

Dokumentacja do projektu programistycznego na przedmiot SKJ

Autor: Rafał Wójcik, student PJATK: s21235

Zostały zrealizowane wszystkie funkcje klienta i serwera, które były podane w zadaniu.

Po stronie serwera, w tym samym momencie otwieram wszystkie gniazda UDP, które zostały podane w liście, po czym równolegle na każdym porcie czekam na sekwencje wiadomości. Po otrzymaniu prawidłowego ciągu, serwer będzie oczekiwał na klienta, po dotarciu klienta, stworzy połączenie, a następnie przekaże to połączenie do osobnego wątku, a sam będzie znowu oczekiwał na kolejne komunikaty od kolejnych hostów.

Po stronie klienta, klient zaczyna pukać równocześnie do wszystkich portów UDP jakie zostały podane w liście, a następnie komunikuje się z nimi niezależnie od innych.

Protokół „SKJ-validation”:

Serwer UDP czeka na 3 pakiety danych o rozmiarze 1 bajt każdy. Po otrzymaniu trzech pakietów danych serwer UDP sprawdza, jakie pakiety otrzymał i od kogo. Jeśli wszystkie trzy pakiety zostały otrzymane od tego samego klienta, a ponadto bajty z kolejno otrzymanych pakietów, w formacie UTF-8, po zamienieniu na znaki i późniejszym dodaniu ich do siebie, w kolejności dotarcia do serwera tworzą napis „SKJ”, to wtedy serwer uznaje takiego klienta za prawidłowego i wysyła takiemu hostowi port serwera TCP, który posłuży do dalszej komunikacji. Klient po wysłaniu pakietów danych, czeka na odpowiedź od serwera UDP przez następne 5 sekund. Jeśli w ciągu 5 sekund nie zostanie żadnej odpowiedzi to kończy pracę. W przeciwnym wypadku, łączy się z serwerem TCP na porcie otrzymanym w odpowiedzi. Po ustanowieniu połączenia pomiędzy serwerem a klientem, klient wysyła żądany format daty w postaci kodu dowolnie wybranego języka.

(<https://www.oracle.com/java/technologies/javase/jdk8-jre8-supported-locales.html>)

Jeśli serwer nie odbierze wiadomości z kodem języka przez 5 sekund od ustanowienia połączenia, to uznaje, że musiał wystąpić błąd i kończy połączenie. W przeciwnym razie, serwer odbiera wiadomość, a następnie przesyła klientowi dzisiejszą datę sformatowaną zgodnie z podanym przez niego kodem języka. Po wysłaniu wiadomości, serwer zamyka połączenie. Po otrzymaniu wiadomości, klient zamyka połączenie.

Przykład działania protokołu:

//klient wysyła sekwencje wiadomości

C:UDP: S

C:UDP: K

C:UDP: J

//prawidłowa sekwencja, serwer odsyła port tcp i czeka na połączenie

S:UDP: <port tcp>

//ustanowienie połączenia TCP, klient wysyła żądany język

C:TCP: pl

S:TCP: 5 sty 2021

