Komunikator internetowy typu IRC

Jakub Frąszczak, Arkadiusz Michalski

1 Temat zadania

IRC (Internet Relay Chat) – usługa sieciowa umożliwiająca rozmowę na wielu dostępnych kanałach komunikacyjnych. Funkcjonuje w architekturze klient-serwer. Składa się z serwera oraz podłączanych programów - klientów. Użytkownik ma możliwość zapisania się istniejącego pokoju lub utworzenia własnego.

2 Opis protokołu komunikacyjnego

2.1 Klient wysyła do serwera 6 rodzajów komunikatow:

- #0 nazwa_uzytkownika$ zmienia nazwe użytkownika i dodaje go do listy użytkowników
- #1 nazwa_pokoju$ utworzenie pokoju i dodanie do niego użytkownika
- #2 nazwa_pokoju$ dołączenie użytkownika do pokoju
- #3 nazwa_pokoju$ wyjscie użytkownika z pokoju i jeśli pokój jest pusty po wyjściu usunięcie go
- #4 wiadomosc$ rozsyła daną wiadomość do pozostałych użytkowników
- #5 nazwa_pokoju$ prosi o dane o obecnym stanie serwera

2.2 Serwer natomiast wysyła stan pokoju do użytkowników w danym pokoju w formacie:

- #@nazwa_pokojuĥasło%uzytkownik1;uzytkownik2;...%wiadomosc1;wiadomosc2;...$

3 Implementacja

3.1 Klient:

Funkcje:

- receive() poprawne odczytywanie wiadomości
- read_server_response(response) odczytuje obecny stan serwera
- readMessage(message) odczytuje wysłaną wiadomość
- formatMessages(messages_raw) odczytuje wszystkie wiadomości
- roomExists(name, rooms) sprawdza czy pokój o zadanej nazwie istnieje
- countUsers() zwraca liczbę użytkowników
- update_server_state(response, room) zapisuje w informacje o wszystkichh pokojach oraz użytkownikach i wiadomościach
- askForServerState(room) pyta o stan serwera
- checkPassword(name, password) sprawdza czy podane hasło jest poprawne
- checkUsers(nick) sprawdza czy w pokoju nie ma już użytkownika o takim samym nicku
- resetRoom() zeruje wszystkie dane o pokoju
- join(self) dołączenie do pokoju
- runGui() wątek Gui
- runClient() wątek nasłuchującego klienta

3.2 Serwer:

Funkcje:

- update_server_response(int roomId) tworzy w tablicy server_response[] string wysyłany do klienta
- deleteUser(int user_id) usuwa użytkownika zajmującego podany slot zerując wszystkie jego tablice
- deleteChatroom(int chatroom_id) usuwa dany chatroom
- isChatroomEmpty(int chatroom_id) zwraca 1 dla pustego pokoju
- getChatroomIDbyName(char *name) zwraca ID chatroomu
- getFristFreeSlotInChatroom(int chatroom_id) zwraca najmniejszy indeks, pod którym do pokoju można dodać użytkownika
- getFirstFreeUserSlot() zwraca najmniejszy indeks, pod którym na serwerze można zapisać użytkownika
- findUserInChatroom(int user_id, int chatroom_id) zwraca indeks użytkownika w pokoju
- joinChatroom(int user_id, int chatroom_id) użytkownik o danym ID dołącza do określonego chatroomu
- getFirstFreeChatroomID() zwraca minimalny indeks w tablicy pokoi, pod którym można umieścić nowy pokój
- sendToChatroom(int chatroom_id, char message[MESSAGE_LENGTH], int msg_length) przesyła wiadomość do określonego pokoju
- broadcast_server_response(int bytes, int userId, int roomId) odpowiedź serwera do wszystkich użytkowników
- *Thread_Listening(void *t_data) wątek tworzony dla każdego połączonego klienta (komunikacja została opisana powyżej)
- handleConnection(int connection_socket_descriptor) przyjmuje połączenie od użytkownika

4 Kompilacja i uruchomienie

4.1 Klient został napisany za pomocą języka Python. Interfejs użytkownika stworzony w oparciu o nakładkę na bibliotekę Qt - PyQt.

```
cd IRC
python interface.py
```

4.2 Serwer jest napisany w C, uruchamialny wyłącznie w systemie Linux. Kompilacja i uruchomienie z konsoli poleceń:

```
cd IRC
gcc -Wall server.c -o server -pthread
./server numer_portu
```