

Julia Chałasiak 136531
Miłosz Książek 136750
Piątek 8:00 I5.1

Projekt zaliczeniowy - Sieci Komputerowe 2 laboratorium

1. Opis protokołu komunikacyjnego

Wiadomości od klienta do serwera w formie:

#od:do:wiadomość\$

Serwer analizuje wiadomość i wysyła ją tylko do użytkownika o nazwie "do".

Po zalogowaniu się nowego klienta wysyła on wiadomość:

*nazwa\$

a następnie serwer rozsyła tę nazwę do istniejących użytkowników oraz przesyła nowemu użytkownikowi nazwy istniejących.

2. Opis implementacji

a. Serwer

Serwer bazuje na tym, który tworzyliśmy podczas zajęć laboratoryjnych. Dołączających użytkowników zapisujemy w tablicy struktur, w skład której wchodzi socket i nazwa użytkownika w formie *nazwa\$.

Wątek w pętli while odczytuje wiadomości i jeśli odebrana ilość bajtów jest większa od 0, to przystępuje do zinterpretowania wiadomości. Klient może wysłać do serwera 2 typy wiadomości - podczas logowania *nazwa\$, co spowoduje dodanie tej wiadomości do users, a następnie wysłanie do nowego użytkownika wiadomości z istniejącymi użytkownikami, żeby znalazły się one u niego na liście kontaktów. Wiadomość z nazwą nowego użytkownika jest również wysyłana do już istniejących. Drugi typ wiadomości, to wiadomość z jakąś treścią wysłana od jakiegoś użytkownika do innego w formie #od:do:wiadomość\$. Najpierw wyodrębniane jest "do", żeby porównywać z istniejącymi nazwami w users, znaleźć należący do niego socket i wysłać do niego wiadomość. Korzystanie z users musi być sekcją krytyczną, żeby mógł tę tablicę jednocześnie edytować tylko jeden klient (realizacja za pomocą mutexu).

b. Klient został zaimplementowany w pythonie. Do stworzenia interfejsu graficznego Używamy dwóch klas - pierwsza to User, w której przy tworzeniu obiektu podajemy nazwę. Oprócz tego każdy User posiada listę z wiadomościami, które są potem wczytywane do okienka z rozmową. Druga klasa to Message - zawiera 3 pola odzwierciedlające od kogo, do kogo i o jakiej treści jest wiadomość. Metoda send wysyła do serwera wiadomość w formie #od:do:wiadomość\$ zakodowaną w UTF-8.

Jednocześnie z okienkiem graficznym działa funkcja odbierająca wiadomości od serwera. W pętli while odbiera 128 bajtów do momentu znalezienia znaku "\$", co oznacza koniec wiadomości, a następnie analizuje ją, żeby dodać nowy kontakt albo wiadomość do listy wiadomości z jakimś użytkownikiem. Gdy przychodząca

wiadomość należy do aktualnie otwartej rozmowy, to dodaje ją od razu do okienka z rozmową.

3. Opis sposobu kompilacji

Serwer:

Będąc w folderze z plikiem, w terminalu wpisujemy komendę:

```
g++ -o serwer ./serwer.cpp
```

Następnie:

```
./serwer numer portu
```

Klient:

W terminalu:

```
python3 client.py
```

4. Opis uruchomienia

Po skompilowaniu, w kliencie wypełniamy host i port, a następnie nazwę użytkownika. Po uruchomieniu większej ilości klientów możemy sprawdzić działanie programu.