Directory Management System

(Lab 3 Distributed Systems)

Bindhu Shree Hadya Ravi - 1001699836

Introduction:

I have created a Client Server application for Directory management system. The server will maintain all the client directories and responds to many clients simultaneously. There are many operations which can be done on the directories at the server and I have explained them below. The overall logic is to create a server application which can handle maximum of 3 clients and maintain each client's directories. For lab2 I have implemented polling mechanism to maintain the consistency between the local disk and the server home directories.

Design and Flow:

Server is made to handle up to 3 clients as expected by the lab requirement. Server is initialized when we execute and it opens a window, this window has text box and few buttons, text box is the place where we display the requests from the client. Server will be listening at the socket right after the application is started.

When we execute the client, we will have text box to see the output and the button using which we send the commands to the server. When client requests for connection, server accepts it and stores its username in server's local data structure. There will be separate threads created for each client which gets executed. When clients exit, the process is killed. Related messages will be displayed on client and server UIs, example: "Server Unavailable".

When client connects to the server, server should create a directory with client name if it doesn't exist. Client should be allowed to Create, Delete, Move, Rename and List directories. Client should not be allowed to access outside its directory. For lab2 I have implemented polling mechanism to maintain the consistency between the local disk and the server home directories. For lab3 I have implemented logging mechanism to maintain the commands executed on directories and to undo them.

Execution Examples:

Import the Eclipse project to your eclipse IDE and make sure you add Apache common io (Commons IO 2.8.0) library. Here is the link: https://commons.apache.org/proper/commons- io/download_io.cgi . Below are some steps and example.

1. Run the Server (Server.java) and its UI opens up
2. Run the Client (Client.java) and its UI opens (Now they will be connected if accepted by

   server). To execute commands, click on the button "Commands" and enter the below arguments:

   - Create command takes 2 arguments, first argument as create keyword and second argument as path where it should be created. It should start with /ClientsUserName else he will throw an error. Example: When client UTA wants to create new directory newone, here is the command for that: create /UTA/newdir/newone
   - List command takes one argument. Example: When client UTA wants to list all the directories and files within that, here is the command for that: list

- Delete command takes 2 arguments, first argument as delete keyword and second argument is the path of the directory to be deleted. All the sub-directories and its files will also be deleted. Path should start with /ClientsUserName else he will throw an error. Example: When client UTA wants to delete a directory by name library which is inside dir, here is the command for that: delete /UTA/dir/library
- Rename command takes 3 arguments, first argument as rename keyword and second argument is the source path of the directory to be rename and third argument is the path of the destination directory. Path should start with /ClientsUserName else he will throw an error. Example: When client UTA wants to rename a directory by name library to ktcenter, here is the command: rename /UTA/dir/library /UTA/dir/ktcenter
- Move command takes 3 arguments, first argument as move keyword and second argument is the source path of the directory to be moved and third argument is the path of the destination directory. All the sub-directories and its files will also be moved. Path should start with /ClientsUserName else he will throw an error. Make sure that the Source directory exist. Example: When client UTA wants to rename a directory by name library to ktcenter, here is the command: move /UTA/dir/library /UTA/dir/ktcenter
- Server will be able to check all the connected clients from its interface
- Send sync followed by Server home directory to sync it with the client. Eg: sync home_directory1
- Send desyn command from client to desync the local system to server home directories Eg: desync
- Do undo option at Server to undo the previous command executed on directories

Technical Requirements:

I ran the application on the following environment -> Java 15, Eclipse Java IDE, Mac OS Catalina, mandatory linking of Apache commons io library (recent version) in Eclipse, Swing for application user interface.

References:

Swing scroll was taken from the reference:

https://stackoverflow.com/questions/10177183/java-add-scroll-into-text-area