest nanokerneliem warstwy abstrakcyjnej sprzętu, operuje on pomiędzy sprzętem komputerowym a systemem operacyjnym na którym pracuje.

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Pobranie Xenomai'a

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

Pobranie jądra Linuksa

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

Rozpakowanie jądra Linuksa

```
tar xfv linux-3.18.20.tar.gz
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Pobranie narzędzi do konfiguracji kernela

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

Patchowanie jądra linuxa przy użyciu łaty Xenomai ipipe

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

Rozpakowanie jądra Linuxa

```
tar xfv linux-3.18.20.tar.gz
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Konfiguracja kernela

```
make xconfig
```

Konfigurowanie kernela bez GUI

```
sudo apt install libncurses5-dev  
make menuconfig
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

- * General setup

- > Local version - append to kernel release: -xenomai-2.6.5

- > Timers subsystem

- > High Resolution Timer Support (Enable)

- * Real-time sub-system

- > Xenomai (Enable)

- > Nucleus (Enable)

- > Pervasive real-time support in user-space (Enable)

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Power management and ACPI options

- > Run-time PM core functionality (Disable)
- > ACPI (Advanced Configuration and Power Interface) Support
 - > Processor (Disable)
- > CPU Frequency scaling
 - > CPU Frequency scaling (Disable)
- > CPU idle
 - > CPU idle PM support (Disable)

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

- * Processor type and features

- > Processor family

- > Core 2/newer Xeon (if `"cat /proc/cpuinfo | grep family"` returns 6, set as Generic otherwise)

- * Power management and ACPI options

- > Memory power savings

- > Intel chipset idle memory power saving driver (Disable)

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

```
* Real-time sub-system
--> Number of registry slots
    --> 4096
--> Size of the system heap
    --> 2048 Kb
--> Size of the private stack pool
    --> 1024 Kb
--> Size of private semaphores heap
    --> 48 Kb
--> Size of global semaphores heap
    --> 48 Kb
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Kompilacja kernela

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

Instalacja kernela

```
cd ..  
sudo dpkg -i linux-headers-3.18.20-xenomai-2.6.5_3.18.20-xenomai-2.6.5-10.00.Custom_amd64.deb linux-image-3.18.20-xenomai-2.6.5_3.18.20-xenomai-2.6.5-10.00.Custom_amd64.deb
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Konfiguracja uprawnień

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

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Konfiguracja GRUB'a

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

```
GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 3.18.20-xenomai-2.6.5"
#GRUB_DEFAULT=saved
#GRUB_SAVEDefault=true
#GRUB_HIDDEN_TIMEOUT=0
#GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=5
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash xeno_nucleus.xenomai_gid=1234"
GRUB_CMDLINE_LINUX=""
```

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Aktualizacja GRUB'a

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

Instalacja Xenomai'a

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

Instalacja systemu Xenomai 2.6.5 na Ubuntu 16.04

Aktualizacja bashrc

```
echo '  
#### Xenomai  
export XENOMAI_ROOT_DIR=/usr/xenomai  
export XENOMAI_PATH=/usr/xenomai  
export PATH=$PATH:$XENOMAI_PATH/bin  
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:$XENOMAI_PATH/lib/pkgconfig  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$XENOMAI_PATH/lib  
export OROCOS_TARGET=xenomai  
' >> ~/.bashrc
```