

SZYMAŃDA Jarosław M.¹

WIRTUALNY SYSTEM PROGRAMOWANIA ZDARZEŃ W LABORATORIACH DYDAKTYCZNYCH

W artykule przedstawiono propozycję włączenia systemu komputerowego do wykonywania zadań w rozproszonych sieciach transmisji danych. Ważnym elementem projektu było opracowanie aplikacji dedykowanych do komunikacji w sieciach LAN i WAN. Wprowadzono pojęcia określające podstawowe funkcje systemu i jego wdrożenia. Proponowany system może obsługiwać wirtualne laboratoria dydaktyczne z elementami kształcenia na odległość (e-learning).

THE VIRTUAL SYSTEM PROGRAMMING EVENT IN TEACHING LABORATORIES

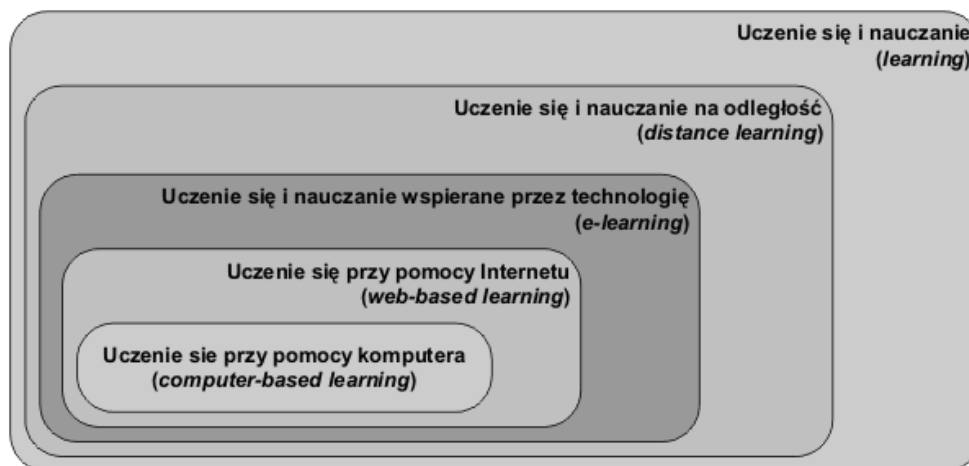
This paper presents a proposal for integrating a computer system to operate in the distributed data communication networks. An important element of the project was the implementation of dedicated applications for communication in LAN and WAN. The concept of defining the basic functions of the system and its implementation is introduced. The proposed system can support virtual teaching laboratories with elements of distance learning (e-learning).

1. WSTĘP

Podstawowym celem pracy było zaprojektowanie i opracowanie wielostanowiskowego systemu informatycznego, integrującego sieć rozproszonych serwerów (baz danych lub innych funkcjonalności dostępnych na wybranych terminalach), zlokalizowanych w sieciach komputerowych lokalnych (LAN) oraz Internecie (WAN). Istotnym elementem projektu było wykonanie dedykowanych wielobieżnych aplikacji umożliwiających przekazywania informacji (danych obliczeniowych, poleceń sterujących, obrazów wideo, itp.) pomiędzy poszczególnymi stacjami komputerowymi wchodzącymi w skład systemu, który mógłby również stanowić element większej i otwartej architektury realizującej zadania w czasie rzeczywistym [1,5,6,9].

Ważnym uzupełniającym założeniem opracowania było zachowanie kompatybilności z wykorzystywanymi na Wydziale Elektrycznym Politechniki Wrocławskiej systemami wspomagającymi procesy dydaktyczne w zakresie „kształcenia na odległość”; gwarantującym możliwość późniejszego zintegrowania obu tych rozwiązań [7,8,10].

¹Politechnika Wroclawska, Wydział Elektryczny, 50-370 Wrocław, Wybrzeże Wyspiańskiego 27
e-mail: jaroslaw.szymanda@pwr.wroc.pl



Rys.1. Umiejscowienie e-learningu w przestrzeni edukacyjnej (oprac. na podstawie danych Corporate E-learning, raport firmy WR Hambrecht and Co, marzec 2000) [7]

2. WYBRANE ELEMENTY KSZTAŁCENIA NA ODLEGŁOŚĆ

Pod pojęciem kształcenia na odległość (*e-learning*) przyjmuje się wszelkie działania wspierające proces szkoleń i edukacji, wykorzystujące technologie teleinformatyczne, komunikacyjne oraz multimedialne (Rys.1). Wyraźne odwoływanie się do technologii informatycznych nie uzasadnia jednak utożsamiania tego obszaru nauczania z technikami informatycznymi. Technologie informatyczne oraz multimedialne pełnią wyłącznie funkcję wspomagającą, zapewniając możliwość dystrybucji danych oraz ich przetwarzania [7,8]. Wdrażanie „*e-learningu*” wymaga zaangażowania różnych osób o odmiennych zakresach odpowiedzialności oraz wiedzy. Spektrum działań w tym zakresie można podzielić na trzy części:

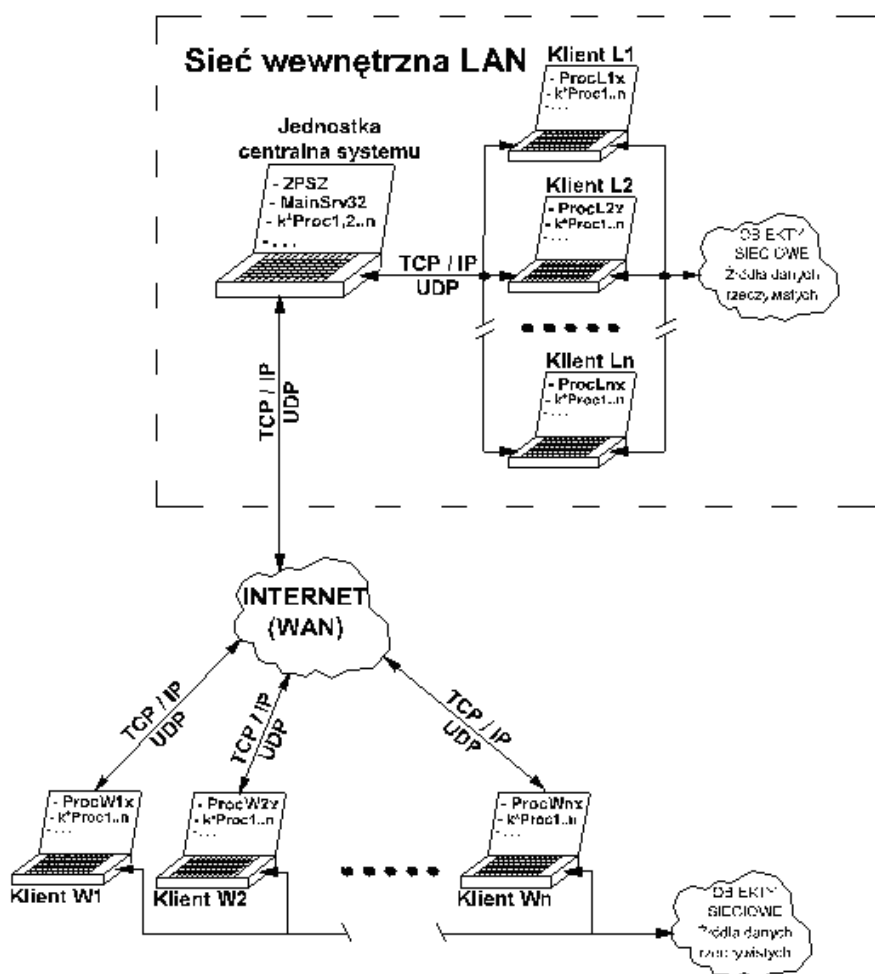
- technologię, w skład której wchodzi infrastruktura (sieć, serwery, stacje robocze, peryferia, terminale) oraz systemy informatyczne (oprogramowanie systemowe i użytkowe);
- treść edukacyjna (szkoleniowa), która obejmuje nie tylko kursy „*e-learningowe*”, ale także wszystkie materiały towarzyszące (rejstry uczestników, prowadzących, informacje, komunikaty itp.);
- usługi nauczania zdalnego, które obejmują wszelkie aktywności i działania związane z uruchomieniem, wsparciem oraz rozwojem procesów „*e-learningowych*” w organizacji (uczelni, wydziale, instytucie, zakładzie);

W kontekście prezentowanego projektu wirtualnego systemu programowania zdarzeń, w szczególności sposób proponuje się zintegrowanie postulatów kształcenia na odległość z zajęciami prowadzonymi w laboratoriach dydaktycznych. Powyższa integracja może stanowić swoistą synergię obu procesów, a także być przykładem kontynuacji dyskusji w zakresie określania warunków merytorycznych, organizacyjnych i prawnych uruchamiania innowacyjnych form dydaktycznych w polskich uczelniach [7].

3. ZAŁOŻENIA PROJEKTOWE

3.1 Środowisko sprzętowe i programowe

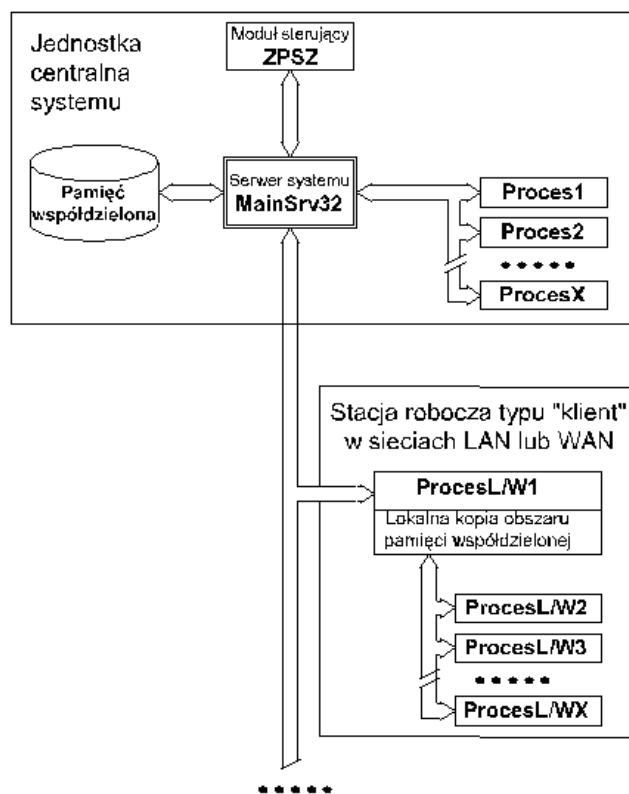
System opracowywany dla ściśle określonego środowiska sprzętowego i programowego, uwzględniający wyposażenie laboratoriów dydaktycznych oraz istniejącą infrastrukturę sieci teleinformatycznej.



Rys.2. Ogólna struktura systemu

Na przedstawionych schematach przepływu informacji (Rys. 2 i 3) wyróżnić można następujące komponenty i funkcjonalności:

- w warstwie sprzętowej: jednostkę centralną systemu, stacje klientów lokalnego (LAN – *Klient-Ln*) i sieciowego (WAN – *Klient-Wm*);
- w warstwie programowej: warstwę protokołów transmisyjnych (*TCP/IP*, *UDP*), oprogramowanie zarządzające jednostki centralnej (*MainSrv32*, *ZPSZ-Zintegrowany Pakiet Systemu Zarządzania*), oprogramowanie użytkowe stacji klienckich i jednostki centralnej (*ProcesLn*, *ProcesWm*);
- współdzielony obszar pamięci jednostki centralnej przeznaczony do wymiany informacji pomiędzy procesami. Obszar ten zarządzany jest bezpośrednio przez proces *MainSrv32*;
- wszystkie procesy uruchomione w jednostce centralnej współpracują bezpośrednio z współdzielonym obszarem pamięci; procesy uruchomione na stacjach klienckich współpracują z lokalnymi kopiami tego obszaru (dynamiczne buforowanie);
- na każdej stacji roboczej typu *klient* pracuje jeden proces, który jest odpowiedzialny za aktualizację lokalnej kopii współdzielonego obszaru pamięci. Wszystkie inne procesy uruchomione na tej stacji współpracują z kopią lokalną;
- każdy proces realizuje pewien zakres konkretnych zleceń zależnych od możliwości stacji komputerowej, na której jest uruchomiony;
- procesy realizujące ten sam zakres zleceń tworzą *Typ* identyfikowany jako klasa procesu;
- procesy identyfikowane są w systemie na podstawie unikalnego numeru identyfikacyjnego zawierającego w sobie m. in. numer IP, nazwę typu procedury oraz kolejny numer wywołania;
- wymiana informacji pomiędzy procesami odbywa się poprzez współdzielony obszar pamięci zainstalowany na jednym z komputerów, pełniącym rolę nadrzędną w stosunku do pozostałych;
- każdy proces po uruchomieniu „rejestruje się”, zgłaszając swój identyfikator do współdzielonego obszaru pamięci, przez który może wymieniać informacje z pozostałymi procesami;
- proces, który jest beczynny, jeśli nie jest jedynym procesem uruchomionym na danej stacji komputerowej samoczynnie zostanie wyłączony;
- każdy proces może przekazać otrzymane zlecenie innemu procesowi tego samego typu, lub jeśli sam nie potrafi wykonać zlecenia - procesowi innego typu;
- każdy proces może uruchomić kolejny proces tego samego, własnego typu (tzw. klon) lub proces innego typu, jeśli ten jest wymagany dla realizacji funkcji, której dany proces nie może wykonać;
- za koordynację działań wszystkich procesów w systemie odpowiada moduł serwera *MainSrv32*;
- wprowadzono definiowanie parametrów globalnych oraz indywidualnych określających strategię wymiany informacji oraz powoływania i zwalniania procesów przetwarzających;
- programowanie aplikacji zrealizowano m.in. w pakiecie kompilatorów *DevC++*, *Delphi7* oraz *Embarcadero Technologies, Inc. @ RAD STUDIO 2010* dla systemów *Microsoft Windows* oraz *Linux SUSE*;



Rys.3. Diagram wymiany danych pomiędzy procesami

3.2 Protokoły komunikacyjne

W opracowanym modelu przetwarzania rozproszonego wykorzystano adresowanie internetowe w oparciu o protokół IP (*Internet Protocol address*) [2,3]. Rozwiązanie to daje gwarancję unikalności zestawiania indywidualnych identyfikatorów dla wszystkich procesów uruchamianych w systemie. Wszystkie istotne dane obliczeniowe, monitorujące i sterujące, ze względu na konieczność zachowania bezpieczeństwa, przygotowywane są i przesyłane poprzez zastosowanie protokołu TCP (*Transmission Control Protocol*). W projekcie protokoły TCP/IP wykorzystany jest do przesyłania danych pomiędzy poszczególnymi stacjami komputerowymi (precyzyjnie pomiędzy poszczególnymi procesami a bazą danych serwera *MainSrv32*). W przypadku przesyłu danych strumieniowych i/lub zapytań *rozgłoszeniowych* wykorzystano szybki protokół UDP (*User Datagram Protocol*). Podobnie jak IP jest to protokół bepołączeniowy, nie używa mechanizmów kontroli przepływu i retransmisji. Korzyścią płynącą z uproszczenia budowy

jest większa szybkość transmisji danych i brak dodatkowych zadań, którymi musi zajmować się terminal (*host*) posługujący się tym protokołem. (np. przesyłanie obrazu).

3.3 Moduł MainSrv32

Moduł *MainSrv32* stanowi podstawę komunikacji pomiędzy wszystkimi elementami systemu. Jest on *serwerem* systemu. Moduł ten może być uruchomiony na dowolnej stacji komputerowej (terminalu). Zgodnie z założeniami projektowymi, integralność całego systemu zachowana jest poprzez wprowadzenie automatycznego uruchamiania modułu poprzez dowolny proces podrzędny. Dla modułu *MainSrv32* w pamięci operacyjnej komputera rezerwowany jest obszar na wyłączność (*Exclusive Memory*). Obszar ten służy do wymiany informacji pomiędzy procesami podrzędnymi. Współdzielony obszar pamięci sformatowany jest jako odrębny typ danych rekordowych.

3.4 Moduł zarządzający

Moduł zarządzający ZPSZ został opracowany jako aplikacja wspomagająca diagnostykę działającego systemu. Moduł po uruchomieniu poszukuje w sieci lokalnej działającego serwera *MainSrv32*. Jeśli serwer nie zostanie odnaleziony, moduł ZPSZ uruchomi go samodzielnie na tej samej stacji komputerowej, na której sam pracuje. W przypadku odnalezienia serwera *MainSrv32*, automatycznie rejestruje go w bazie danych. Do istotnych funkcji modułu należy również administracyjne sterowanie i monitorowanie procesów działających w sieci. Informacje dotyczące stanu pracy procesów wyświetlane są np. w zakładce *MonitorProcesów* oraz wprowadzane do zintegrowanej bazy danych.

	Identyfikator	S	Polecenie	Odpowiedź
1	70804311.ADO.0000	0	Połączony z BD!	
2		0		
3		0		
4		0		
5		0		

Rys.4. Okno modułu ZPSZ – zakładka *MonitorProcesów*

W każdym rekordzie tabeli umieszczonej w oknie głównym, przedstawione są dane dotyczące poszczególnych procesów. Wszystkie dane występujące w poszczególnych polach (poza wartością pola *S*) są typu tekstowego; ich znaczenie zależy od sposobu interpretacji przez program. Poszczególne pola mają następujące znaczenie:

- Identyfikator - unikalny w całym systemie numer przydzielony każdemu z procesów. Numer ten w sposób jednoznaczny identyfikuje stację komputerową, na której uruchomiony został proces oraz jego klasa (*Typ*).

Identyfikator procesu składa się z następujących części:

xxxxxxx . aaa . nnnn

gdzie:

nnnn - numer kolejnej kopii procesu typu aaa, uruchomionej na stacji komputerowej o numerze IP: xxx...xxxxx. Jeśli procesów danego typu będzie n, będą one miały numery od 1...n;

aaa - nazwa typu procesu;

xxxxxxx - nr IP stacji komputerowej na której proces jest uruchomiony, wyświetlany jako liczba typu *longint* (po konwersji z formatu *dotpoint*);

● S - status procesu. Liczba określająca w jakim stanie pracy znajduje się w danym momencie proces. Status informuje między innymi o tym czy proces jest zajęty i nie może przyjąć zlecenia; czy udzielił odpowiedzi na zlecenie; czy jest „wolny” i może przyjąć zlecenie. W szczególności systemowy status procesu może przyjmować następujące wartości:

● ● 0 - proces jest zarejestrowany w tablicy bazy danych serwera *MainSrv32* lub rekord jest pusty (*brak procesu*). Jeżeli proces jest zarejestrowany to znaczy, że może przyjąć zlecenie od innego procesu. Jeżeli dodatkowo proces nie realizuje żadnego zlecenia to nastąpi jego wyłączenie po zadanym czasie bezczynności (*timeout*);

● ● 1 - zarezerwowane dla wewnętrznych funkcji systemu ;

● ● 2 - status ustawiany przez proces, który przesłał zlecenie do danego procesu. Oznacza, że zlecenie zostało wpisane w pole Polecenie rekordu procesu wykonującego, tablicy bazy danych serwera *MainSrv32*. Jeżeli po zadanym czasie Status nie zmieni wartości (zlecenie nie zostanie przyjęte przez proces) nastąpi samoczynne wystawienie statusu 6 (brak możliwości udzielenia odpowiedzi);

● ● 3 - status ustawiany przez proces realizujący zlecenie. Oznacza, że zlecenie jest obecnie w trakcie kompilacji. Jeżeli po zadanym czasie Status nie zmieni wartości (zlecenie nie zostało skompilowane) nastąpi samoczynne wystawienie Status=6 (brak możliwości udzielenia odpowiedzi);

● ● 4 - status ustawiany przez proces realizujący zlecenie. Oznacza, że zlecenie jest obecnie w trakcie realizacji przez proces wykonujący. Jeżeli po zadanym czasie Status nie zmieni wartości (zlecenie nie zostało zrealizowane) nastąpi samoczynne wystawienie Statusu=6 (brak możliwości udzielenia odpowiedzi);

● ● 5 - status ustawiany przez proces realizujący zlecenie. Oznacza, że zlecenie nie mogło zostać zrealizowane przez dany proces, gdyż ten aktualnie realizuje inne zlecenie, ale przekazał zlecenie innemu procesowi. W tym przypadku w polu Odpowiedź procesu umieszczona zostanie informacja dla procesu, który zlecenie przysłał, zawierająca treść zlecenia oraz numer rekordu w tablicy bazy danych serwera *MainSrv32* procesu, któremu zlecenie zostało przekazane. Status jest kasowany (zmieni wartość na 0) po odbiorze odpowiedzi przez proces, który wcześniej przesłał zlecenie, albo po zadanym czasie, przez proces, który przesłał zlecenie dalej;

● ● 6 - status ustawiany przez proces realizujący zlecenie. Oznacza, że zlecenie nie mogło zostać zrealizowane przez dany proces i nie ma możliwości udzielenia odpowiedzi. Status jest kasowany (zmieni wartość na 0) po zadanym czasie;

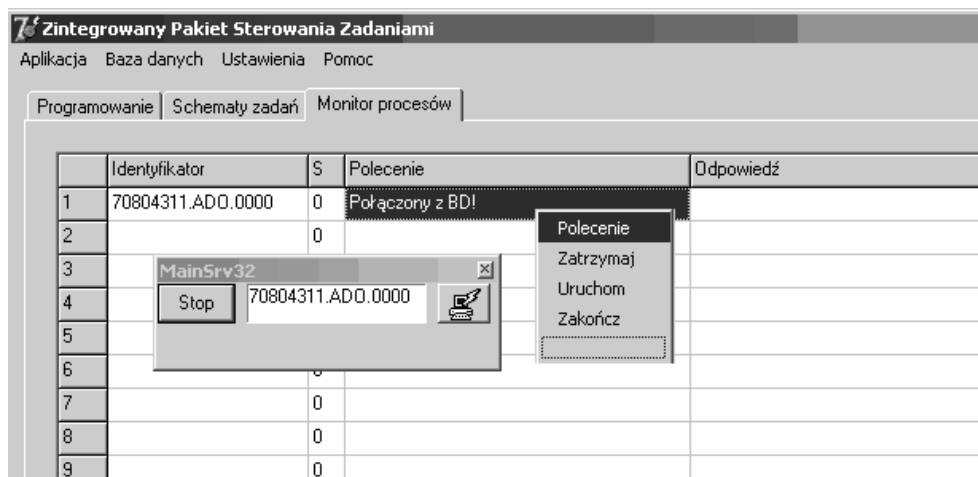
• • 7 - status ustawiany przez proces realizujący zlecenie. Oznacza, że zlecenie zostało zrealizowane. Jednocześnie w polu Odpowiedź procesu umieszczona zostanie odpowiedź na zlecenie oraz informacja dla którego procesu jest to odpowiedź. Status jest kasowany (zmieni wartość na 0) po zadany czasie lub po odebraniu odpowiedzi przez proces pytający;

• Polecenie - pole, w którym przekazywane są polecenia dla procesu. Polecenie może być przekazane do procesu przez inny proces lub ręcznie z wykorzystaniem odpowiedniej funkcji administracyjnej modułu ZPSZ;

• Odpowiedź - pole, w którym proces umieszcza odpowiedź na polecenia komunikacyjne otrzymane od innych procesów;

Menu (okno) z listą dostępnych funkcji uruchamiane jest po naciśnięciu prawego klawisza myszy. Funkcje dostępne w menu dotyczą aktualnie wybranego procesu.

Procesem wybranym jest proces zarejestrowany w podświetlonym rekordzie tabeli *MonitoraProcesów* - w przykładzie na rysunku nr 5 procesem wybranym jest moduł *MainSrv32* zarejestrowany w rekordzie nr 1 (klasa *ADO* ;nazwa własna w systemie ZPSZ).



Rys.5. Okno modułu ZPSZ z funkcjami do obsługi administracyjnej procesów

Do funkcji tych należą:

- Polecenie -funkcja administracyjnego przekazanie polecenia do procesu;
- Zatrzymaj -funkcja administracyjnego wstrzymania działania procesu;
- Uruchom -funkcja administracyjnego uruchomienia i zarejestrowania nowego procesu;
- Zakończ -funkcja administracyjnego zakończenia działania procesu.

Innym sposobem przekazania danych administracyjnych do procesu, umożliwiającym modyfikację dowolnego pola w rekordzie procesu, jest bezpośrednie wprowadzenie wartości do wybranego pola. Wybór pola polega na podwójnym „kliknięciu myszą” w dane pole, co spowoduje otwarcie okna dialogowego umożliwiającego aktualizację danej wartości.

Zintegrowany Pakiet Sterowania Zadaniem
 Aplikacja Baza danych Ustawienia Pomoc

Programowanie Schematy zadań **Monitor procesów**

	Identyfikator	S	Polecenie	Odpowiedź
1	70804311.ADO.0000	0	Połączony z BD!	
2	70804311.P01.0001	0		
3	70804311.P01.0002	0		
4	70804311.P01.0003	0		
5	70804311.P01.0004	0		
6		0		

Rys.6. Okno modułu ZPSZ – przykładowa konfiguracja po uruchomieniu kilku procesów

Zintegrowany Pakiet Sterowania Zadaniem
 Aplikacja Baza danych Ustawienia Pomoc

Programowanie Schematy zadań **Monitor procesów**

	Identyfikator	S	Polecenie	Odpowiedź
1	70804311.ADO.0000	0	Połączony z BD!	
2	70804311.P01.0001	0		
3	70804311.P01.0002	0		

Zadanie: 70804311.P01.0002

Identyfikator: 70804311.P01.0002

IP (Serwer): 87.99.56.4

IP (Zadanie): 87.99.56.4

TimeS0: 52 TimeS1: 0 TimeS2: 0 TimeS3: 0 TimeS4: 0 TimeS5: 0 TimeS6: 0 TimeS7: 0 TimeEx1: 0 TimeEx2: 0

Monitor pracy procesu: _____

Wewnętrzny bufor roboczy

Ind w ZPSZ	Status	Zlecenie	Odpowiedź	ProcZlecający
3	0			

Tabela realizacji zleceń Obszar roboczy

	Priorytet	Parametr	Idn. procesu wykonującego	Ind. z ZPSZ	W/Priorytet	W/Parametr
1	0			0	0	
2	0			0	0	
3	0			0	0	
4	0			0	0	
5	0			0	0	
6	0			0	0	
7	0			0	0	
8	0			0	0	
9	0			0	0	

Rys.7. Okno modułu ZPSZ – widok okna roboczego dla przykładowego procesu

Dla każdego poprawnie powołanego potomnego procesu można wyświetlić jego parametry pracy, niezależnie od aktualnie wykonywanych zdalnie procedur (Rys.7).

Znaczenie poszczególnych pól jest następujące:

- Identyfikator - identyfikator procesu za pomocą którego proces zarejestrował się w tablicy bazy danych serwera *MainSrv32*;
- IP(Serwer) - identyfikator IP stacji komputerowej, na której zainstalowany jest serwer systemu *MainSrv32* wraz z bazą danych procesów;
- IP(Zadanie)- identyfikator IP stacji komputerowej, na której uruchomione jest bieżące zadanie (o identyfikatorze w polu Identyfikator) ;
- Monitor pracy procesu - narzędziowe pole tekstowe, w którym wyświetlane są komunikaty dotyczące działań podejmowanych przez proces w trakcie jego pracy;
- TimeS0.....Ex2 - wartości poszczególnych *Timer'ów* kontrolujących pracę procesu.

4. WNIOSKI

W artykule przedstawiono wybrane elementy projektu systemu teletransmisji wykorzystującej dynamiczną strukturę przetwarzania rozproszonego w sieciach lokalnych LAN oraz Internecie (WAN). Wprowadzono pojęcia określające podstawową funkcjonalność systemu oraz warunki jego realizacji. Zaproponowano integrację systemu zarządzania z technikami wspomagającymi procesy dydaktyczne w zakresie „kształcenia na odległość” (e-learningu). Wersja testowa opracowanych aplikacji będzie wprowadzona do programu zajęć laboratoryjnych w ramach kursu „Sieci i systemy teleinformatyczne” w roku akademickim 2011/12.

5. BIBLIOGRAFIA

- [1] Boorstyn, R.R., Frank, H.: *Large-Scale Network Topological Optimization*: IEEE Trans.Commun.: Vol. COM-25: 1977: s.29-47.
- [2] Comer, D. E., Stevens, D. L.: *Sieci komputerowe TCP/IP 2 – Projektowanie i realizacja protokołów*: WNT: Warszawa 1997
- [3] Comer, D. E., Stevens, D. L.: *Sieci komputerowe TCP/IP 3 – Programowanie w trybie klient-serwer*: WNT: Warszawa 1997
- [4] Freeman, R.L.: *Practical Data Communications*: John Wiley & Sons: New York 1995
- [5] Gavish, B.: *Topological Design of Centralized Computer Networks- Formulations and Algorithms*: Networks: Vol 12: 1982: s.355-377
- [6] Gerla M., Kleinrock L.: *On the Topological Design of Distributed Computer Networks*: IEEE Trans.Commun.:Vol.COM-25:1977: s.48-60
- [7] Hyla M.: *Przewodnik po e-learningu*: Oficyna Ekonomiczna: Kraków 2005.
- [8] Szymańda, J.M., Rezmer J.: *Platforma zdalnej edukacji „Moodle” – projektowanie kursu komplementarnego*: Oficyna Wydawnicza Politechniki Wrocławskiej: Wrocław 2008: s.131-138.
- [9] Szymańda, J.M., Noszczyński, M.: *Modelowanie procedur przetwarzania rozproszonego w sieciach informatycznych laboratoriów dydaktycznych*: XIII ZKwE'08 Poznań: 2008: s.313-314.
- [10] Szymańda J.M.: *Dydaktyczne aspekty projektowania wirtualnego w kształceniu na odległość*: XIII ZKwE'08 Poznań, Kwiecień 14-16, 2008 s.361-362.