

# SPRAWOZDANIE

## Temat 7: *Komunikator internetowy typu GG*

Naszym zadaniem było stworzenie aplikacji sieciowej opartej na gniazdach BSD. Serwer zaimplementowaliśmy w języku C++ z obsługą gniazd w systemie Linux oraz Mac OS X, natomiast aplikację stworzyliśmy dla systemu Windows w języku C# (Visual Studio 2013).

Pliki źródłowe:

a) Serwer

main.cpp – zawierający główne funkcje programu (obsługa wątków, obsługa komunikatów od klienta (poza komunikacją na socketach)  
Communication.cpp, Communication.hpp – klasa Communication odpowiedzialna za odbieranie i wysyłanie danych przez socket.

b) Klient

Program.cs – klasa z funkcją main(), tworzymy serwer i wyświetlamy okno logowania

Serwer.cs – główna klasa odpowiedzialna za połączenie z serwerem, komunikacją z serwerem i wyświetlanie okien zależnych od komunikatów otrzymanych z serwerem (główna klasa aplikacji)

User.cs – klasa z informacjami o użytkowniku (wiadomości, kontakty, otwarte okna rozmów z użytkownikami)

MainWindows.cs – okno z listą kontaktów

ChatWindow.cs – okno rozmowy z użytkownikiem

LoginWindows.cs – okno logowania oraz wybór adresu i portu serwer

## Protokół komunikacyjny:

### a) Klient -> Serwer:

Login:

1;login;hasło|

Wyślij wiadomość:

3;adresat;wiadomość|

Wylogowanie:

4;1|

### b) Serwer -> Klient

Logowanie prawidłowe (nowy użytkownik lub prawidłowy login i hasło):

1;1|

Hasło nieprawidłowe:

1;0;Wrong password.|

Pozostałe nieprzewidziane problemy z logowaniem:

1;0;Unexpected error of server.|

Wysłanie listy użytkowników:

2;name1;name2;name3|

Wysłanie wiadomości:

3;name\_nadawca;czas;wiadomość|

Poprawne wylogowanie:

4;1|

## Obsługa projektu:

### a) Serwer

Stworzony został Makefile. W folderze serwera (communicator\_serwer/communicator\_serwer) należy uruchomić polecenie make, aby skompilować projekt. A następnie włączyć serwer poleceniem ./serwer.

### b) Klient

Wymagany program Visual Studio w wersji minimum 2013. Zbudowanie i uruchomienie projektu nie wymaga większego wyjaśnienia (standardowy build & run). Plik wykonywalny .exe projektu w standardowej lokalizacji (communicator-socket/CommunicatorSocket/obj/Debug lub communicator-socket/CommunicatorSocket/obj/Release).

*Dariusz Pałuch,  
Tomasz Tomys*