

УТВЕРЖДАЮ
Зав. кафедрой ИБСТ

к.т.н., доцент

С.Л.Зефилов

28 02 2024 г.

ЗАДАНИЕ
на курсовую работу

по теме: Разработка приложений с сетевым взаимодействием.

1 Дисциплина: Сетевые технологии.

2 Вариант задания: 8.

3 Студент: Гусев Д. А.

4 Группа: 21ПИ1.

5 Цель: разработка сетевого приложения с архитектурой клиент-сервер.

6 Исходные данные на проектирование:

— требования к сетевому взаимодействию:

— используемый протокол транспортного уровня – TCP;

— в протоколе прикладного уровня должны быть предусмотрены возможности обмена информацией о возникающих ошибках при сетевом взаимодействии;

— требования к программе-клиенту:

— программа-клиент должна обеспечивать передачу текстовых файлов на сервер если версия программы-сервера равна 1;

— программа-клиент должна обеспечивать передачу бинарных файлов на сервер если версия программы-сервера равна 2;

— в программе-клиенте должна быть реализована возможность определения версии программы-сервера;

— в программе-клиенте должна быть предусмотрена возможность регистрации нового пользователя в программе-сервере ;

— в программе-клиенте должна быть предусмотрена обработка ошибок, возникающих при сетевом взаимодействии;

— требования к программе-серверу:

— программа-сервер должна сохранять текстовый файл, полученный от программы клиента, если версия программы-сервера равна 1;

— программа-сервер должна сохранять бинарный файл, полученный от программы клиента, если версия программы-сервера равна 2;

— программа-сервер должна сообщать программе-клиенту об ошибке, если передаваемый файл не соответствует версии программы-сервера;

— в программе-сервере должна выполняться проверка IP-адреса программы-клиента – сохранение файла должно выполняться, если IP-адрес находится в диапазоне разрешенных адресов;

— в программе-сервере должна быть реализована возможность одновременного подключения к серверу нескольких программ-клиентов;

— в программе-сервере должна быть предусмотрена обработка ошибок, возникающих при сетевом взаимодействии;

— максимальное количество одновременно подключенных программ-клиентов – 3;

— при попытке подключения программы-клиента сверх максимального количества, ей должно возвращаться сообщение об ошибке подключения;

— должна быть реализована аутентификация пользователя программы-клиента при подключении к серверу;

— должен быть разработан протокол прикладного уровня, предназначенный для реализации сетевого взаимодействия между программой-клиентом и программой-сервером;

— должна быть проведена проверка работоспособности программ;

— должно быть разработано руководство пользователя.

7 Структура проекта:

7.1 Пояснительная записка (содержание работы):

— разработка протокола прикладного уровня;

— программная реализация клиента и сервера;

— проверка работоспособности разработанных программ;

— разработка руководства пользователя.

7.2 Графическая часть:

UML-диаграммы протокола прикладного уровня.

7.3 Экспериментальная часть:

Тестирование разработанных программ.

8 Календарный план выполнения работы:

— оформление задания	к 2 неделе семестра;
— разработка протокола прикладного уровня	к 5 неделе семестра;
— программная реализация клиента и сервера	к 11 неделе семестра;
— проверка работоспособности программ	к 13 неделе семестра;
— разработка руководства пользователя	к 14 неделе семестра;
— оформление отчета	к 15 неделе семестра.

Задание получил «8» февраля 2024 г.

Студент

Руководитель

Д. А. Гусев

О. В. Липилин

Тестирование программы

1) Было составлено описание функций, выполняемых клиентом.

1.1) Подключение (connect):

- Описание: Клиент пытается подключиться к серверу. Он выбирает метод (аутентификация или регистрация) и версию протокола. Если сервер подтверждает успешное подключение, клиент переходит к следующему этапу (аутентификация или регистрация).

- Вводимые данные: метод (auth или reg) и версия (1 или 2).

- Сообщение протокола: CONN_SUC для успешного подключения, CONN_ERR для ошибки подключения.

1.2) Аутентификация (auth):

- Описание: Клиент отправляет запрос на аутентификацию, вводя логин и пароль. Если сервер подтверждает успешную аутентификацию, клиент переходит к следующему этапу (загрузка файла).

- Вводимые данные: логин и пароль.

- Сообщение протокола: AUTH_SUC для успешной аутентификации, AUTH_ERR для неверного логина или пароля.

1.3) Регистрация (reg):

- Описание: Клиент отправляет запрос на регистрацию, вводя логин и пароль. Если сервер подтверждает успешную регистрацию, клиент переходит к следующему этапу (загрузка файла).

- Вводимые данные: логин и пароль.

Сообщение протокола: REG_SUC для успешной регистрации, REG_ERR если пользователь уже существует.

1.4) Загрузка файла (upload):

- Описание: Клиент загружает файл на сервер. Он вводит путь к файлу, который затем отправляется на сервер. Если сервер подтверждает успешную передачу файла, процесс завершается.

- Вводимые данные: путь к файлу.

- Сообщение протокола: UP_SUC для успешной передачи файла, UP_ERR для неверного типа файла.

2) Было составлено описание функций, выполняемых сервером.

2.1) Подключение (connect):

- Описание: Сервер обрабатывает запрос на подключение от клиента. Он проверяет заголовок, метод и версию протокола. Если все проверки проходят успешно, сервер подтверждает подключение и переходит к следующему состоянию (аутентификация или регистрация).

- Вводимые данные: заголовок, метод и версия протокола.

- Сообщение протокола: CONN_SUC для успешного подключения, CONN_ERR для ошибки подключения, REQ_ERR для неверного запроса.

2.2) Аутентификация (auth):

- Описание: Сервер обрабатывает запрос на аутентификацию от клиента. Он проверяет заголовок и валидность логина и пароля. Если все проверки проходят успешно, сервер подтверждает аутентификацию и переходит к следующему состоянию (загрузка файла).

- Вводимые данные: заголовок, логин и пароль.

- Сообщение протокола: AUTH_SUC для успешной аутентификации, AUTH_ERR для неверного логина или пароля, REQ_ERR для неверного запроса.

2.3) Регистрация (reg):

- Описание: Сервер обрабатывает запрос на регистрацию от клиента. Он проверяет заголовок и валидность логина и пароля. Если все проверки проходят успешно, сервер подтверждает регистрацию и переходит к следующему состоянию (загрузка файла).

- Вводимые данные: заголовок, логин и пароль.

- Сообщение протокола: REG_SUC для успешной регистрации, REG_ERR если пользователь уже существует, REQ_ERR для неверного запроса.

Загрузка файла (upload):

Он проверяет заголовок, имя файла и содержимое файла. Если все проверки проходят успешно, сервер сохраняет файл и подтверждает успешную загрузку.

- Сообщение протокола: UP_SUC для успешной загрузки файла, UP_ERR для неверного типа файла, REQ_ERR для неверного запроса.

3.1) Описание взаимодействия клиента и сервера при успешной регистрации:

- Регистрация (reg): Клиент отправляет запрос на регистрацию с именем пользователя (user1) и паролем (pass1). Сервер подтверждает успешную регистрацию.

- Загрузка файла (upload): Клиент загружает файл (file.jpg) на сервер. Это бинарные данные, которые представляют собой изображение в формате JPEG. Сервер подтверждает успешную передачу файла.

(#?8#N??w.q3???S?L??e?Y??#?'m@"? ?#?#e??P??g?#P?& ?

```
#B#K@d{?-J+##+ك#<##BL#@end
upload@sepуспешная передача файла@end
```

3.2) Описание взаимодействия клиента и сервера при успешной аутентификации:

- Подключение (connect): Клиент (Python3 Client Win64) подключается к серверу. Сервер подтверждает успешное подключение.

- Аутентификация (auth): Клиент отправляет запрос на аутентификацию с именем пользователя (user) и паролем (password). Сервер подтверждает успешную аутентификацию.

- Загрузка файла (upload): Клиент загружает текстовый файл (file.txt) на сервер. Текст файла: "Eat some more of these soft French rolls, and drink some tea". Сервер подтверждает успешную передачу файла.

TCP поток представлен ниже:

```
connect@sepPython3 Client Win64@sepauth@sep1@end
connect@sepуспешное подключение@end
auth@sepuser@seppassword@end
auth@sepуспешная аутентификация@end
upload@sepfile.txt@sepEat some more of these soft French rolls, and
drink some tea@end
upload@sepуспешная передача файла@end
```

3.3) Описание взаимодействия клиента и сервера при Ошибка подключения (сервер переполнен):

- Подключение (connect): Клиент (Python3 Client Win64) пытается подключиться к серверу. Однако сервер сообщает об ошибке подключения.

TCP поток представлен ниже:

```
connect@sepPython3 Client Win64@sepauth@sep1@end
connect@seпошибка подключения@end
```

3.4) Описание взаимодействия клиента и сервера при неверных запросах при подключении, ошибке при регистрации и ошибке неверного типа файла:

- Подключение (connect): Клиент (Python3 Client Win64) пытается подключиться к серверу с запросом на помощь, но сервер сообщает о неверном запросе. Затем клиент повторно пытается подключиться к серверу с запросом на аутентификацию, и сервер подтверждает успешное подключение.

- Аутентификация (auth): Клиент отправляет запрос на аутентификацию с именем пользователя (user2) и паролем (pass2), но сервер сообщает о неверном логине или пароле. Затем клиент повторно отправляет запрос на аутентификацию с другим именем пользователя (user) и паролем (password), и сервер подтверждает успешную аутентификацию.

- Загрузка файла (upload): Клиент пытается загрузить файл (file.jpg) на сервер, но сервер сообщает о неверном типе файла. Затем клиент загружает текстовый файл (file.txt) на сервер с текстом "Eat some more of these soft French rolls, and drink some tea", и сервер подтверждает успешную передачу файла.

TCP поток представлен ниже:

```
connect@sepPython3 Client Win64@sephelp@sep3@end
connect@sepневерный запрос@end
connect@sepPython3 Client Win64@sepauth@sep1@end
connect@sepуспешное подключение@end
auth@sepuser2@seppass2@end
auth@sepневерный логин или пароль@end
auth@sepuser@seppassword@end
auth@sepуспешная аутентификация@end
upload@sepfile.jpg@sep##### #JFIF ### # # C #####
#####2!#####,.$2I@LKG@FEPZsbPUmVEFd{N` }s~|
C#####;!!;|SFS|##### ##
# ###" ##### # ##### $# ##### # ## #####!
A#1#2B"Qq##### ### ## #####12AQ
## ##### ? ## dÜ"Im#[#C]z2>i
_@u.;orYT+Z g0skl*3`VmYr}e
%Z@B#x<[óY#83cfG#WI';d_#C#Sj<V#n#0#6
```

```
(#8Nw.q3SLeY# 'm" #eP#g#P& ?  
#B#K@d{?-J+##+?ك#<##BL#@end
```

```
upload@серневерный тип файла@end
```

```
upload@sepfile.txt@sepEat some more of these soft French rolls, and  
drink some tea@end
```

```
upload@серуспешная передача файла@end
```

3.5) Описание взаимодействия клиента и сервера при неверных запросах при подключении, ошибке при аутентификации и ошибке неверного типа файла:

- Подключение (connect): Клиент (Python3 Client Win64) пытается подключиться к серверу с запросом на помощь, но сервер сообщает о неверном запросе. Затем клиент повторно пытается подключиться к серверу с запросом на аутентификацию, и сервер подтверждает успешное подключение.

- Аутентификация (auth): Клиент отправляет запрос на аутентификацию с именем пользователя (user2) и паролем (pass2), но сервер сообщает о неверном логине или пароле. Затем клиент повторно отправляет запрос на аутентификацию с другим именем пользователя (user) и паролем (password), и сервер подтверждает успешную аутентификацию.

- Загрузка файла (upload): Клиент пытается загрузить файл (file.jpg) на сервер, но сервер сообщает о неверном типе файла. Затем клиент загружает текстовый файл (file.txt) на сервер с текстом “Eat some more of these soft French rolls, and drink some tea”, и сервер подтверждает успешную передачу файла.

ТСР поток представлен ниже:

```
connect@sepPython3 Client Win64@sephelp@sep3@end
```

```
connect@серневерный запрос@end
```

```
connect@sepPython3 Client Win64@sepauth@sep1@end
```

```
connect@серуспешное подключение@end
```

```
auth@sepuser2@seppass2@end
```

```
auth@серневерный логин или пароль@end
```

```
auth@sepuser@seppassword@end
```

```
auth@серуспешная аутентификация@end
```

```
upload@sepfile.jpg@sep##### #JFIF ### # # C #####
```


#####2!####=,.\$2I@LKG@FEPZsbPUmVEFd{N` }s~|
C#####;!!;|SFS|#####
###" ##### # ##### \$# ##### # ## ####!
A#1#2B"Qq ##### ### ## ## #####12AQ
? ## dÜ" #I m#[##C] z2> i
_u.;orY T+Z g0#s k1*3`V mY r} e
%Z@B#x<[ó Y#83cfG#WI'; d,C#Sj<V#n#0#6
(#8#N w.q3 S L e Y # 'm@" # #e P g #P & ?
#B# K@ d{?-J+ ###+ ك# <##BL#@end

upload@серневерный тип файла@end

upload@sepfile.txt@sepEat some more of these soft French rolls, and
drink some tea@end

upload@серуспешная передача файла@end