

Java大作业文档

1. 实验目的
2. 实验要求
 - 2.1. 服务器端功能要求：
 - 2.2. 客户端功能要求：
3. 实验环境描述
4. 实验设计思路
 - 4.1. 服务器端：
 - 4.2. 客户端
5. 代码结构
 - 5.1. 服务器端
 - 5.2. 客户端
6. 简要系统使用说明
 - 6.1. 配置文件说明
 - 6.2. 用户登录说明
 - 6.3. 用户操作说明
 - 6.3.1. dir指令说明
 - 6.3.2. cd指令说明
 - 6.3.3. get指令说明
 - 6.3.4. put指令说明
 - 6.3.5. exit指令说明
 - 6.4. 日志记录说明
7. 实验总结

学院	班级	学号	姓名
计算机学院（国家示范性软件学院）	2021211321	2021211016	戴鑫旺

1. 实验目的

设计并实现一个基于字符界面的C/S结构的文件传输程序。

2. 实验要求

2.1. 服务器端功能要求：

- 1、启动时首先读入一个配置文件，配置文件中要保存全部用户（至少10个）的用户名及密码（明文保存，真实系统不会明文保存的）；系统所使用的目录（就是上传、下载文件的根目录，网络用户只能访问这个目录及其子目录）。
- 2、在固定端口（8000以上）侦听，等待客户端连接；
- 3、客户端连接成功后有一个验证用户名和密码的过程。如果用户名密码正确，要给该客户端发成功验证的提示信息，否则给客户端发送用户名或密码错误信息，并等待用户再次验证，直到验证成功或客户端断开；系统还可以接受一个匿名用户，该用户用户名为anonymous，该用户登录时不需要密码，但它与普通用户相比，缺少上传文件的权限，也就是说不允许它上传文件。
- 4、至少支持5个客户端同时访问；
- 5、响应客户请求，并根据客户端的命令给出相应回应；
- 6、日志功能。用文本文件记录每个用户的行为。这些行为包括每次登录成功信息（要包含IP地址，时间等）、每次登录失败信息（要包含IP地址、时间等）、登录成功后执行的所有操作等。

2.2. 客户端功能要求：

- 1、启动后首先要连接到服务器端；
- 2、连接成功后提示用户输入用户名及密码，并将用户输入的用户名及密码发给服务器端验证。验证成功后则成功登录，否则根据服务器端反馈提示用户重新输入用户名及密码，直到验证成功或用户结束程序；如果是匿名用户，则无需用户输入密码，直接登录；
- 3、成功登录后要有提示符；
- 4、成功登录后在提示符下用户可以通过命令完成相应操作，服务器可以接受的命令有如下5个：①dir：列出当前目录下所有文件（包括子目录）；② cd 目录名：进入名字为“目录名”的子目录，其中cd ..表示回到上一级目录；③put 文件名：把本地文件上传到服务器端当前目录；④ get 文件名：将服务器上的文件下载到本地当前目录；这里的“文件名”同前边的“目名”一样，都是命令的参数，参数与命令之间有

一个空格。这些参数的具体值要根据实际情况处理，如果文件或目录不存在要给用户提示；⑤exit：结束客户端程序。

3. 实验环境描述

- Java软件开发工具：IntelliJ IDEA Ultimate 2023.1.2
- JDK版本：Oracle OpenJDK version 20.0.01
- 操作系统：Windows 11

4. 实验设计思路

4.1. 服务器端：

1. 实现ConfigReader类，用于读取服务器配置文件中保存的用户信息、目录等数据，并将它们存储在相应的变量中，以便其他类能够访问这些数据。
2. 实现Authenticator类，用于验证客户端发送的用户名和密码是否正确。该类应该访问ConfigReader类中的数据进行验证，并向客户端发送响应消息。
3. 实现Logger类，用于记录每个用户的行为。该类应该能够记录每次登录成功信息（包含IP地址、时间等）、每次登录失败信息（包含IP地址、时间等）和登录成功后执行的所有操作等。记录数据时，可以将其写入文本文件中。
4. 实现Command类，用于解析客户端发送的命令并确定要执行的操作。应支持dir、cd、put、get和exit命令。
5. 实现CommandProcessor类，用于执行客户端发送的命令并返回响应消息。该类应该调用相应的功能类来执行命令，例如上传或下载文件。在处理命令时，还应检查用户是否具有执行该命令所需的权限。
6. 实现User类，用于存储与每个用户相关的数据，如用户名、密码、当前登陆状态等。
7. 实现ClientThread类，在服务器端上为每个连接的客户端创建一个线程。该线程应该接收客户端发送的命令并将它们传递给CommandProcessor类进行处理。
8. 在服务器类中，启动时应创建一个ServerSocket对象并侦听指定端口，等待客户端连接。当有客户端连接时，创建一个ClientThread对象来处理与该客户端的通信。

4.2. 客户端

1. 在客户端类中，启动时应先尝试连接到服务器。如果连接成功，则提示用户输入用户名和密码，并

将它们发送到服务器进行验证。如果验证成功，则提示符出现，并等待用户输入命令。如果验证失败，则要求用户重新输入用户名和密码。该验证过程由Authenticator类实现。

2. 当客户端发送命令时，将其发送到服务器端并等待响应。一旦收到响应，客户端应将其打印到屏幕上，并显示新的提示符，等待用户输入下一个命令。该过程由CommandPrompt类实现

5. 代码结构

5.1. 服务器端

类名	Authenticator	ClientThread	CommandProcessor
作用	用户登录验证类,用于验证用户登录的用户名和密码	客户端线程类，用于处理客户端的请求	命令处理器类，用于解析并处理用户输入的命令
成员变量	<ul style="list-style-type: none">• users: List<User>类型，当前系统中已注册的用户列表• in: BufferedReader类型，客户端输入流• out: BufferedWriter类型，客户端输出流• ip: String类型，客户端的IP地址• log: Logger类型，日志记录器• socket: Socket类型，客户端连接的Socket	<ul style="list-style-type: none">• socket: Socket类型，客户端连接的Socket对象• users: List<User>类型，当前系统中已注册的用户列表• rootDir: String类型，根目录路径	<ul style="list-style-type: none">• in: BufferedReader类型，从客户端读取数据的缓冲字符输入流• out: BufferedWriter类型，向客户端发送数据的缓冲字符输出流• currentPath: Path类型，当前工作目录的路径• ip: String类型，客户端IP地址• log: Logger类型，日志记录器• user: User类型，当前登录的用户• rootDir: String类型，FTP服务器根目录的路径• socket: Socket类型，连接到客户端的Socket

成员方法	<ul style="list-style-type: none"> 构造函数 Authenticator(-)：初始化用户认证器的相关属性 run()：用户登录验证过程，返回登录成功的用户对象 	<ul style="list-style-type: none"> 构造函数 ClientThread(-)：初始化客户端线程的相关属性 方法 run()：执行客户端请求处理逻辑，包括用户登录验证和命令提示符状态下的命令处理，可能会抛出 IOException异常 	<ul style="list-style-type: none"> 构造函数 CommandProcessor(-)：初始化相关属性 方法 parseCommand(String input)：解析命令，返回Command对象 方法 dir()：处理dir命令，列出当前目录下的文件和文件夹 方法 cd(String arg)：处理cd命令，切换目录 方法 put(String arg)：处理put命令，上传文件到服务器 方法 get(String arg)：处理get命令，下载服务器上的文件到客户端 方法 exit()：处理exit命令，退出服务器 方法 process()：处理用户输入的命令，根据命令调用相应的处理方法
类名	ConfigReader	Logger	Server
作用	读取配置文件类，用于从文件中读取配置信息，包括用户信息和根目录路径	日志记录器，用于记录操作信息和时间戳到指定文件中	服务器类，用于启动服务器并监听客户端连接请求
成员变量	path：String类型，配置文件的路径	<ul style="list-style-type: none"> LOG_FILE_NAME：String类型，日志文件名 DATE_FORMAT：SimpleDateFormat类型，时间格式 fileWriter：FileWriter类型，日志文件写入流 	<ul style="list-style-type: none"> DEFAULT_PORT：int类型，默认端口号 MAX_CLIENTS：int类型，最大客户端连接数 users：List<User>类型，用户列表 rootDir：String类型，服务器根目录路径

成员函数	<ul style="list-style-type: none"> 方法 getUsers(): 从配置文件中读取用户信息并存储到List中, 返回包含用户信息的List。可能会抛出IOException异常。 方法 getRootDir(): 从配置文件中读取根目录路径并返回绝对路径。 	<ul style="list-style-type: none"> 构造函数 Logger(-): 创建一个新的日志文件写入流。可能会抛出IOException异常。 方法 log(String ip, String message): 记录日志方法, 包含IP地址、时间和操作信息, 并将其写入日志文件中。 方法 close(): 关闭日志记录器, 关闭文件写入流。 	<ul style="list-style-type: none"> 构造函数 Server(-): 从配置文件中读取用户列表和根目录路径, 初始化相关属性。 方法 start(): 启动服务器并监听客户端连接请求 方法 main(String[] args): 服务器的入口方法, 创建服务器对象并启动服务器
------	---	--	--

5.2. 客户端

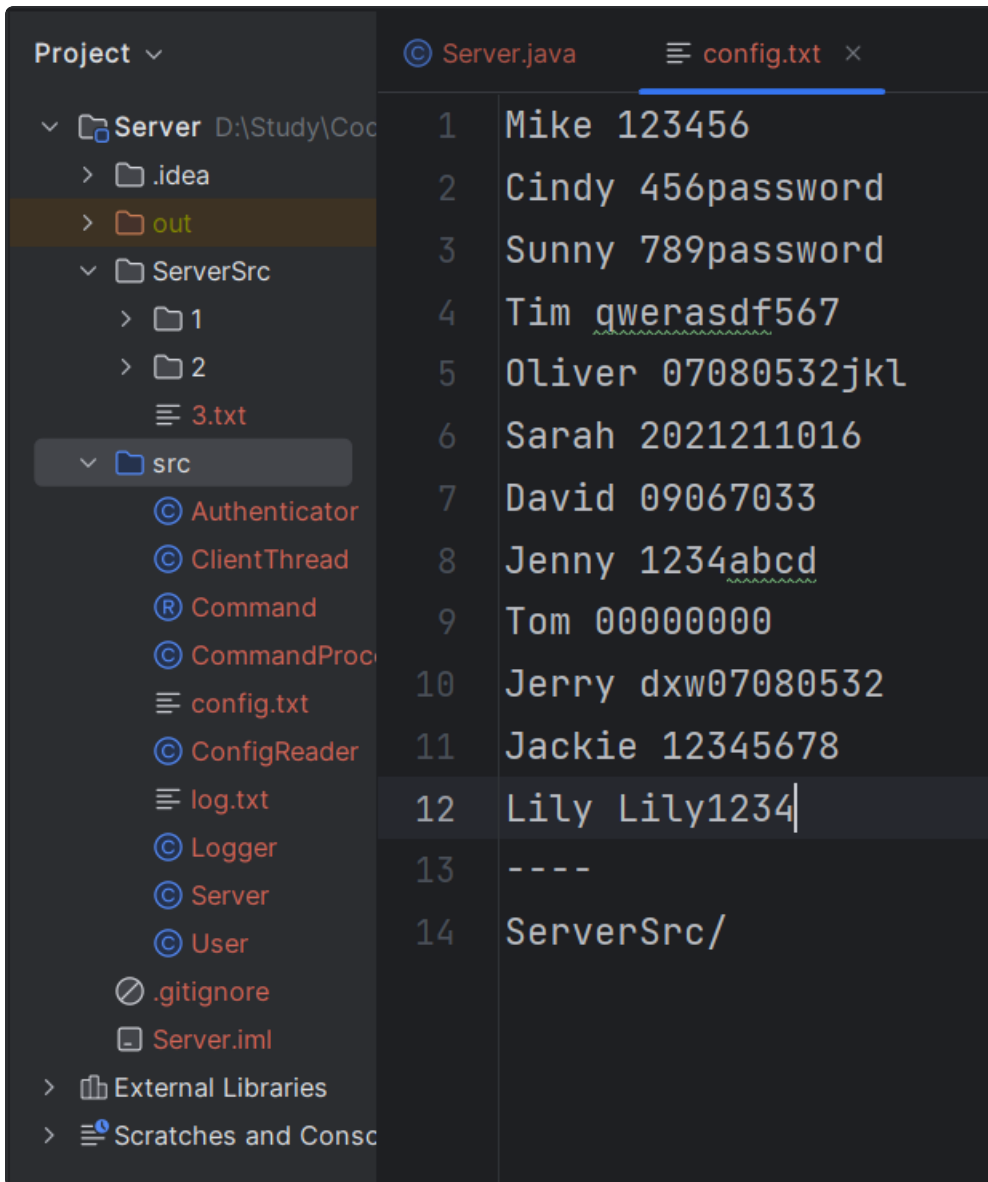
类名	Client	Authenticator	CommandPrompt
作用	客户端类, 用于与服务器建立连接并进行交互	用户身份验证类, 用于验证用户的用户名和密码是否正确	命令提示符类, 用于处理用户的命令请求并与服务器进行交互

成员变量	<ul style="list-style-type: none"> • host: String类型, 服务器的IP地址 • port: int类型, 服务器监听的端口号 • rootDir: String类型, 客户端的资源目录路径 	<ul style="list-style-type: none"> • in: BufferedReader类型, 输入流, 从服务器接收数据 • out: BufferedWriter类型, 输出流, 向服务器发送数据 • consoleIn: BufferedReader类型, 控制台输入流, 用于获取用户输入的用户名和密码 	<ul style="list-style-type: none"> • in: BufferedReader类型, 输入流, 从服务器接收数据 • out: BufferedWriter类型, 输出流, 向服务器发送数据 • consoleIn: BufferedReader类型, 控制台输入流, 用于获取用户输入的命令 • rootDir: String类型, 客户端资源文件夹的路径
成员方法	<ul style="list-style-type: none"> • 构造函数 Client(String host, int port): 初始化客户端对象的相关属性 • start(): 启动客户端, 与服务器建立连接并进行交互 • main(String[] args): 客户端的主函数, 创建客户端对象, 并启动客户端 	<ul style="list-style-type: none"> • 构造函数 Authenticator(BufferedReader in, BufferedWriter out, BufferedReader consoleIn): 初始化用户身份验证器的相关属性 • authenticate(): 用户登录验证过程 	<ul style="list-style-type: none"> • 构造函数 CommandPrompt(-): 初始化命令提示符对象的相关属性 • dirOrCd(): 处理dir或cd命令 • get(String input): 处理get命令 • put(String input): 处理put命令 • exit(): 处理exit命令 • open(): 命令提示符状态, 处理用户的各种命令请求

6. 简要系统使用说明

6.1. 配置文件说明

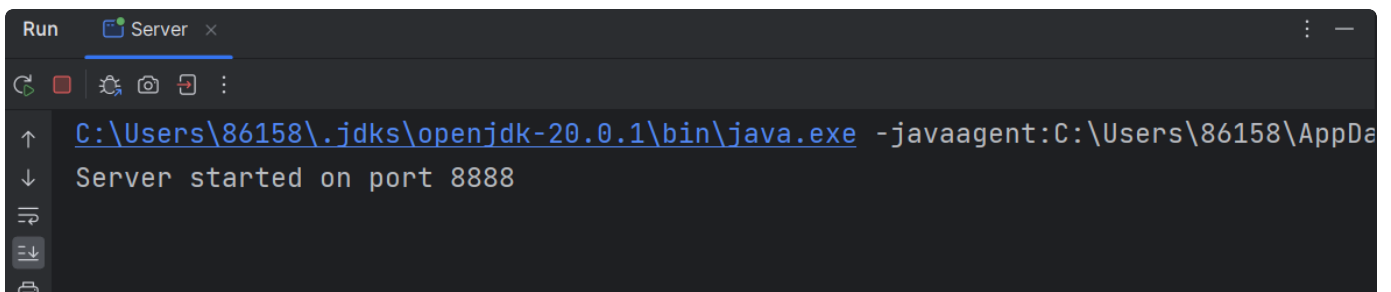
服务器端的文件结构以及配置文件如下:



配置文件中存储了12个用户的用户名及密码，在分割符后为系统所使用的目录ServerSrc（就是上传、下载文件的根目录，网络用户只能访问这个目录及其子目录）

6.2. 用户登录说明

启动服务器程序后，服务器将在指定端口监听：



启动客户端，输入服务器程序所在主机的IP地址进行连接，连接成功后服务器端输出提示信息：


```
C:\Users\86158\.jdk\openjdk-20.0.1\bin\java.exe -javaagent:C:\Users\86158\AppData
Please input server IP address: 0.0.0.0
```

```
C:\Users\86158\.jdk\openjdk-20.0.1\bin\java.exe -javaagent:C:\Users\86158\AppData
Server started on port 8888
Client 2.0.0.1 connected.
```

之后客户端提示用户输入用户名和密码（假如匿名登陆则不用输入密码），登陆成功后，客户端输出欢迎信息并进入指令输入状态，服务器端也将输出提示信息：

```
Username: Mike
Password: 123456
Login successful, welcome Mike!
Mike@2.0.0.1:ServerSrc$ |
```

```
Server started on port 8888
Client 2.0.0.1 connected.
Mike login successful.
```

此时在另一台电脑上登录同一个账号，将提示用户已经登录并拒绝后来的登陆者：

```
Username: Mike
Password: 123456
User already logged in.
Username: |
```

登录过程中，假如连续三次错误输入用户名或密码，客户端将拒绝再次登录并断开连接：

```
Username: Cindy
Password: sadklfn
Invalid username or password.
Username: Cindy
Password: jnsadfkL
Invalid username or password.
Username: Cindy
Password: ndfakl
Invalid username or password.
Too many attempts, closing connection.
Process finished with exit code 0
```

假如以匿名登陆，不需要输入密码，但是后续操作中无上传操作的权限：

```
Username: anonymous
Login successful, welcome anonymous!
anonymous@2.0.0.1:ServerSrc$ put local.txt
Access denied.
```

6.3. 用户操作说明

用户登陆成功后自动进入指令输入状态，显示当前用户名、用户IP地址、当前目录，提示用户输入指令：

```
Mike@2.0.0.1:ServerSrc$
```

6.3.1. dir指令说明

输入“dir”指令，会显示当前路径下的目录和文件，目录显示为蓝色，文件显示为白色，每行显示6个目录或文件：

```
Mike@2.0.0.1:ServerSrc$ dir
1    2    3.txt
Mike@2.0.0.1:ServerSrc$ |
```

6.3.2. cd指令说明

输入"cd 目录名", 会进入指定的目录, 假如输入cd ..会返回上一层目录, 假如目录名为不存在的目录或文件会返回错误信息, 假如访问根目录的上一层目录会提示无权限:

```
Mike@2.0.0.1:ServerSrc$ dir
1  2  3.txt
Mike@2.0.0.1:ServerSrc$ cd 1/555
Mike@2.0.0.1:ServerSrc\1\555$ cd ..
Mike@2.0.0.1:ServerSrc\1$ cd ..
Mike@2.0.0.1:ServerSrc$ cd ..
Access denied.
Mike@2.0.0.1:ServerSrc$ dir
1  2  3.txt
Mike@2.0.0.1:ServerSrc$ cd 4
Directory not exists.
Mike@2.0.0.1:ServerSrc$ cd 3.txt
Directory not exists.
```

6.3.3. get指令说明

输入"get 文件名", 会将当前目录下的文件下载至客户端的ClientSrc文件下, 假如文件名为不存在的文件或目录会返回错误信息, 假如ClientSrc文件下有同名目录会提示是否覆盖原文件, 并根据用户的输入执行操作:

```
Mike@2.0.0.1:ServerSrc$ dir
1  2  3.txt
Mike@2.0.0.1:ServerSrc$ get 1
Cannot download directory.
Mike@2.0.0.1:ServerSrc$ get 3.txt
File downloaded.
Mike@2.0.0.1:ServerSrc$ get 3.txt
File already exists. Do you want to overwrite it? (Y/N) Y
File downloaded.
```

6.3.4. put指令说明

输入“put 文件名”，会将客户端ClientSrc目录下的文件上传至服务器端的当前目录下，假如文件名为不存在的文件或目录会返回错误信息，假如服务器当前目录下有同名文件会提示是否覆盖原文件，并根据用户的输入执行操作：

```
Mike@2.0.0.1:ServerSrc$ dir
1  2  3.txt
Mike@2.0.0.1:ServerSrc$ put local.txt
File uploaded.
Mike@2.0.0.1:ServerSrc$ put none
File not exists.
Mike@2.0.0.1:ServerSrc$ put local.txt
File exists. Overwrite? (Y/N) N
Mike@2.0.0.1:ServerSrc$ dir
1  2  3.txt  local.txt
```

6.3.5. exit指令说明

输入“exit”，客户端目前登录用户会退出登录，客户端与服务器断开连接：

```
Mike@2.0.0.1:ServerSrc$ exit
Goodbye.
Process finished with exit code 0
```

```
Mike exit.
Client 2.0.0.1 disconnected.
```

6.4. 日志记录说明

日志文件每条记录的格式为“时间+IP地址+（用户名）+操作”，示例如下：

```
17 [2023-06-08 22:09:43] 2.0.0.1: Invalid username or password.
18
19 [2023-06-08 22:09:47] 2.0.0.1: Invalid username or password.
20
21 [2023-06-08 22:09:50] 2.0.0.1: Invalid username or password.
22
23 [2023-06-08 22:09:50] 2.0.0.1: Too many attempts, closing connection.
24
25 [2023-06-08 22:11:12] 2.0.0.1: Mike login successful.
26
27 [2023-06-08 22:14:23] 2.0.0.1: Mike list files in ServerSrc
28
29 [2023-06-08 22:16:47] 2.0.0.1: Mike change directory to ServerSrc\1\555
30
31 [2023-06-08 22:16:52] 2.0.0.1: Mike list files in ServerSrc\1\555
32
33 [2023-06-08 22:16:56] 2.0.0.1: Mike change directory to ServerSrc\1
34
35 [2023-06-08 22:16:58] 2.0.0.1: Mike list files in ServerSrc\1
36
37 [2023-06-08 22:17:02] 2.0.0.1: Mike change directory to ServerSrc
38
39 [2023-06-08 22:17:05] 2.0.0.1: Mike list files in ServerSrc
40
41 [2023-06-08 22:17:16] 2.0.0.1: Mike change directory to ServerSrc\1\555
42
43 [2023-06-08 22:17:19] 2.0.0.1: Mike change directory to ServerSrc\1
44
45 [2023-06-08 22:17:23] 2.0.0.1: Mike change directory to ServerSrc
46
47 [2023-06-08 22:17:28] 2.0.0.1: Mike change directory to ServerSrc
48
49 [2023-06-08 22:17:30] 2.0.0.1: Mike list files in ServerSrc
50
51 [2023-06-08 22:17:34] 2.0.0.1: Mike change directory to ServerSrc
52
53 [2023-06-08 22:17:38] 2.0.0.1: Mike change directory to ServerSrc
54
55 [2023-06-08 22:19:44] 2.0.0.1: Mike list files in ServerSrc
```

```
55 [2023-06-08 22:17:44] 2.0.0.1: Mike list files in ServerSrc
56
57 [2023-06-08 22:20:00] 2.0.0.1: Mike download file ServerSrc\3.txt
58
59 [2023-06-08 22:20:09] 2.0.0.1: Mike download file ServerSrc\3.txt
60
61 [2023-06-08 22:23:24] 2.0.0.1: Mike list files in ServerSrc
62
63 [2023-06-08 22:23:30] 2.0.0.1: Mike upload file ServerSrc\local.txt
64
65 [2023-06-08 22:23:56] 2.0.0.1: Mike list files in ServerSrc
66
67 [2023-06-08 22:25:17] 2.0.0.1: Mike exit.
```

7. 实验总结

本次实验中，我设计并实现了一个基于字符界面的C/S结构的文件传输程序。该程序分为服务器端和客户端两部分。其中，服务器端在启动时读入一个配置文件，之后验证用户登录信息，并根据客户端的命令给出相应回应，同时支持日志功能。而客户端可以连接服务器端，通过输入用户名及密码进行登录，成功登录后可以通过命令完成相应操作。

实验中遇到的主要问题包括处理多个用户端同时访问的情况，协调服务器端和客户端之间的输入输出流，处理用户的错误操作并返回提示信息，通过查询相关资料和代码调试等途径，我解决了这些问题，并在这个过程中增强了对相关知识（多线程、网络编程等）的掌握，锻炼了代码调试的能力。