

Sprawozdanie

Technologie sieciowe – lista 5

Aleksandra Malinowska, WPPT INF, 4 semestr, czerwiec 2019

1. Wstęp

1.1. Cel listy

Celem listy było przetestowanie serwera HTTP napisanego w języku Perl oraz zmodyfikowanie go na potrzeby zadań.

2. Serwer HTTP

2.1. Podstawowa wersja serwera

```
use HTTP::Daemon;
use HTTP::Status;

my $d = HTTP::Daemon->new(
    LocalAddr => 'localhost',
    LocalPort => 4321,
)|| die;

print "Please contact me at: <URL:", $d->url, ">\n";

while (my $c = $d->accept) {
    while (my $r = $c->get_request) {
        if ($r->method eq 'GET') {
            $file_s = "./index.html";
            $c->send_file_response($file_s);
        } else {
            $c->send_error(RC_FORBIDDEN)
        }
    }
    $c->close;
    undef($c);
}
```

Ten prosty serwer jest uruchamiany na porcie **4321** na lokalnym komputerze. Jego podstawowym i jedynym działaniem jest odpowiadanie na zapytania typu **GET**. Nie jest istotne dla niego jednak, o co tak naprawdę pyta klient, dlatego odpowiada mu zawsze przesłaniem pliku z katalogu, w którym znajduje się plik z serwerem, o nazwie **index.html**. Plik ten jest plikiem hipertekstowym i zawiera zawartość strony internetowej, która zostaje wyświetlona użytkownikowi po dowolnym zapytaniu.

2.2. Łączenie za pomocą przeglądarki

Aby połączyć się z serwerem należy wpisać w pasku adresowym przeglądarki następujący adres: **localhost:4321**. Efektem połączenia będzie wyświetlenie strony z pliku index.html. Dla sprawdzenia, że zapytanie nie ma znaczenia dla wyniku połączenia, można dodać do adresu, po znaku ukośnika (/) dowolny ciąg znaków. Zostanie on zupełnie zignorowany przez serwer.

2.3. Nagłówek jako odpowiedź

Aby serwer w odpowiedzi na zapytanie przysyłał nagłówek naszego zapytania należy zmienić wewnątrz głównej pętli skryptu.

```
while (my $c = $d->accept) {
    while (my $r = $c->get_request) {
        if ($r->method eq 'GET') {
            $resp= $r->header('Accept') ;
            $c->send_response($resp) ;
        } else {
            $c->send_error(RC_FORBIDDEN)
        }
    }
    $c->close;
    undef($c);
}
```

W powyższym skrypcie pogrubioną czcionką zaznaczono zmieniony fragment kodu. Zmiany te powodują wydobycie z zapytania, które przesyła klient, nagłówka i przesłanie go z powrotem do niego. Nagłówek zapytania HTTP zawiera dane takie jak: adres IP i numer portu źródła, żądane URI, typ przeglądarki itp.

2.4. Strona internetowa jako odpowiedź

Aby rozszerzyć działanie podstawowego serwera o wyświetlanie faktycznych stron, o które pyta klient, ponownie należy zmienić wewnątrz głównej pętli.

```
while (my $c = $d->accept) {
    while (my $r = $c->get_request) {
        if ($r->method eq 'GET') {
            $dir= "page";
            $path= $r->uri->path;
            if($path eq "/") {
                $path = "/index.html";
            }
            $file_s= $dir . $path;
            $c->send_file_response($file_s) ;
        }
        else {
            $c->send_error(RC_FORBIDDEN)
        }
    }
}
```

Zmiana ta powoduje wyświetlenie strony z katalogu **page**, która jest podana przez użytkownika w adresie po znaku ukośnika. W razie nie podania żadnej strony (wpisania samego adresu serwera), wyświetlony zostanie plik index.html. Natomiast gdy podany zostanie błędny adres, przeglądarka wyświetli błąd 404.

2.5. Przechwytywanie komunikatów

Do przechwytywania komunikatów między klientem a serwerem lokalnym na systemie Windows można użyć programu **RawCap**. Pozwala on na sprawdzenie co dokładnie jest przesyłane od i do serwera.

No.	Time	Source	Destination	Protocol	Length	Info
19	4.056747	127.0.0.1	127.0.0.1	HTTP	637	GET /index.html HTTP/1.1
35	4.067100	127.0.0.1	127.0.0.1	HTTP	244	HTTP/1.1 200 OK (text/html)
37	4.220580	127.0.0.1	127.0.0.1	HTTP	589	GET /favicon.ico HTTP/1.1
51	4.221577	127.0.0.1	127.0.0.1	HTTP	93	HTTP/1.1 404 Not Found (text/html)
62	6.544114	127.0.0.1	127.0.0.1	HTTP	656	GET /subpage.html HTTP/1.1
78	6.546110	127.0.0.1	127.0.0.1	HTTP	231	HTTP/1.1 200 OK (text/html)

Powyżej widać przykład przesyłania żądania o wyświetlenie strony internetowej z pliku index.html. Na to żądanie, serwer z rozdziału 2.5., odsyła zawartość witryny.

No.	Time	Source	Destination	Protocol	Length	Info
121	17.217851	127.0.0.1	127.0.0.1	HTTP	610	GET /test.html HTTP/1.1
131	17.218876	127.0.0.1	127.0.0.1	HTTP	42	HTTP/1.1 200 OK (text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3)

Kolejny przykład pokazuje działanie serwera z rozdziału 2.4. gdzie odpowiedzią na żądanie miał być nagłówek zapytania. Widać tu wyraźnie, że najpierw przesyłane jest zapytanie GET, a następnie serwer odsyła nagłówek w postaci tekstowej.

3. Podsumowanie

Funkcjonalności powyższych serwerów są bardzo ograniczone. W rzeczywistości serwer HTTP musi obsługiwać znacznie więcej żądań od klienta. Mimo to powyższe przykłady świetnie ukazują fundamentalną istotę istnienia serwera http – wyświetlanie stron internetowych.