

**Курсовая работа**  
**по программному обеспечению компьютерных сетей**  
**для студентов, получающих второе высшее образование**  
**3 курс, 5 семестр**  
**Тема: "Разработка web-сервера"**

**Задание**

Разработать программу-сервер, способную принимать запросы по протоколу HTTP и выдавать соответствующие ответы. В качестве пользовательского интерфейса в программе используется Internet-браузер, например, Internet Explorer.

- 
1. Web-сервер для решения квадратного уравнения

---

  2. Web-сервер для вывода таблицы простых чисел, не превышающих заданного значения

---

  3. Web-сервер для сортировки строкового массива

---

  4. Web-сервер для разложения числа на простые множители

---

  5. Web-сервер для решения системы алгебраических уравнений

---

  6. Web-сервер для проверки орфографии введенного текста

---

  7. Web-сервер – тренажер для изучения английского языка

---

  8. Web-сервер для перевода английских слов

---

  9. Web-сервер – калькулятор

---

  10. Web-сервер – игра "крестики-нолики"

---

**Пример программы**

```
//  
// Web-сервер  
// Copyright (C) Oleg Strelnikov, 2003-2004
```

```

//

#include <winsock2.h>
#include <stdio.h>
#include <string.h>

const u_short port = 16000;

int main() {

printf("Web-Server\n");

// Загрузка библиотеки Windows Sockets 2.0
WSADATA wsaData;
if (WSAStartup(MAKEWORD(2, 0), &wsaData) != 0) {
printf("Can't startup Windows Sockets\n");
return 0;
}
printf("Windows Sockets started\n");

//создание сокета
SOCKET s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
if (s == INVALID_SOCKET) {
printf("Can't create socket\n");
return 0;
}
printf("Socket Created\n");

//адрес
sockaddr_in sai;
memset(&sai, 0, sizeof (sai));
sai.sin_family = AF_INET;
sai.sin_port = htons(port);
sai.sin_addr.s_addr = INADDR_ANY;

//связывание сокета с адресом
if (bind(s, (sockaddr*)&sai, sizeof (sai)) == SOCKET_ERROR) {
printf("Bind error\n");
return 0;
}
printf("Bind OK\n");

//создание очереди
if (listen(s, 1) == SOCKET_ERROR) {
printf("Listen error\n");
return 0;
}
printf("Listen OK\n");

while (true) {
//начало очередного сеанса связи

```

```

printf("\nAccepting...\n");
struct sockaddr_in saic;
int addrlen = sizeof(saic);
SOCKET sc = accept(s, (sockaddr *)&saic, &addrlen);
if(sc == INVALID_SOCKET) {
    printf("Accept error\n");
    break;
}
printf("Connection opened: %s\n", inet_ntoa(saic.sin_addr));
printf("Data exchanging...\n");

//получение запроса HTTP
char http_request[2000];
int received = recv(sc, http_request, sizeof(http_request), 0);

if(received == SOCKET_ERROR) {
    printf("\nSocket Error: receive\n");
    closesocket(sc);
    continue;
}
else if(received == 0) {
    printf("\nConnection closed by remote host\n");
    closesocket(sc);
    continue;
}
else {
    printf("\n\nHTTP request received:\n\n");
    fwrite(http_request, received, 1, stdout);
}

//проверяем, что метод запроса GET
if(strncmp(http_request, "GET ", 4)) {
    //Запрос не начинается с "GET "
    char http_response[] = "HTTP/1.0 200 Ok\r\nContent-type: text/html;\n";
    char* http_charset = "charset=windows1251\r\n\r\nДопустимы только запросы GET !";
    send(sc, http_response, sizeof(http_response), 0);
    closesocket(sc);
    continue;
}
//сразу после "GET " - URI
char* uri = http_request + 4;
//находим завершение строки запроса - " HTTP/1.0\r\n" или " HTTP/1.1\r\n"
char* http_ver = strstr(uri, " HTTP/1.0\r\n");
if(!http_ver) {
    http_ver = strstr(uri, " HTTP/1.1\r\n");
}
if(!http_ver) {
    //строка запроса не содержит версию HTTP 1.0 или 1.1
    char http_response[] = "HTTP/1.0 200 Ok\r\nContent-type: text/html;\n";
    char* http_charset = "charset=windows1251\r\n\r\nДопустимы только версии HTTP/1.0 и HTTP/1.1 !";
    send(sc, http_response, sizeof(http_response), 0);
}

```

```

    closesocket(sc);
    continue;
}

//завершаем строку uri двоичным нулём
http_ver[0] = 0;

//проверяем, содержит ли uri имя пользователя, то есть "?username=";
char* username = strstr(uri, "?username=");
if (username) {
    //содержит - выводим приветствие
    username += strlen("?username=");
    printf("username = %s", username);
    char http_response_begin[] = "HTTP/1.0 200 Ok\r\nContent-type: text/html;
charset=windows1251\r\n\r\nЗдравствуйте, ";
    send(sc, http_response_begin, sizeof (http_response_begin), 0);
    send(sc, username, strlen(username), 0);
    send(sc, "!", 2, 0);
    closesocket(sc);
    continue;
}
else {
    //не содержит - выводим форму для ввода имени
    char http_response[] = "HTTP/1.0 200 Ok\r\nContent-type: text/html;
charset=windows1251\r\n\r\n<FORM>Ваше имя: <INPUT type=\"text\"
name=\"username\"></FORM>";
    send(sc, http_response, sizeof (http_response), 0);
    closesocket(sc);
    printf("\n\nHTTP response sent:\n\n");
    fwrite(http_response, 1, sizeof (http_response), stdout);
    continue;
}

} //while

closesocket(s);

return 0;
}

```