Sieci Komputerowe 2 - laboratoria
Projekt zaliczeniowy
Komunikator IRC - sprawozdanie

18 stycznia 2016

Prowadzący:

dr inż. Michał Kalewski

Autorzy:

Michał Hejduk 117268

Maksymilian Marcinowski 117253

Zajęcia:

wtorek 16:50

1. Wstęp

Tematem projektu zaliczeniowego była implementacja komunikatora internetowego typu IRC z zastosowaniem architektury klient-serwer z użyciem protokołu TCP. Implementację serwerów współbieżnych należało wykonać dla systemów operacyjnych GNU/Linux z użyciem języka C/C++ a implementację aplikacji klienta - dla systemów Microsoft Windows z użyciem języka C/C++, C#, Java lub Python wraz z graficznym interfejsem.

2. Podstawowe informacje o aplikacji

Serwer - plik źródłowy "serwer.c", Klient - pliki źródłowe : pliki .cs w folderze IRC/IRC
Klient został zaimplementowany w języku C#, z użyciem wpf.

Współbieżność serwera obsługiwana jest poprzez tworzenie procesów potomnych za pomocą funkcji fork().
Gdy pojawia się nowy użytkownik, zostaje on dodany do tablicy użytkowników i przekazany procesowi potomnemu. Gdy klient połączy się z serwerem, wysyła do niego swój nick w postaci:

<new_user>nick<end>

Serwer odczytuje nick, dodaje go do tablicy przechowującej nicki a następnie wysyła do klienta listę wszystkich dostępnych pokojów (kanałów) w postaci

<room_list><pokój1name><pokój1maxmembers><pokój2name><pokój2maxmembers> ... <end>.

(oprócz głównego, istniejącego od początku w kliencie i będącego "pokojem domyślnym"). Klient odczytując listę wydziela z niej nazwy i maksymalną liczbę osób i wypisuje użytkownikowi istniejące na serwerze pokoje.

Do przechowywania listy kanałów (pokojów) na serwerze stworzyliśmy tablicę struktur. Bazowo istnieją trzy kanały, ale gdy inni użytkownik utworzy nowy, zostaje on dodane do tablicy na serwerze i informacja o tym jest wysyłana do klientów. (po wydaniu polecenia dodania pokoju, do serwera zostaje wysłana przez klienta wiadomość w postaci: *<new_room><name><maxmembers><end>)*

Wszyscy klienci dodają sobie nowy kanał do listy i uaktualniają ListBoxa.

Serwer odczytuje wiadomość, dokonuje iteracji po wszystkich użytkownikach których ID!=-1 (co oznacza, że są dostępni) i wysyła im otrzymany komunikat, który zostaje przez pozostałych klientów odpowiednio zinterpretowany. Wydanie przez użytkownika polecenia wysłania wiadomości powoduje przesłania do serwera komunikatu:

<new_message><godzina_nick_z_treścią><nazwapokoju><end>

Po otrzymaniu takiego komunikatu, serwer wysyła identyczny komunikat każdemu użytkownikowi, następnie aplikacja klienta interpretuje otrzymany komunikat. Jeżeli wartość *actualroom* (użytkownik jest w tym pokoju) jest wartością odpowiadającą pokojowi, z którego została wysłana wiadomość, to treść wiadomości zostaje wyświetlona w graficznym interfejsie użytkownika (dokładnie w roomChatboxie). Jeżeli zaś wartości te się różnią, otrzymuje on powiadomienie o nowej wiadomości w innym pokoju, do którego należy. Jeżeli nie należy do tego pokoju, wiadomość jest ignorowana przez serwer.

Po wydaniu polecenia odłączenia się użytkownika, klient wysyła do serwera komunikat:

<delete_user>nick<end>, Co zamyka połączenie i gniazdo.