Serwer HTTP zgodny z RCF 2616 w zakresie żądań GET, HEAD, PUT, DELETE

Zuzanna Piniarska 136782
Mateusz Kałamoniak 136730

Opis protokołu

Serwer oczekuje na połączenie i czyta dane jednorazowo lub przez określony czas w zależności od ustawień serwera. Otrzymane zapytania mają formę “<METODA> <URL> HTTP/1.1\r\n<NAGŁÓWKI>\r\n\r\n<BODY>” . Zapytania przekazywane są do parsera, który rozpoznaje rodzaj żądania, relatywną ścieżkę na serwerze oraz wersję protokołu HTTP, która ze względu na wymagania RCF 2616 powinna być 1.1. Czytane są nagłówki zapytania oraz jeżeli wysłano zapytanie PUT czytane jest jego “body”. Sprawdzane jest wystąpienie nagłówka Content-Length, jeżeli występuje, to przechodzimy do kodu parsującego “body”, w przeciwnym razie poprawnie odczytano zapytanie. Czytanie “body” kończy się w kiedy ilość odczytanych bitów jest równa wartości nagłówka Content-Length.
W razie błędnego odczytu na dowolnym etapie zwracany jest kod 400 - Bad Request wraz z krótkim kodem html wyświetlającym informację o błędzie. W programie wykorzystano mutexa mającego na celu zabezpieczenie operacji na zbiorze połączeń. W celu zabezpieczenia konkretnego połączenia przed modyfikacją w tym samym czasie przez wiele wątków użyto shared pointery, które gwarantują, że operacje wykonane na konkretnej instancji nie naruszą obszaru krytycznego.

Opis implementacji

Implementacja samego serwera opiera się na podstawowej wiedzy z zakresu budowy serwera TCP. Serwer jest współbieżny, do czego wykorzystano wątki. Ścieżka zawarta w linku zapytania jest relatywną ścieżką do lokalizacji na serwerze. Serwer pozwala na odczyt i modyfikację plików w folderze podanym w konfiguracji serwera. Wszystkie odpowiedzi wysyłane są z nagłówkami Content-Length oraz Content-Type. Zapytanie GET zwraca zawartość pliku z serwera. Zapytanie HEAD zwraca to samo co GET jednak bez zawartości pliku, czyli kod odpowiedzi, wersję HTTP oraz nagłówki. Zapytanie PUT zapisuje

dowolny plik w dowolnej lokalizacji. Plik może być zapisany w dowolnie zagłębionym folderze. Jeżeli foldery lub foldery nie istnieją, to serwer tworzy je i zapisuje plik. Plik może zostać również nadpisany jeżeli istnieje. Zapytanie DELETE usuwa plik z serwera i zwraca kod 204 w razie sukcesu lub 400 w razie porażki.

Kompilacja, uruchomienie

Uruchomienie serwera

Serwer kompilujemy w linuxowym terminalu używając w folderze z kodem serwera polecenia : **g++ -Wall -pthread main.cpp connection.cpp connection.h connection_manager.cpp connection_manager.h request.cpp request.h request_handler.cpp request_handler.h request_parser.cpp request_parser.h response.cpp response.h server.cpp server.h settings.h -o server**

Serwer włączamy przez polecenie **./server**

Uruchomienie klienta

Najpierw trzeba zbudować moduł klienta c++ do programu w pythonie. Robimy to poleceniem **python setup.py install**

Następnie uruchamiamy kod poleceniem **python client.py**

W programie podajemy link w sposób
<IP>:<PORT>/<PATH>

Wybieramy metodę, dodajemy nagłówki i dla PUT wybieramy plik tekstowy do wysłania.
s