

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника" профиль "Программное обеспечение
средств вычислительной техники и автоматизированных систем"

Кафедра вычислительных систем (ВС)

**Курсовая работа по дисциплине
Сетевое программирование**

**Разработка сетевого приложения «Разработка сетевого приложения. Клиент на
базе протокола FTP в пассивном режиме».**

Выполнил:

студент гр. ИП – 113

Силантьев Александр Сергеевич

ФИО студента

«20» мая 2024г.

Проверил старший преподаватель кафедры ВС

Ревун Артем Леонидович

ФИО преподавателя

20 мая 2024г.

Оценка _____

Новосибирск 2024 г.

Содержание

Постановка задачи	3
Теоретические сведения	4
Описание реализации	5
Сервер	5
Клиент	5-6
Результаты работы программы	7
Листинг программы	11
Список источников	13

Постановка задачи

- Необходимо разработать клиентское приложение FTP, которое будет подключаться к серверу FTP в пассивном режиме
- Для выполнения задачи понадобится использовать сокеты для установления соединения с сервером FTP, отправка команд и прием ответов. Также потребуется обработка ответов сервера, парсинг информации и управление файлами.

Задачи:

1. Аутентификация:

- Клиент должен быть способен отправить логин и пароль на сервер для аутентификации.

2. Получение списка файлов:

- Клиент должен иметь возможность получить список файлов в текущем каталоге на сервере и отобразить его на консоли.

3. Загрузка файла:

- Клиент должен иметь возможность скачать выбранный файл с сервера на локальную машину.

4. Загрузка файла на сервер:

- Клиент должен иметь возможность загрузить выбранный локальный файл на сервер.

5. Рабочий путь:

- Клиент должен сохранять текущий рабочий путь, чтобы иметь возможность переходить между каталогами на сервере.

Теоретические сведения

Следует знать теоретические сведения о работе протокола FTP, IntelliJ IDEA как среда разработки, язык Java, транспортный протокол TCP

1. TCP (Transmission Control Protocol):

- TCP является одним из основных протоколов передачи данных в компьютерных сетях. В контексте микросервисов, его использование обеспечит надежное установление соединения между клиентом и сервером для обмена данными.

2. IntelliJ IDEA:

- Интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.

3. Java:

- Строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle)..

4. FTP (File Transfer Protocol):

— это базовый протокол передачи файлов, основанный на отношениях «клиент-сервер». Иначе говоря, пользователь, используя специальную программу — FTP-клиент, по этому протоколу может подключиться к серверу — компьютеру в сети, на котором также установлено специальное ПО — FTP-сервер, чтобы производить различные действия с файлами.

Описание реализации

Сервер

1. **Класс** `FTPServer`:
 - Этот класс содержит метод `main`, который является точкой входа в программу.
 - Создается серверный сокет на порту 4000.
 - В бесконечном цикле ожидается подключение клиентов. Как только клиент подключается, создается новый экземпляр класса `ftp1` для обработки соединения с клиентом.
2. **Класс** `ftp1`:
 - Этот класс наследует `Thread`, что позволяет обрабатывать каждое соединение с клиентом в отдельном потоке.
 - Конструктор `ftp1` принимает сокет клиента, инициализирует входной и выходной потоки для обмена данными с клиентом.
 - Метод `SendFile()` отправляет файл клиенту. Он сначала считывает имя файла от клиента, проверяет его наличие. Если файл существует, отправляет сообщение "READY" и затем посылаются данные файла по байтам.
 - Метод `ReceiveFile()` принимает файл от клиента. Он сначала считывает имя файла от клиента и проверяет его наличие. Если файл уже существует на сервере, запрашивается у клиента решение о перезаписи файла. Затем происходит прием данных файла и сохранение его на сервере.
 - Метод `BrowseDir()` отображает содержимое указанной директории клиенту.
 - Метод `verify()` аутентифицирует пользователя. Он считывает имя пользователя и пароль, сравнивает их с данными из файла `users.txt`. Если аутентификация прошла успешно, сообщает об этом клиенту.
 - Метод `run()` является точкой входа для обработки соединения с клиентом. Он считывает команды от клиента и вызывает соответствующие методы для их обработки.

Клиент

1. **Класс** `FTPClient`:
 - Этот класс содержит метод `main`, который является точкой входа в программу.
 - Создается сокет для подключения к серверу FTP, который находится на локальном хосте и слушает порт 4000.
 - Создается экземпляр класса `ftp`, который затем вызывает метод `displayMenu()` для отображения главного меню клиента.
2. **Класс** `ftp`:
 - Этот класс отвечает за взаимодействие с сервером FTP.
 - Конструктор `ftp` инициализирует входной и выходной потоки для обмена данными с сервером, а также объект для чтения с клавиатуры.
 - Метод `chooseFile()` позволяет пользователю выбрать файл для отправки на сервер. Он отображает содержимое текущей директории и спрашивает у пользователя выбор файла или переход в поддиректорию.
 - Метод `SendFile()` отправляет выбранный файл на сервер. Перед отправкой он проверяет наличие файла и, если файл существует, спрашивает пользователя, хочет ли он перезаписать файл на сервере.
 - Метод `ReceiveFile()` позволяет клиенту получить файл с сервера. Пользователь указывает имя файла для загрузки, и затем файл с сервера передается и сохраняется на локальной машине.
 - Метод `BrowseDirectory()` позволяет клиенту просматривать содержимое текущей директории на сервере.
 - Метод `BackDirectory()` позволяет клиенту вернуться на уровень выше в иерархии директорий.

- Метод `displayMenu()` отображает главное меню клиента, где он может выбрать действия: отправить файл на сервер, получить файл с сервера, просмотреть содержимое текущей директории на сервере, вернуться на уровень выше или выйти из программы.

В общем, код реализует базовый функционал FTP-клиента, который позволяет пользователю взаимодействовать с сервером для передачи файлов и управления директориями.

Результаты работы программы

```
C:\Users\Kento\.jdk\corretto-17.0.11\bin\java.exe "-javaagent:C:\P
FTP Server Started on Port Number 5217
Waiting for Connection ...
FTP Client Connected ...
Waiting for Connection ...
```

1. Интерфейс сервера

```
C:\Users\Kento\.jdk\corretto-17.0.11\bin\java.exe "-javaagent:C:\V
Enter Username:
Alsciel
Enter Password:
23
login successful
pwd:
[ MENU ]
1. Send File
2. Receive File
3. Browse
4. Back
5. Exit

Enter Choice :|
```

2. Интерфейс клиента, пример аутентификации.

```

Enter Choice :3
Enter Directory Name:D:\
Directories:
1:$RECYCLE.BIN
2:BlueStacks
3:Client
4:Coromon
5:Darksiders Warmastered Edition [2016](RePack =nemos=)
6:Dave the Diver (2023)
7:EN.LPROJ
8:Genshin Impact game
9:GTK1
10:Java
11:labs
12:Lethal Company v40 by Pioneer
13:Nadir
14:QtProjects
15:Redist
16:Riot Games
17:Risk of Rain 2 v1.2.4.4 by Pioneer
18:Serv
19:SteamLibrary
20:System Volume Information
21:Unity
22:Virtual
23:[R.G. Mechanics] Need for Speed - Most Wanted

Files
1:19045.3570.230929-1700.22H2_RELEASE_SVC_PROD1_CLIENTMULTI_X64FRE_RU-RU.ISO
2:DumpStack.log
3:DumpStack.log.tmp
4:ru_windows_7_64bit_Ultimate.iso
5:UnzipErrorLog.Bytes

pwd: D:\

```

```

pwd: D:\
[ MENU ]
1. Send File
2. Receive File
3. Browse
4. Back
5. Exit

Enter Choice :3
Enter Directory Name:Serv
Directories:

Files
1:245.docx

pwd: D:\Serv\
[ MENU ]
1. Send File
2. Receive File
3. Browse
4. Back
5. Exit

Enter Choice :

```

3. Демонстрация перемещения по файловой системе


```

D:\
Directories:
1:$RECYCLE.BIN
2:BlueStacks
3:Client
4:Coromon
5:Darksiders Warmastered Edition [2016](RePack =nemos=)
6:Dave the Diver (2023)
7:EN.LPROJ
8:Genshin Impact game
9:GTK1
10:Java
11:labs
12:Lethal Company v40 by Pioneer
13:Nadir
14:QtProjects
15:Redist
16:Riot Games
17:Risk of Rain 2 v1.2.4.4 by Pioneer
18:Serv
19:SteamLibrary
20:System Volume Information
21:Unity
22:Virtual
23:[R.G. Mechanics] Need for Speed - Most Wanted

Files
1:19045.3570.230929-1700.22H2_RELEASE_SVC_PROD1_CLIENTMULTI_X64FRE_RU-RU.ISO
2:DumpStack.log
3:DumpStack.log.tmp
4:ru_windows_7_64bit_ultimate.iso
5:UnzipErrorLog.Bytes

1.Select file
2.Enter Subdirectory

```

```

1.Select file
2.Enter Subdirectory
2
Subdirectory: Serv
Directories:

```

```

Files
1:245.docx

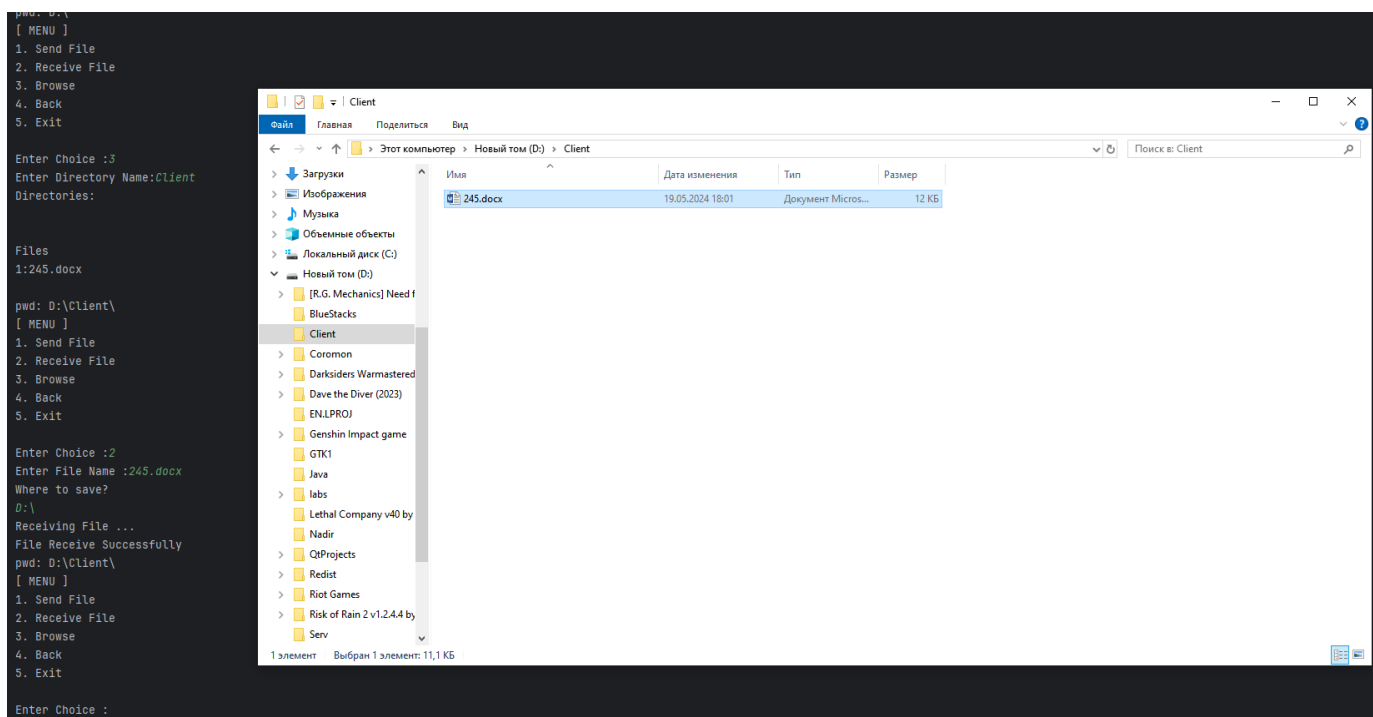
```

```

1.Select file
2.Enter Subdirectory
1
File: 245.docx
Where to save?--filename with extension: Client
File Already Exists. Want to OverWrite (Y/N) ?
Y
Sending File ...

```

4. Отправка файла в директорию



5. Получение файла с сервера на клиент

Листинг программы

FTPServer.java

```
public class FTPServer
{
    public static void main(String args[]) throws Exception
    {
        ServerSocket socfd=new ServerSocket(4000);
        System.out.println("FTP Server Started on Port Number 5217");
        while(true)
        {
            System.out.println("Waiting for Connection ...");
            ftp1 t=new ftp1(socfd.accept());
        }
    }
}

class ftp1 extends Thread
{
    Socket ClientSoc;
    DataInputStream din;
    DataOutputStream dout;
    ftp1(Socket soc)
    {
        try
        {
            ClientSoc=soc;
            din=new DataInputStream(ClientSoc.getInputStream());
            dout=new DataOutputStream(ClientSoc.getOutputStream());
            System.out.println("FTP Client Connected ...");
            start();
        }
        catch(Exception ex)
        {
        }
    }
}
```

...

FTPClient.java

```
public void displayMenu() throws Exception
{
    while(true) {
        System.out.println("Enter Username:");
        String usr=br.readLine();
        dout.writeUTF(usr);
        System.out.println("Enter Password:");
        String pass=br.readLine();
        dout.writeUTF(pass);
        String reply = din.readUTF();
        System.out.println(reply);
        if(reply.equalsIgnoreCase("login successful")) break;
    }
    while(true)
    {
        System.out.println("pwd: "+dir+"\n[ MENU ]");
        System.out.println("1. Send File");
        System.out.println("2. Receive File");
        System.out.println("3. Browse");
        System.out.println("4. Back");
    }
}
```

```

System.out.println("5. Exit");
System.out.print("\nEnter Choice :");
int choice;
choice=Integer.parseInt(br.readLine());
if(choice==1)
{
    dout.writeUTF("SEND");
    SendFile();
}
else if(choice==2)
{
    dout.writeUTF("RECEIVE");
    ReceiveFile();
}
else if(choice==3)
{
    dout.writeUTF("LIST");
    BrowseDirectory();
}
else if(choice==4)
{
    dout.writeUTF("LIST");
    BackDirectory();
}
else
{
    dout.writeUTF("DISCONNECT");
    System.exit(1);
}
}
}
}

```

users.txt

	FTPServer.java	FTPClient.java	users.txt
1			Alsciel
2			23
3			Emon
4			147
5			Kento
6			sifat420
7			

Полный листинг:

https://github.com/Alsciel/Kurs_Set.prog.git

Список источников

1. Merrow. Java Network Programming: A Complete Guide to Network Programming and Network Technologies Using Java. - ISBN: 978-0201419759.
2. Introduction to FTP // GeeksforGeeks. - URL: <https://www.geeksforgeeks.org/introduction-to-ftp/>.
3. FTP Tutorial // Tutorialspoint. - URL: <https://www.tutorialspoint.com/ftp/>.
4. How to Create a Simple FTP Client and Server in Java // Baeldung. - URL: <https://www.baeldung.com/java-ftp-client-server>.
5. Java FTP Client Example // JournalDev. - URL: <https://www.journaldev.com/673/java-ftp-client-example#java-ftp-upload>.
6. How to Implement FTP in Java // Edureka. - URL: <https://www.edureka.co/blog/how-to-implement-ftp-in-java/>.