Nicoll Bryła

Opis projektu TFTP

1.Opis kodu znajdującego się w clientTFTP.py:

Program przyjmuje trzy parametry. Pierwszy to nazwa serwera, z którym ma się skomunikować; drugi to numer portu; trzeci to nazwa pliku, który ma ściągnąć.

Linie 1-28 to import potrzebnych bibliotek, inicjalizacja zmiennych oraz stworzenie nowego socketa.

Linia 29/30 zawiera żądanie odczytu, które za chwilę wyślemy do serwera.

Pętla while, która obejmuje linie 33-73 przetwarza wszystkie wiadomości, które odbieramy od serwera, dopóki nie dostaniemy informacji o błędzie.

Linie 35-39: jeżeli pojawił się wyjątek timeout i jeszcze nie odebraliśmy potwierdzenia na żądanie odczytu to wysyłamy je jeszcze raz, a jeżeli to potwierdzenie otrzymaliśmy już wcześniej to ponownie wysyłamy serwerowi potwierdzenie ostatniego dobrego pakietu.

Linie 43-47: Dostajemy wiadomość dotyczącą wielkości okna, więc zapisujemy wilkość okna i wysyłamy do serwera wiadomość o akceptacji.

Linie 48-52: Dostajemy wiadomość o błędzie, zatem zamykamy połączenie oraz wypisujemy odpowiedni komunikat na konsolę.

Linie 53-73: Dostajemy wiadomość z danymi. Jeżeli pakiet, który przyszedł jest następnym brakującym to go zapisujemy. Jeżeli otrzymaliśmy już pakiety w ilości *windowsize* to wysyłamy potwierdzenie do serwera z numerem ostatniego bloku lub jeżeli otrzymaliśmy ostatni plik to również wysyłamy potwierdzenie. *Po wysłaniu potwierdzenia kończymy odbieranie danych.* Jeżeli numer, który dostaliśmy jest większy od aktualnego to wysyłamy potwierdzenie z numerem ostatniego bloku, który pasował nam w kolejności. Resztę ignorujemy.

Linie 75-80:

Jeżeli odebraliśmy już cały plik i nie dostaliśmy żadnego komunikatu o błędzie to wypisujemy sumę kontrolną MD5 ściągniętego pliku.

2.Opis kodu znajdującego się w serverTftp.py:

Program przyjmuje dwa parametry. Pierwszy parametr to numer portu na którym ma nasłuchiwać. Drugi parametr to ścieżka katalogu, z którego ma wysyłać pliki.

Linie 1-17 to import potrzebnych bibliotek oraz inicjalizacja potrzebnych zmiennych.

Linie 30-53 to kod klasy ServerTftp, który tworzy socket oraz oczekuje w nieskończoność na zapytanie o plik od klienta. Jeżeli już otrzyma wiadomość to sprawdza czy otrzymany pakiet ma poprawny format oraz czy plik, o który został poproszony w ogóle istnieje. Jeżeli coś jest nie tak jak powinno wysyła do klienta wiadomość o błędzie, w przeciwnym razie wybieramy odpowiednia wielkość okna zgodnie z protokołem. Gdy mamy wszystkie potrzebne nam informacje tworzymy nowy wątek za pomocą klasy Client, aby obsłużyć to konkretne zapytanie.

Linie 56-138 zawierają kod wspomnianej już klasy Client.

W liniach 72-87 czekamy na potwierdzenie końca negocjacji wielkości *windowsize.* Jeżeli otrzymamy informację o błędzie lub 6 razy pojawi się wyjątek timeout to kończymy przesyłanie danych.

Pętla 90-97 czyta dane z pliku oraz pilnuje tego, aby pakietów gotowych do wysłania klientowi było zawsze tyle na ile windowsize został ustalony.

W pętli 101-105 przygotowane w pętli wcześniej pakiety zostają wysyłane.

W pętli while w liniach 108-130 oczekujemy na wiadomości potwierdzające otrzymanie przez klienta danych pakietów. Jeżeli pojawi się wyjątek timeout to znowu wysyłamy do klienta wszystkie wcześniej wczytane pakiety. Jeżeli po siódmym pojawieniu się wyjątku nadal nie dostaniemy wiadomości z akceptacją pakietów to przerywamy połączenie. V W liniach 114-121 liczę, a następnie usuwam pakiety, które już otrzymały potwierdzenia, aby nie wysyłać ich ponownie.