



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 _____

КАФЕДРА _____

ОТЧЕТ

Отчет по лабораторной работе №3 по курсу «Компьютерные сети»

Вариант _____16_____

Студент ИУ7-73Б
(Группа)

(Подпись, дата)

Сукочева А.
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Рогозин Н.О.
(И.О.Фамилия)

2021 г.

Часть I

Разработать клиент-серверное приложение с использованием сетевых сокетов и протокола TCP для передачи бинарных файлов от сервера клиентам. Сервер должен позволять подключение нескольких клиентов. Имя файла задается клиентом при подключении в виде текстовой строки. Сервер должен отобразить, используя пользовательский интерфейс, названия всех файлов в папке с исполняемым файлом сервера, предоставив клиенту выбор.

Часть II

Разработать http сервер для обработки GET-запросов и предоставления статической информации клиенту. Разработать http-клиент для проверки данного сервера с помощью GET-запросов. Использовать системные сокет и транспортный протокол TCP. Язык C/C++. Возможно использование ранее разработанного TCP-сервера в качестве основы.

Использование фреймворков не допускается.

1) Наименьшее количество баллов выставляется за сервер с последовательной обработкой запросов, без выполнения доп. задачи

2) Наибольшее количество баллов выставляется за многопоточный сервер с обработкой соединений через использование динамически распределяемого пула потоков + выполнение доп. задачи

Вариант доп. задачи:

Сохранять статистику о различных форматах файлов, к которым обращался тот или иной пользователь.

Часть 1

Сервер работает следующим образом. При отправке строки ls возвращает содержимое директории с файлами. При отправке другой строки пытается найти файл с таким именем и если находит – отправляет файл и разрывает соединение.

Пример работы программы:

Клиент:

```
client [main] ⚡ ./main
FILES:
main
test.c
test
Makefile
compile_flags.txt

Enter file to download: test
Enter output file name: new_test
client [main] ⚡ ls
bin compile_flags.txt downloads main Makefile src
client [main] ⚡ cd downloads
downloads [main] ⚡ ls
new_test
downloads [main] ⚡ sudo chmod +x new_test
[sudo] пароль для lis:
downloads [main] ⚡ ls
new_test
downloads [main] ⚡ ./new_test
Hello!
downloads [main] ⚡
```

Сервер:

```
server [main] ⚡ ./main
Server is running on 0.0.0.0:3001
New connect from 127.0.0.1:12941
Filename: 'test'
^C
server [main] ⚡ ls
bin compile_flags.txt include main Makefile src test test.c
server [main] ⚡ ./test
Hello!
server [main] ⚡
```

Клиент указывает название желаемого файла и название файла, в который запишется содержимое. Как видно на приведенных рисунках, бинарный файл успешно передан и запущен.

Часть 2

Весь код с поясняющими комментариями приведен во вложениях.

Пример работы программы:

Содержимое папки. К этим файлам у клиента есть доступ.

```
part_2 [main] ⚡ ls
client headers index.html Makefile README.md server src statistics.txt tmp
part_2 [main] ⚡
```

Запускаем клиента. Будем выполнять GET-запросы.

Вводим название желаемого файла (index.html) и получаем следующий ответ.

```
part_2 [main] ⚡ ./client
Please, input file name for HTTP GET:
/index.html
HTTP/1.1 200 OK
Connection: close
Server: localhost
Content-Type: text/html
Content-Length: 6
hello
```

Пытаемся получить доступ к файлу, который находится в папке выше.
Доступ к запрошенному ресурсу запрещен.

```
part_2 [main] ⚡ ./client
Please, input file name for HTTP GET:
/../../
HTTP/1.1 403 Forbidden
Connection: close
Server: localhost
```

Пытаемся получить доступ к несуществующему файлу.

```
part_2 [main] ⚡ ./client
Please, input file name for HTTP GET:
sadadwdw
HTTP/1.1 404 Not Found
Connection: close
Server: localhost
```

Файл со статистикой.

Статистика представлена в следующем виде.

Формат файла (расширение).

Далее идет информации о пользователе и времени, когда он обращался к данному файлу.

```
md
127.0.0.1 data: Sun Dec  5 17:58:25 2021

txt
127.0.0.1 data: Sun Dec  5 17:58:08 2021
127.0.0.1 data: Sun Dec  5 17:58:13 2021
```