

Data oddania zadania 2.12.2005 godzina 23:59

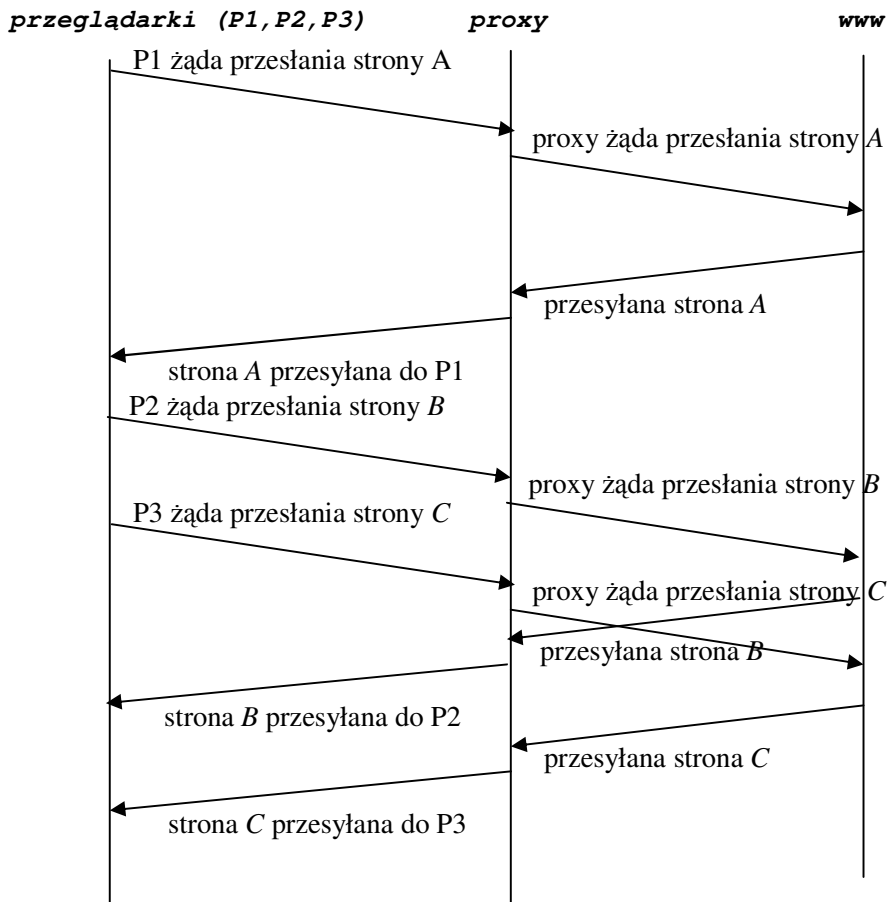
Maksymalna ilość punktów do zdobycia: 7 (5 za www + 2 za proxy)

Zadanie programistyczne 2: Wielowątkowy serwer WWW

Należy zaimplementować wielowątkowy serwer WWW w **Javie**, korzystający wyłącznie z gniazd TCP (nie z bibliotek Javy implementujących protokół HTTP). Serwer ma implementować metodę GET protokołu HTTP/1.0. Serwer powinien składać się z głównego wątku, który nasłuchuje na głównym gnieździe, oraz uruchamia odrębne wątki do obsługi żądań. Każde żądanie powinno zostać obsłużone przez oddzielny wątek, który po obsłudze żądania zamknie połączenie TCP. Serwer powinien współpracować ze standardową przeglądarką WWW (np. Mozilla). Serwer może działać na dowolnym porcie na adresie localhost. Dodatkowym parametrem serwera powinna być ścieżka dostępu do katalogu zawierającego udostępnione pliki. Serwer powinien poprawnie obsługiwać błędne żądania HTTP, zwracając odpowiednie komunikaty. Powinien jak najwierniej implementować standard HTTP/1.0 (RFC 1945).

Rozszerzona wersja zadania 2

Dodatkowo zaimplementuj wielowątkowy serwer *proxy*, który pośredniczyłby w przesyłaniu informacji pomiędzy przeglądarką, a właściwym serwerem *www*, bez żadnego *cache*-wania informacji. Serwer *proxy* powinien poprawnie przesyłać żądania klientów, zawartość stron i informacje o błędach. Po odpowiednim skonfigurowaniu przeglądarki (adres serwera *proxy* i numer portu), przesyłanie informacji powinno przebiegać zgodnie z poniższym scenariuszem: Załóżmy, że naszym komputerze uruchomione są trzy przeglądarki - P1, P2 i P3.



Poprawny projekt powinien zawierać:

- pliki źródłowe (*java*)
- pliki wykonywalne (*class*, względnie *jar*)
- plik *HowTo.txt*, o formacie jak poniżej
- dodatkowo, przy nagłówku każdej funkcji i każdej klasy powinien być krótki komentarz, w którym należy zamieścić informacje o przeznaczeniu danej klasy i funkcji (można to zrobić również w *javadoc-u*)
- projekt powinien być spakowany (najlepiej *zip*) i nazwany zgodnie ze wzorcem *YYY_SXXXX_ZZZZ.ZIP*, gdzie:
 - YYY - pierwsze trzy litery nazwiska prowadzącego
 - XXXX - numer indeksu studenta
 - ZZZZ - numer grupy

Format pliku *HowTo.txt*:

```
autor:
numer indeksu:
grupa:
przedmiot:

Zawartość projektu:
Ping.java - krótki opis zawartości pliku
Ping.class - krótki opis przeznaczenia pliku
.....
.....

Parametry programu Ping.class:
E:\>java Ping

Sposób użycia: java Ping [-t] [-a] [-n liczba] [-l rozmiar] [-f] [-i TTL] [-v TOS]
                [-r liczba] [-s liczba] [[-j lista_hostów] | [-k lista_hostów]]
                [-w limit_czasu] lista miejsc docelowych

Opcje:
-t                Odpytuje określonego hosta do czasu zatrzymania.
                  Aby przejrzeć statystyki i kontynuować,
                  naciśnij klawisze Ctrl+Break.
                  Aby zakończyć, naciśnij klawisze Ctrl+C.
-a                Tłumacz adresy na nazwy hostów.
-n liczba        Liczba wysyłanych powtórzeń żądania.
-l rozmiar       Rozmiar buforu transmisji.
-f                Ustaw w pakiecie flagę "Nie fragmentuj".
-i TTL           Czas wygaśnięcia.
-v TOS           Typ usługi.
-r liczba        Rejestruj trasę dla przeskoków.
-s liczba        Sygnatura czasowa dla przeskoków.
-j lista_hostów Swobodna trasa źródłowa wg listy lista_hostów.
-k lista_hostów Ścisłe określona trasa źródłowa wg listy lista_hostów.
-w limit_czasu  Limit czasu oczekiwania na odpowiedź (w milisekundach).

Przykładowe sposoby uruchomienia:

java Ping localhost - krótki opis
java Ping -n 100 -l 1 -i 9 www.wp.pl - krótki opis
```