

Sprawozdanie Sieci Komputerowe 2

System komunikacji grupowej typu IRC

Mateusz Kempa 139952 Mateusz Lemański 136276

1. Opis protokołu komunikacyjnego

Komunikacja klient-serwer przebiega następująco:

Wiadomości, które otrzymuje serwer:

- Nadanie nicku klientowi, sprawdzenie unikalności i maksymalnej liczby użytkowników. Długość nicku jest uzupełniana nieznaczącymi zerami z lewej jeżeli ma mniej niż 10 bajtów.

5 znakowy kod wiadomości: login	długość nicku (max 99 bajtów)	Nick
---	-------------------------------	------

- odebranie prośby klienta o zmianę pokoju

5 znakowy kod wiadomości: cRoom	nr pokoju do którego chce przejść klient
--	--

- Prośba o stworzenie nowego pokoju na wolnym slotie (jednym z dwóch). Długość nazwy pokoju jest uzupełniana nieznaczącymi zerami jeżeli ma mniej niż 10 bajtów.

5 znakowy kod wiadomości: aRoom	długość nazwy pokoju (max 99 bajtów)	Nazwa pokoju
---	--------------------------------------	--------------

- Prośba o usunięcie pokoju z wolnego slotu. Długość nazwy pokoju jest uzupełniana nieznaczącymi zerami jeżeli ma mniej niż 10 bajtów. W tym przypadku nazwa pokoju to + czyli znak wolnego slotu.

5 znakowy kod wiadomości: dRoom	długość nazwy pokoju (max 99 bajtów)	Nazwa pokoju czyli +
---	--------------------------------------	-------------------------

- Odebranie wiadomości od klienta. Nick użytkownika serwer określa sam za pomocą numeru deskryptora. Długość wiadomości jest uzupełniana nieznaczącymi zerami.

5 znakowy kod wiadomości: mSend	długość wiadomości (max 999 bajtów)	Treść wiadomości
---	-------------------------------------	------------------

- Ostatnia opcja to: **close**. Oznacza ona zamknięcie aplikacji przez użytkownika (serwer zamyka wątek danego klienta i zwalnia jego deskryptor i usuwa jego dane).

Wiadomości, które otrzymuje klient:

- Wiadomości dotyczące logowania: 0- powodzenie, 1- przekroczono maksymalną liczbę użytkowników, 2- niepowodzenie (błąd odczytu, lub błąd unikalności nicku).
- Wysłanie wiadomości do użytkowników znajdujących się w tym samym pokoju. Zarówno długości nicku jak i wiadomości są uzupełniane nieznaczącymi zerami z lewej strony.

5 znakowy kod wiadomości: messU	długość nicku wysyłającego użytkownika (max 99 bajtów)	Nick	Długość wiadomości (max 999 bajtów)	Treść wiadomości
---	--	------	-------------------------------------	------------------

- Dołączenie/opuszczenie pokoju przez użytkownika.

5 znakowy kod: roomJ	dł nicku osoby dołączającej (max 99 bajtów)	Nick
5 znakowy kod: roomL	dł nicku osoby opuszczającej (max 99 bajtów)	Nick

- Dodanie/usunięcie pokoju na wolnym slocie.

5 znakowy kod: roomA	Numer dodanego pokoju	Długość nazwy nowego pokoju (max 99 bajtów)	Nazwa nowego pokoju
5 znakowy kod: roomD	Numer usuniętego pokoju	Długość nazwy pokoju po usunięciu (max 99 bajtów)	Nazwa po usunięciu czyli +

2. Opis struktury projektu.

Serwer:

Program serwera znajduje się w folderze Server w pliku "server.cpp". Serwer na początku wykonuje potrzebne operacje sieciowe wymagane do komunikacji z klientem. Dla każdego kolejnego klienta tworzony jest nowy wątek w którym odbywa się komunikacja między danym klientem a serwerem. Po zamknięciu aplikacji klienckiej, wątek klienta zostaje zakończony i wszystkie jego dane są usuwane, deskryptor również zostaje zwolniony.

Strukturą user zawiera informacje użytkownika czyli: nick (nazwa użytkownika), descriptor (deskryptor użytkownika), room (pokój w którym znajduje się użytkownik).

Struktura thread_data_t zawiera dane, które zostają przekazane do wątku: strukturę user z informacjami o użytkowniku. Wektor users, który zawiera dane wszystkich zalogowanych użytkowników. Zmienną boolowską errorHandler, która informuje o wystąpieniu błędu podczas działania funkcji read/write. Mutex: users_mutex (synchronizacja współbieżności na wektorze users). Tablica mutexów dla każdego pokoju: rooms_mutex[6] (synchronizacja współbieżności podczas operacji w pokojach).

Klient:

Program klienta obsługuje 5 wątków. Główny wątek: initialize jest odpowiedzialny za akcje użytkownika związane z interfejsem graficznym aplikacji odbiera on również komunikaty od serwera i wykonuje odpowiednie akcje, działa on w pętli aż do zakończenia działania aplikacji. Wątek NewRoom uruchamiany jest gdy użytkownik chce dodać nowy pokój. Wątek DeleteRoom uruchamiany jest gdy użytkownik chce usunąć pokój. Wątek RoomChange jest uruchamiany gdy użytkownik chce zmienić pokój. Wątek: Message jest uruchamiany gdy wątek chce wysłać wiadomość do pokoju w którym aktualnie się znajduje.

3. Opis sposobu kompilacji i uruchamiania projektu

Serwer:

Należy przejść do folderu Server w terminalu. Gdy jesteśmy w odpowiedni katalogu należy skompilować program poprzez polecenie: `g++ -pthread -Wall server.cpp -o server`

Po skompilowaniu jeżeli nie wystąpiły żadne błędy można uruchomić serwer za pomocą polecenia: `./server 3333 127.0.0.1`

3333 - to numer portu na którym ma działać aplikacja (ten sam należy podać w panelu logowania aplikacji klienta)

127.0.0.1 - to adres hosta (ten sam należy podać w panelu logowania aplikacji klienta)

Klient:

Aplikacja klienta została napisana w java GUI przy użyciu JavaFX. Należy otworzyć folder Client z projektem w wybranym środowisku, które obsługują javaFX (twórcy używali IntelliJ). Należy dodać konfigurację, skompilować i uruchomić cały program.