Computer Science 2B

Practical Assignment 07

2023-09-19

Deadline: 2023-10-03 12h00 Marks: 100

This practical assignment must be uploaded to eve.uj.ac.za **before** 2023-10-03 12h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Make use of proper coding conventions and documentation. Marks will be deducted if these are not present. Your submission should include a batch file.

The reminder page includes details for submission and queries. Please ensure that **ALL** submissions follow the guidelines. The reminder page can be found on the last page of this practical - read the reminder page carefully.

This practical will focus on semester test revision

1. Please note: [100]

   - You are required to create a **Java FX application** for the **client GUI** component.
   - Make use of the **additional files** provided for your project. A base project has been provided.
   - Ensure your project is structured in accordance with the **"Project Setup For Batch Files"** document attached as the last page of this document.
   - Your **client main class** should be called Client.java in the csc2b.client package and your **server main class** should be called Server.java in the csc2b.server package.

A group of citizens from planet ZEDEM have decided they need a way to download their favorite audio files from a central server. They have asked you to develop a networked client application and server application. Your application should allow for the transfer of audio files over a network by using the ZEDEM protocol. The ZEDEM protocol requires clients to login to the server before other commands can be processed. The server keeps track of the available audio files by storing each audio file with its corresponding ID in a text file. ZEDEM runs on port 2021.

The following request commands are available in ZEDEM :

- **`BONJOUR <Name> <Password>`**
  Provide a name and password for client login, e.g.

  `BONJOUR` User Pass235. The server should validate the provided name and password. If the provided credentials are not valid, an error message must be returned. However if the login is successful, the server should allow the client to make other requests.

- **`PLAYLIST`**
  Return a list of available audio files

  The server retrieves the available audio files from the `List.txt` text file and return the list to the client.

- **`ZEDEMGET <ID>`**
  Return a response message followed by the requested audio file to the client

  The server should validate the ID and return an error message if the ID is not valid. If the ID is valid, a confirmation message should be sent including the file size. The File should then be read in and transmitted to the client.

- **`ZEDEMBYE`**
  Log a client off.

The following responses are used in ZEDEM :

- `JA <Message>` Successful command with `<Message>` providing a helpful message from the server.
- `NEE <Message>` Unsuccessful command with `<Message>` providing a reason that the command did not work.

Complete the **`Server`** class. This class is responsible for binding to the ZEDEM port to listen for clients. The server must be able to handle multiple clients. Any clients which connect are handled by the **`ZEDEMHandler`**.

Complete the **`ZEDEMHandler`** class. This class is responsible for handling commands which are received from a client. This class is also responsible for handling client login. Registered users are stored in a text file called `users.txt`.

Complete the `Client` and `ZEDEMClientPane` classes. The `Client` class will provide the starting point for the client side. The `ZEDEMClientPane` class will have buttons for each command. You must use the JavaFX library for your GUI. Any pane layout may be used for your `ZEDEMClientPane`. The audio files that are downloaded should be saved to disk.

Any errors from the server must be displayed to the client.

Reminder

Your submission must follow the naming convention:

`SURNAME_INITIALS_STUDENTNUMBER_CSC02B2_2021_PTA`

Your submission must include the following folders:

- `bin` - Should be empty at submission but will contain runnable binaries when your submission is compiled.
- `docs` - Contains the .bat files for compiling your project without an IDE should you choose to make use of that approach. (For the purpose of the test - JavaDocs will not be generated)
- `src` - Contains all relevant source code.
- `data` - Contains sub-folders for client and server where transferred files are saved.

NB

Submissions which **do not compile** will be capped at 40%

Execution marks are awarded for a correctly functioning application and not for having some related code.

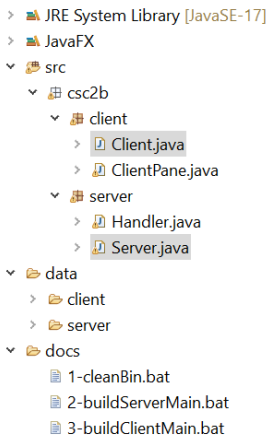
Failure to save the solution to the correct locations will mean that the Academy will not be able to mark the submission and you will forfeit marks as a result.

Mark sheet

1. `Server`
   (a) Create `ServerSocket`. ____ out of [02]
   (b) Accept client and pass to `ZEDEMHandler`. ____ out of [03]
   (c) Multi-threaded client handling. ____ out of [05]
2. `ZEDEMHandler`
   (a) Handle `BONJOUR` . ____ out of [05]
   (b) Handle `PLAYLIST` . ____ out of [05]
   (c) Handle `ZEDEMGET`
      i. Process parameters. ____ out of [03]
      ii. Returning file size. ____ out of [05]
      iii. Returning audio file . ____ out of [05]
   (d) Handle `LOGOFF`. ____ out of [02]
3. `Client` and `ZEDEMClientPane`
   (a) Setup application starting point. ____ out of [02]
   (b) Connect to `Server`. ____ out of [02]
   (c) Setup streams. ____ out of [02]
   (d) Send commands. ____ out of [02]
   (e) Process responses. ____ out of [04]
   (f) `BONJOUR` button and listener. ____ out of [02]
   (g) `PLAYLIST` button and listener. ____ out of [02]
   (h) `ZEDEMGET` button and listener. ____ out of [02]
   (i) `ZEDEMBYE` button and listener. ____ out of [02]
4. Coding convention (structure, layout, OO design) ____ out of [05]
5. Commenting (normal and JavaDoc commenting). ____ out of [05]
6. Correct execution
   (a) Show `ZEDEMClientPane`. ____ out of [05]
   (b) Sending requests and responses. ____ out of [10]
   (c) Successfully sending and saving audio files . ____ out of [10]
   (d) Handle exceptions. ____ out of [05]
   (e) Connection cleanup. ____ out of [05]

2022-CSC2B-Project-Setup

Project Setup for Java Client-Server Applications to work with provided batch files sample:



For the practical test – a skeleton project has already been provided with the required structure. Ensure you keep the structure the same.

Ensure you name the Main class components for your project as in the screenshot example above.

1. Create the following **packages**: **csc2b.client** and **csc2b.server**
2. Ensure your class with the main method for the client is called "Client.java" and that your class with the main method for the server is called "Server.java"
3. Include the provided data folder (with the client and server subfolders) in your project.
4. Include the provided batch files in the docs folder of your project:
    - There are 3 bat files:
        i. 1-cleanBin.bat which clears all the class files from the bin folder
        ii. 2-buildServerMain.bat which compiles and runs the main server class (Server.java)
        iii. 3-buildClientMain.bat which compiles and runs the main client class (Client.java)
    - Run these bat files in the above order to compile and run your code.

Eclipse run configuration VM arguments for JavaFX:

--module-path "C:\javafx-17\lib" --add-modules
javafx.base,javafx.controls,javafx.fxml,javafx.graphics,javafx.media

Refer to the Eclipse-JavaFX-Setup-Guide-2022 pdf document for more details on JavaFX Setup.