

Sprawozdanie

SK2 - projekt

Jędrzej Kościelniak, 145260, I5.2
Bartłomiej Kowalewski, 145204, I5.2

1. Opis projektu

Celem naszego projektu było zaimplementowanie klienta i serwera działającego w systemie publish/subscribe. Klient przesyła żądanie wykonania danej operacji, która weryfikowana i przetwarzana jest na serwerze. Klient może wykonać operacje:

- dodania tematu
- zasubskrybowania tematu
- anulowania subskrypcji tematu
- wysłania wiadomości
- wyświetlenia listy tematów
- wyświetlenia wiadomości z zasubskrybowanych tematów

2. Opis protokołu komunikacyjnego

Aplikacja wykorzystuje TCP jako model komunikacji. Serwer jest wielowątkowy. Każdy wątek odpowiada za połączenie z jednym klientem. Klient wykorzystuje socket do komunikacji z serwerem. Wysłanie wiadomości pomiędzy serwerem a klientem odbywa się na dwa sposoby. Pierwszy z nich polega na wysłaniu żądania wykonania operacji zapisanego w następujący sposób:

"[litera służąca za odpowiednią operację][długość tematu][temat][wiadomości]" np. "s05booksKsiążki".

Serwer wykonuje akcję w zależności od pierwszej litery. Drugi sposób polega na wysłaniu wiadomości zwrotnej do klienta po wykonaniu akcji przez serwer (zarówno w przypadku powodzenia jak i niepowodzenia operacji).

3. Opis struktury projektu

a. Klient

Klient został zaimplementowany w języku Python. Do komunikacji z serwerem użyliśmy biblioteki `socket`. Interfejs został wykonany z wykorzystaniem biblioteki `tkinter`. Użytkownik może wykonywać akcje i

komunikować się z serwerem poprzez klikanie w odpowiednie przyciski. W zależności od akcji wyświetlane są osobne okna.

b. Serwer

Serwer napisany został w języku C. Za wielowątkowość odpowiada biblioteka *pthread.h*. Każdy wątek w pętli oczekuje na komunikaty od klienta, za którego połączenie odpowiada. W zależności od danej operacji modyfikuje stan na serwerze lub/i odsyła wiadomość zwrótną do klienta.

4. Kompilacja i uruchomienie

Kompilacja pliku serwera z flagą `-Wall` nie wykazała żadnych ostrzeżeń. Serwer został również pozytywnie sprawdzony pod kątem wycieków pamięci poprzez narzędzie do debugowania *valgrind*.

W celu uruchomienia programu najpierw należy skompilować serwer z wykorzystaniem flagi `-pthread`:

```
"gcc -pthread server.c -o server"
```

Po skompilowaniu, serwer uruchamiamy poleceniem `./server`. Natomiast klienta uruchamiamy poleceniem `python3 client.py [adres_ip] 1234`.

Do uruchomienia klienta wymagane jest posiadanie zainstalowanej biblioteki ***tkinter***.

Projekt został umieszczony na Githubie pod linkiem: [KowalewskiBartlomiej/SK2_Projekt: Repozytorium projektu zaliczeniowego z przedmiotu Sieci Komputerowe 2 \(github.com\)](https://github.com/KowalewskiBartlomiej/SK2_Projekt)