

Hubert Kamieniarz 136716

Konrad Socha 136800

Grupa poniedziałkowa - 16:50

Komunikator internetowy typu GG

20.01.2020

TEMAT ZADANIA

Implementacja komunikatora internetowego na wzór "Gadu-Gadu". Każdy użytkownik posiada swój unikalny numer, który wykorzystuje się przy rejestracji i logowaniu oraz w celu zaadresowania wiadomości. Komunikaty nadane do zalogowanych w danej chwili użytkowników przekazywane są natychmiast, natomiast w przypadku niezalogowanych - dostarczane są przy najbliższym logowaniu użytkownika.

PROTOKÓŁ KOMUNIKACYJNY

Typy komunikatów wysyłane przez klienta:

- ('1' + numer_użytkownika) - rejestracja/logowanie w aplikacji
- ('2' + numer_adresata + ';' + treść_wiadomości) - wiadomość do innego użytkownika
- '0' - wylogowanie z aplikacji

Typy komunikatów wysyłane przez serwer:

- ('\$;' + tresc_komunikatu) - komunikat systemowy (np. o błędzie)
- (numer_nadawcy + ';' + tresc_wiadomosci) - nowa wiadomość do użytkownika

IMPLEMENTACJA

Serwer

- struct Message - struktura opisująca pojedynczą wiadomość
- struct User - struktura opisująca pojedynczego użytkownika
- struct cln - struktura zawierająca dane połączenia klienta
- void passMessages(int id) - przesłanie wiadomości do użytkownika, gdy ten jest zalogowany
- int findUser(char *number) - zwróć id (indeks na liście 'users') użytkownika o danym numerze
- void *serve_single_client(void *arg) - obsługa klienta

Klient

- `fill_up_list()` - uzupełnienie listy aktywnych konwersacji
- `refresh_list()` - obsługa listy aktywnych konwersacji
- `fetch_new_messages()` - obsługa wiadomości odebranych od serwera
- `display_new_messages()` - wyświetlanie nowych wiadomości w oknie czatu
- `display_chat_history()` - wyświetlanie historii czatu
- `open_chat(number)` - wyświetlanie okna czatu
- `open_chat_new()` - otwarcie czatu z nowym rozmówcą poprzez wpisanie numeru
- `open_chat_list(item)` - otwarcie czatu z nowym rozmówcą za pośrednictwem listy aktywnych rozmów
- `send_message()` - obsługa nowej wiadomości wychodzącej
- `log_in()` - logowanie użytkownika

KOMPILACJA

Serwer należy skompilować na systemie Linux poleceniem:

```
gcc server.c -o server.o -Wall -lpthread
```

A następnie uruchomić:

```
./server.o
```

Serwer nie wymaga dalszej obsługi, komunikaty dotyczące jego pracy będą na bieżąco wyświetlane w terminalu.

Jeśli serwer jest uruchomiony, klienta należy uruchomić poleceniem:

```
python klient.py
```

Zostanie wyświetlony interfejs graficzny - należy podążać za wyświetlanymi na nim instrukcjami.