



**INNOWACYJNA
GOSPODARKA**
NARODOWA STRATEGIA SPÓJNOŚCI

Investujemy
w Waszą
przyszłość



INSTEPRO
Zintegrowane
Sterowanie
Produkcją

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ
ROZWOJU REGIONALNEGO



Raport wewnętrzny projektu InStePro

Nr. 3.1: Analiza i wybór narzędzi programistycznych wspomagających tworzenie systemu INSTEPRO

Data

30.09.2010

Przygotował zespół :

M. Klemiato
A. Piłat
W. Byrski

1. System operacyjny

Mając na uwadze wcześniejsze doświadczenia zespołu związane z budową systemów sterowania w czasie rzeczywistym, jako podstawowy system operacyjny przyjęto system QNX firmy Quantum Software. Istnieją dwie możliwości instalacji tego systemu:

- a) klasyczna instalacja na osobnej partycji dysku (QNX ma swój własny system plików)
- b) system wirtualny, uruchamiany w Windows za pomocą narzędzi firmy VMware (<http://www.vmware.com/pl/>). VMware Player jest darmowy, Workstation – płatny.

Sposób b) jest przez nas przetestowany i uważamy, że jest bardzo wygodny. Przy poprawnej instalacji działa bardzo szybko – wirtualizacji praktycznie się nie zauważa (w trybie tekstowym – konsola). Komunikacja z Windows jest możliwa za pomocą menedżera systemu plików SMB (*Server Message Block*). Usługa SMSfsys realizuje protokół SMB, wykorzystując NBT (NetBIOS przez TCP/IP). Po uruchomieniu SMSfsys i podmontowaniu zasobu na serwerze (Windows), otrzymuje się dostęp do systemu plików serwera jako do struktury drzewa lokalnego systemu plików.

Problem z klawiaturą przy pracy z platformą Photon (nakładka graficzna na QNX) został rozwiązany poprzez odpowiednią modyfikację plików systemowych.

Oprogramowanie pracujące w systemie QNX będzie realizowało wszystkie zadania merytoryczne, w tym pracujące w reżimie czasu rzeczywistego.

2. Kompilator C/C++

QNX 4.25 posiada wbudowany kompilator GCC. Nie ma jednak środowiska programistycznego. Jest co prawda Photon Builder, ale służy jedynie do projektowania interfejsów graficznych (okienkowych). Można oczywiście wykorzystywać wbudowany edytor Vim, który ma całkiem duże możliwości, ale nie jest wygodny – działa w trybie tekstowym.

Przy założeniu, że programista będzie się trzymał standardu POSIX i nie będzie wykorzystywał funkcji specyficznych dla QNX, można równie dobrze korzystać z kompilatora pod Windows. Taki kod powinien się także kompilować w QNX. Oprogramowanie obsługi operatora, pracujące w systemie Windows Server daje możliwość wykorzystania nowoczesnych narzędzi do budowania interfejsów graficznych (GUI). Przy takim podejściu, można wykorzystać nast. kompilatory:

- a) Visual Studio 2008 Prof – kombajn „do wszystkiego”, mocno przerośnięty niepotrzebnymi nam funkcjami, ale AGH ma wykupioną licencję
- b) Open Watcom (http://www.openwatcom.org/index.php/Main_Page) – kontynuacja kompilatora Watcom C/C++, darmowy dla zastosowań komercyjnych, posiada własne środowisko programistyczne (IDE)
- c) Dev-C++ (<http://www.bloodshed.net/devcpp.html>) – darmowy, na licencji GNU, oparty na GCC, posiada własne środowisko programistyczne (IDE)

Gdybyśmy zdecydowali się na najnowszą wersję QNX 6.4, to problem środowiska przestaje istnieć. QNX 6.4 ma wbudowany pakiet Momentics (<http://www.swdsoft.pl/files/pdf/Momentics.pdf>) – jest to zaawansowane środowisko programistyczne oparte na Eclips. Można nawet pracować w Windows, a dopiero wyniki kompilacji przesyłać do QNX.

3. System kontroli wersji

Wszelkie zmiany w tworzonym oprogramowaniu powinny być dokumentowane za pomocą któregoś z systemów kontroli wersji. Najpopularniejszy to CVS (*Concurrent Versions System*) udostępniany na licencji GPL. Obecnie nowszym i lepszym systemem jest SVN (*Subversion*), który ma zastąpić CVS, udostępniany na licencji Apache (<http://subversion.tigris.org>). Istnieje kilka darmowych klientów jak np. TortoiseSVN (<http://tortoisesvn.tigris.org>), integrujący się z powłoką Windows, czy VisualSVN (<http://www.visualsvn.com/visualsvn>), wtyczka do Visual Studio.

Aby istniał dostęp sieciowy do repozytorium plików, konieczne będzie zainstalowanie serwera SVN. Autorzy tego oprogramowania sugerują wykorzystanie serwera Apache lub svnservice. Ten drugi można

nawet zainstalować jako usługę w systemie Microsoft Windows (dysponujemy licencją AGH na Microsoft Server 2003).

4. Automatyczne tworzenie dokumentacji

Istnieje kilka systemów do tworzenia dokumentacji kodu źródłowego, jednym z bardziej popularnych jest Doxygen (<http://www.doxygen.org/>), stworzony dla wielu języków programowania, m.in. Python, C, C++, JAVA i PHP. Doxygen na podstawie odpowiednio sformatowanych komentarzy kodu źródłowego generuje dokumentację w następujących formatach: HTML, PDF, RTF, PostScript, HTML Help (format plików pomocy w środowisku Windows - *.chm), XML, Man pages (strony dla podręcznika man), LaTeX. Program dostępny jest na licencji GNU/LINUX, rozwijany jest w systemie Linux oraz MAC OS X, ale dostępny jest również na innych platformach UNIX'owych i w MS Windows.

Nakłady poniesione na odpowiednie formatowanie komentarzy szybko się zwracają w postaci eleganckiej i zunifikowanej dokumentacji, szczególnie w dużych projektach informatycznych.

W sieci istnieje sporo instrukcji dotyczących Doxygena, np. <http://wierzba.miks.uj.edu.pl/~gurgul/doxygen>.

5. Inne narzędzia wspomagające

Warto jeszcze rozważyć narzędzia do automatycznego formatowania kodu (tzw. *code beautifier*). Gwarantują one, że kody pisane przez różnych autorów będą formatowane w taki sam sposób. Ułatwia to czytelność kodów. Istnieje kilka narzędzi do tego celu, np. BCPP (<http://www.faqs.org/docs/Linux-HOWTO/C-C++Beautifier-HOWTO.html>), Uncrustify (<http://uncrustify.sourceforge.net>) czy Artistic Style (<http://astyle.sourceforge.net/astyle.html>).

Poza tym, do projektowania architektury systemu pomocne mogłyby być narzędzia do tworzenia diagramów przepływu, zależności, powiązań, etc. Do tego celu można wykorzystać język UML i zastosować np. pakiet Visio (AGH ma wykupioną licencję).