CS328 Distributed Systems Assignment 1: RMI-based Simple File Server

Due by 12:00 Friday noon, March 20

March 6, 2020

1 Introduction

You are required to design and code a working program and submit well-commented code together with a lab report. Any text should be in English only. Plagiarism is strictly prohibited.

2 Preamble

Before working on this assignment, you should be familiar with Java Remote Method Invocation (RMI). Tutorials are available in the last page of slides of Lab 3. Also, some basic knowledge of manipulating files using Java's file API is needed.

3 Exercise Specification

In this assignment, you are asked to build a simple file server using Java RMI framework. More specifically, your file server should implement the following functionalities: **read**, **create**, **edit**, **delete**, **copy**, **move**, and **rename**. Also, file status should be supported such as **size**, **last modify time** and **last access time**. Directory support is currently not required.

The program should be developed as a client server application. The files are stored as actual files on your local filesystem (In this case, store all files under the same directory of your server .class file) on the server side. While running your program for demonstration, you are required to start server and client from different directories, and program a set of test cases to test your file system. For simplicity, you may just hardcode your test cases in your Client's `main` function.

You need to provide screenshots of your running results (e.g. console output, file folder screenshot) and paste them into report.

Required running method should be like this:

```
# In the first terminal window
$ /home/user/server java FileServer
File server is ready......

# In the second terminal window
$ /home/user/client java FileClient
File client: creating file 1.txt
...
```

Then after running, take screenshots of both terminal, and the file content under server directory (e.g. `/home/user/server`)

A reference of file server interface design is provided as follows:

```
// This is just a reference, you may use your own definition
// as long as you meet the requirements

// The interface is provided for reference purpose,
// there may be something missing for it to be actually implemented,
// such as throwing exceptions

public interface IFileServer {

    public String read(String fileName);

    public void create(String fileName);

    public void edit(String fileName, String newContent);

    public void delete(String fileName);

    public void copy(String sourceFileName, String destinationFileName);

    public void move(String sourceFileName, String destinationFileName);

    public int size(String fileName); // returns size in bytes

    public long lastModified(String fileName); // returns timestamp

    public long lastAccessed(String fileName); // returns timestamp

}
```

4 Submissions

Submit your assignment (code and report) in an archive to the course's Sakai page. Submissions out of the Sakai system is not accepted unless otherwise stated. Be sure to submit your work before deadline, or there may be penalties.

5 Assessment

The full marks for this assignment is 100, and are distributed as follows:

Code	
Functions	**Marks**
File Operations	$5 * 9 = 45$
RMI Server	10
RMI Client	10
Quality of code (comments, naming, etc)	10
Total	**75**

Report	
Items	**Marks**
Demonstration (Screenshots and explanation)	25
Total	**25**

The template (in Microsoft Word format) is provided in Sakai site. You can also choose to use other typesetting tools such as LaTeX and Markdown, as long as you do not miss any required parts in the report. Any content in the report template which is not listed in the mark table above is not required.