**Badanie Właściwości serwerów http**

Technologie sieciowe

Filip Rurak, 244960

Laboratorium 9:15, poniedziałek

1. **Cel listy**

Celem zadania jest analiza programu serwer3.pl, podany na listingu nr 1. Przykładowe jego uruchomienie i modyfikacje według podanych poleceń.

use HTTP::Daemon;

use HTTP::Status;

#use IO::File;

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"; # index.html - jakis istniejacy plik

$c->send\_file\_response($file\_s);

}

else {

$c->send\_error(RC\_FORBIDDEN)

}

}

$c->close;

undef($c);

} Listing nr 1, plik serwer3.pl

1. **Realizacja**
   1. **Uruchomienie programu**

Po uruchomieniu programu zostaje wypisana linia Please contact me at:URL:http://DESKTOP-4SC\*\*\*\*:4321/ co oznacza oczekiwanie serwera na połączenie.

* 1. **Połączenie za pomocą przeglądarki**

Po stworzeniu skryptu podanego na listingu nr 2, uruchomieniu serwera zostaje nawiązane połączenie z nim wpisując w pasek wyszukiwań localhost:4321. Po wykonaniu zapytania pojawia się na stronie tekst Hello world!, który jest treścią skryptu index.html. Dzieje się tak, ponieważ serwer oczekuje na połączenie metodą GET i w razie jego otrzymania, przekierowuje użytkownika do skryptu html. W razie otrzymania innej metody, np. POST, serwer wyświetli błąd 403 Forbidden.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Hello</title>

</head>

<body>

<h1>Hello world! </h1>

</body>

</html> Listing nr 2, skrypt html do połączenia z serwerem

* 1. **Edycja pliku serwer3.pl, aby wysyłał nagłówek żądania**

W czwartym podpunkcie zadania zmodyfikowano plik tak, aby dla każdego żądania wysyłał jego nagłówek. Kod zmodyfikowanego pliku przedstawiony jest na listingu nr 3. Nie jest sprawdzane dla jakiej metody został wywołany tylko wysyła odpowiedź pobraną funkcją $r->headers\_as\_string.

use HTTP::Daemon;

use HTTP::Status;

use HTTP::Response;

#use IO::File;

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) {

$resp = HTTP::Response->new (200, 'Test ok');

$resp->content($r->headers\_as\_string);

$c->send\_response($resp);

}

$c->close;

undef($c);

} Listing nr 3, plik zwracający nagłówek

Przy ponownym wyszukaniu localhost:4321 przeglądarka wyświetla wiadomość pokazaną na listingu nr 4. W jej treści podane są: typ połączenia Connection: keep-alive , akceptowalne przez przeglądarkę treści, kodowanie i języki, adres hosta, i informacja Upgrade-Insecure-Requests: 1, mówiąca o tym, czy użytkownik preferuje szyfrowane połączenia.

Cache-Control: max-age=0

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

Accept-Encoding: gzip, deflate, br

Accept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7

Host: 127.0.0.1:4321

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36

Upgrade-Insecure-Requests: 1 Listing nr 4, odpowiedź przeglądarki

* 1. **Edycja pliku serwer3.pl, aby obsługiwał serwis WWW**

W piątym podpunkcie zadania zmodyfikowano plik serwer3.pl, którego edycję widać na listingu nr 5, aby umożliwiał obsługę serwisu WWW. Stworzone zostały trzy pliki a.html, b.html i c.html z wzajemnymi odwołaniami, przykład a.html podany jest na listingu nr 6. Jeśli nie został podana konkretny adres wyświetlony zostanie domyślny skrypt a.html.

use HTTP::Daemon;

use HTTP::Status;

use HTTP::Response;

#use IO::File;

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->uri eq "/") {

$file\_s = "./a.html";

} else {

$file\_s = "./".$r->uri;

}

$c->send\_file\_response($file\_s);

}

$c->close;

undef($c);

} Listing nr 5, program wysyłający odpowiednie pliki html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>a</title>

</head>

<body>

<h1>Test A</h1>

<ul>

<li><a href=b.html>Test B</a>

<li><a href=c.html>Test C</a>

</ul>

</body>

</html> Listing nr 6, przykład skryptu a.html

Przy wyszukaniu adresu localhost:4321 zostaje wyświetlona strona z możliwościami przełączania między testami A, B lub C.