

Sprawozdanie sk2

Komunikator internetowy

1. Opis projektu.

Komunikator napisany jest w architekturze klient-serwer. Serwer używa wielu wątków, by umożliwić połączenie się do niego wielu klientów. Klient ma możliwość rejestracji, logowania oraz wysłania wiadomości do każdego użytkownika podłączonego do serwera.

Użyta technologia:

- serwer w C
- klient w Python
- bsd sockets
- threads
- tkinter oraz turtle (do GUI)
- zapisywanie do plików .txt

2. Opis komunikacji pomiędzy serwerem i klientem.



Zaraz po uruchomieniu programu klienta, łączy się on z serwerem, a po udanym zalogowaniu tworzony jest drugi wątek klienta, który również łączy się z serwerem. Drugi wątek służy tylko do odbierania wiadomości od innych użytkowników. Programy komunikują się wysyłając do siebie wiadomości za pomocą socketów. Najczęściej jest to sekwencja wiadomości - pierwsza wiadomość określa rodzaj wykonywanej akcji, a następne wiadomości zawierają szczegółową informację o dotyczącej tej akcji, np. najpierw wysyłana jest wiadomość "register", a dopiero potem w dwóch wiadomościach nazwa użytkownika i hasło.

3. Podsumowanie.

Dane użytkownika jak i lista użytkowników online zapisywana jest w plikach .txt po stronie serwera. Razem z hasłem, w pliku .txt użytkownika przechowywany jest identyfikator socketu łączącego wątek klienta odbierający wiadomości, z serwerem. Aplikacja jest odporna na podstawowe błędy, takie jak: błędne dane logowania,

próba rejestracji istniejącego już użytkownika, próba wysłania pustej wiadomości. Wylogowanie z serwera następuje po wyłączeniu GUI aplikacji.

Wysyłanie wiadomości możliwe jest tylko do zalogowanych użytkowników. Są oni wyświetlani po naciśnięciu przycisku 'users online' jako osobne przyciski, kliknięcie na dowolny z nich otwiera nowe okienko do wysyłania wiadomości. Wszystkie wiadomości od innych użytkowników pojawiają się w jednym okienku wraz z ich nickami. Okienko mieści 12 wiadomości średniej długości (max. szerokość okienka - reszta nie będzie widoczna), zatem można czytać wiadomości z niedalekiej przeszłości. Kiedy na okienku wyświetlonych jest 12 wiadomości i przyjdzie nowa wiadomość, okienko jest czyszczone, a nowa wiadomość pojawia się na samej górze.

Zostawiliśmy pomocnicze printy w terminalu, które pomagają zrozumieć kolejność działania komunikacji. Nieobsługiwane błędy, o których wiemy: przy wylogowaniu się użytkownika o danej nazwie np. "user", z listy użytkowników online usuwani są wszyscy użytkownicy, co posiadają substring identyczny z nazwą wylogowanego użytkownika.

Początkowo problem sprawiała implementacja zapisywania danych o użytkownikach ze strony serwera. Większe trudności napotkaliśmy przy dekodowaniu wiadomości w wątku służącym do odbierania wiadomości od innych użytkowników jak i sama implementacja GUI służącego do ich wypisywania. Większość używanych do tego bibliotek musi być używana w głównym wątku i nie obsługuje wielu wątków.