|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 **Информатика и вычислительная техника**

**Отчет**

**по домашней работе № 2**

**Дисциплина:** Разработка приложений на языке C#

**Название работы:** Разработка веб-сервера на языке C#



Студент гр. ИУ6-72Б **\_\_**16.12.2022**\_\_\_\_\_\_\_** И.С.Марчук

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.М. Минитаева

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Задание:**

Нужно реализовать http веб-сервер на языке C#, который будет выдавать html страницу.

Варианты: Нечетные по списку - Реализовать сервер на Thread per Request

**Выполнение работы:**

Код программы:

private int \_port { get; }

TcpListener \_listener;

public Server()

{

\_port = 8080;

}

public void Start()

{

\_listener = new TcpListener(IPAddress.Any, \_port);

try

{

\_listener.Start();

}

catch (Exception e)

{

throw;

}

while (true)

{

TcpClient Client = \_listener.AcceptTcpClient();

Thread Thread = new Thread(new ParameterizedThreadStart(WorkerThread));

Thread.Start(Client);

}

}

private static void WorkerThread(Object listener)

{

TcpClient Client = (TcpClient)listener;

int Count;

string Request = "";

byte[] Buffer = new byte[1024];

while ((Count = Client.GetStream().Read(Buffer, 0, Buffer.Length)) > 0)

{

Request += Encoding.ASCII.GetString(Buffer, 0, Count);

if (Request.IndexOf("\r\n\r\n") >= 0)

{

break;

}

}

string requestUri = parse(Client, Request);

Console.WriteLine(requestUri);

sendFile(Client, requestUri);

}

private static string parse(TcpClient Client, string request)

{

// Парсим строку запроса с использованием регулярных выражений

// При этом отсекаем все переменные GET-запроса

Match ReqMatch = Regex.Match(request, @"^\w+\s+([^\s\?]+)[^\s]\*\s+HTTP/.\*|");

// Если запрос не удался

if (ReqMatch == Match.Empty)

{

SendError(Client, 400);

return "";

}

string RequestUri = ReqMatch.Groups[1].Value;

RequestUri = Uri.UnescapeDataString(RequestUri);

RequestUri = RequestUri.Substring(1, RequestUri.Length - 1);

return RequestUri;

}

private static void sendFile(TcpClient Client, string FilePath)

{

if (!File.Exists(FilePath))

{

SendError(Client, 404);

return;

}

FileStream FS;

try

{

FS = new FileStream(FilePath, FileMode.Open, FileAccess.Read, FileShare.Read);

}

catch (Exception)

{

SendError(Client, 500);

return;

}

string Headers = "HTTP/1.1 200 OK\nContent-Type: text/html" + "\nContent-Length: " + FS.Length + "\n\n";

byte[] HeadersBuffer = Encoding.ASCII.GetBytes(Headers);

Client.GetStream().Write(HeadersBuffer, 0, HeadersBuffer.Length);

byte[] Buffer = new byte[1024];

while (FS.Position < FS.Length)

{

int Count = FS.Read(Buffer, 0, Buffer.Length);

Client.GetStream().Write(Buffer, 0, Count);

}

FS.Close();

Client.Close();

}

private static void SendError(TcpClient Client, int Code)

{

string CodeStr = Code.ToString() + " " + ((HttpStatusCode)Code).ToString();

string Html = "<html><body><h1>" + CodeStr + "</h1></body></html>";

string Str = "HTTP/1.1 " + CodeStr + "\nContent-type: text/html\nContent-Length:" + Html.Length.ToString() + "\n\n" + Html;

byte[] Buffer = Encoding.ASCII.GetBytes(Str);

Client.GetStream().Write(Buffer, 0, Buffer.Length);

Client.Close();

}

~Server()

{

if (\_listener != null)

{

\_listener.Stop();

}

}

}

**Вывод:** был реализован http веб-сервер на языке C#, который будет выдавать html страницу.