

IPC Chat Service — projekt zaliczeniowy

Igor Szczepaniak
inf151918

Politechnika Poznańska
styczeń 2023

1 Zasady funkcjonowania relacji klient-serwer

Komunikacja odbywa się za pomocą jednej kolejki komunikatów tworzonej przez serwer. Każdemu użytkownikowi przydzielany jest numer identyfikacyjny (zależny od zmiennej `MAX_USERS`, czyli maksymalnej liczby użytkowników na serwerze, oraz od wartości jaką zwraca funkcja mieszająca na podstawie nazwy użytkownika), który stanowi jedyny typ wiadomości, na który dany użytkownik będzie oczekiwać. Serwer nasłuchuje wiadomości o typach 1-8 i nadaje wiadomości przeznaczone dla konkretnych użytkowników.

Użytkownicy logują się do serwera za pomocą loginu oraz hasła. Po trzykrotnej, błędnej próbie logowania konto użytkownika zostaje zablokowane.

2 Struktura wiadomości kolejki

```
struct {  
    long mtype;  
    int code;  
    char sender [];  
    char receiver [];  
    char mtext [];  
} msgbuff;
```

- `mtype` - typ wysyłanej wiadomości
- `code` - kod informujący o błędzie
- `sender` - nadawca wiadomości
- `receiver` - odbiorca wiadomości
- `mtext` - treść wiadomości

3 Opis typów wiadomości

- 1 - żądanie logowania
- 2 - żądanie listy zalogowanych użytkowników
- 3 - żądanie wylogowania
- 4 - żądanie wysłania wiadomości do użytkownika
- 5 - żądanie listy grup użytkowników
- 6 - żądanie wysłania wiadomości do grupy użytkowników
- 7 - żądanie dołączenia do grupy
- 8 - żądanie opuszczenia grupy
- $[MAX_USERS; 2 \times MAX_USERS]$ - typy wiadomości, na które oczekują użytkownicy

4 Przykładowe żądania i odpowiedzi

Użytkownik test1 wysłał żądanie wysłania wiadomości do użytkownika test2:

```
struct {  
    long mtype=4;  
    char sender []="test1";  
    char receiver []="test2";  
    char mtext []="Hello _there!";  
}msgbuff;
```

Serwer po otrzymaniu żądania przesyła wiadomość użytkownikowi test2:

```
struct {  
    long mtype=32; //unikatowy mtype dla test2  
    char sender []="test1";  
    char receiver []="test2";  
    char mtext []="Hello _there!";  
}msgbuff;
```

Użytkownik test1 wysyła żądania dołączenia do grupy `group0`:

```
struct{
    long mtype=7;
    char sender []="test1";
    char receiver []="server";
    char mtext []="0";
}msgbuff;
```

Grupa okazuje się pełna - serwer odpowiada:

```
struct{
    long mtype=58; //unikatowy mtype dla test1
    int code=403 //informacja o niepowodzeniu;
    char sender []="server";
    char receiver []="test1";
}msgbuff;
```