E-goat

Sieć wymiany plików peer-to-peer

<https://github.com/julnow/e-goat/>

**Założenia**

Projekt składa się z 2 programów: klienta i serwera. Na serwerze przechowywana jest lista plików udostępnianych przez poszczególnych klientów, wraz z informacją o klientach, którzy je posiadają. Klienci wymieniają się plikami bezpośrednio między sobą.

**Wymagania**

- Java 15

**Opis działania**

Do komunikacji pomiędzy serwerem a klientami wykorzystany jest protokół UDP (podczas komunikacji sprawdzane są sumy kontrolne poszczególnych plików, nie potrzebna jest zatem kontrola na poziomie protokołu).

1. Uruchamiamy serwer:

*$ java UDPServer*

1. Uruchamiamy klienta:

$ java UDPClient

1. Podajemy ścieżkę do katalogu, w którym znajdują się udostępniane pliki. Klient oblicza sumy kontrolne *SHA512* plików znajdujących się w katalogu i wysyła do serwera wiadomość *List of files*. Następnie wysyła do serwera listę z sumami kontrolnymi plików

3a) Do przechowywania informacji o plikach (ich sumy kontrolnej i adresu) stworzona została dodatkowa klasa *Files.*

1. Serwer po odebraniu listy wysyła odpowiedź *files added to the list*.
2. Następnie klient prosi o podanie sumy SHA512 pliku, który chcemy pobrać i wysyła ją do serwera.

6) Serwer w odpowiedzi odsyła listę wszystkich klientów (a dokładniej ich adresów), którzy zgłosili, że taki plik posiadają. W przypadku gdy nikt nie udostępnia takiego pliku serwer wysyła odpowiednią wiadomość.

7) (niezaimplementowane) Jeśli plik istnieje, klient pobiera go bezpośrednio od osoby, która go udostępnia i sprawdza czy otrzymał plik o prawidłowej sumie kontrolnej.

**Opis kodu:**

* Klasa Files



Obiekty klasy *Files* zawierają adres, gdzie znajduje się kod konkretnego pliku (*address*) oraz jego hash *sha512*. Ponadto może zapisać *IP* klienta posiadającego plik i jego *port*.

* Klasa UDPClient

Klasa UDPClient zawiera funkcję, zamieniającą dany plik na strumień bitów, a następnie obliczającą *sha512* pliku:



Następnie klient nawiązuje połączenie z serwerem i prosi użytkownika o podanie ścieżki do folderu, który ma być udostępniony. Dla każdego z plików znajdujących się w folderze tworzony jest hash:



Następnie lista plików wysyłana jest na serwer jako string w formacie *sha512*+*’\t’*+*adres*+*’\n’* i sprawdzana jest odpowiedź serwera.



Następnie pyta użytkownika, czy chce pobrać jakiś plik – jeżeli tak (‘*y’*), pobiera listę dostępnych plików:



Po wysłaniu danych klient prosi użytkownika o podanie *sha512* pliku, którego dostępność chciałby sprawdzić na serwerze, aby następnie móc go pobrać:



* Klasa UDPServer

Serwer otwiera gniazdo z konkretnym portem, a następnie zaczyna nasłuchiwać:



Jeżeli otrzyma komunikat *„List of files”* zaczyna zapisywać otrzymane pliki w *ArrayList<Files> files* – odczytuje string w tym samym formacie, w jakim kodowany jest przez klienta (*sha512*+*’\t’*+*adres*+*’\n’*):



Jeżeli otrzyma komunikat *‘y’*, zwraca listę dostępnych plików:



W przeciwnym razie, próbuje odebrać *sha512* pliku, który ma być pobrany, a następnie szuka go w liście plików *files* i wysyła odpowiedź składającą się z adresów wszystkich plików o tym hashu:

