

DSPro

Optional Homework on Distributed Systems

© 2021 by José María Foces Morán

General guidelines

- The total number of points granted to this homework is 25. You have no obligation to deliver the solution to all of the homework exercises; any part thereof will count if properly explained and developed.
- This homework is worth 1 point toward the maximum of 10 points that correspond to the overall course grade. Take into account that the grade obtained in this homework cannot be used toward obtaining the passing grade which is exclusively composed of the two Term Exams: Term Exam 1 (TE1) and Term Exam 2 (TE2). Please, recall as well that passing each of TE1 and TE2 is a necessary condition for passing the whole course.
- Program sources must be clearly and succinctly commented
- Apply a simple OO design strategy or structured programming strategy depending on whether you are programming in Java or in C, respectively
- Include rich explanations of your design decisions and the unit tests that demonstrate the correction of your programs
- This is an **independent study** homework: You can only submit your original work, programs and explanations. You can incorporate source code from open software projects, in which case you must cite the authors and their overall weight must be small.
- Submit the solution to **each exercise** in a **separate folder** which name must be "Exercise 1", "Exercise 2", etc. Each folder must contain one folder for the sources (**src**), another folder for the explanations (**docs**) and a last folder for the executable files (**bin**)
- Include software construction instructions and shell scripts if necessary
- Compress the **complete folder structure** mentioned above in a **.zip file** (Please, use .zip exclusively, otherwise, I might not be able to decompress the archive which might damage your grade). Upload the zip file for your DSPro homework. Do the upload to the following virtual campus homework:



- The **.zip** file name should have the following structure:
DS-Pro-<Your ID>.zip. Example: DS-09876543A.zip
- Deadline for submission is 19th-January-2021 at 23:59

1. [4] Simple probabilistic time synchronization algorithm

Using ICMP timestamps and Raw IP Sockets, build a C program that synchronizes its host's clock (The client) with another host of your choosing (The server). The client program will fetch the time from the server a number of times and after the last one, it will set its own clock in sync with the server's.

- a. The program must capture the server name or IP address from the command line along with the number of times to retry synchronization with the server.
- b. The NTP client must be stopped before your program can run, just to make sure that NTP doesn't interfere with your programs, somehow. Explain how you will go about stopping the ntp client in your Linux
- c. The source code provided in the Lab Practices of DS can be used as the basis for developing this program but you must mention in the comments that some parts of your code are not original. If you use the base code provided in the practice scripts, you must be aware that that code might contain semantic errors and, consequently, you are expected to correct them. Particularly, check that the given source program faithfully implements the Christian's time adjustment procedure explained in the lectures of the present academic year, 2020 (Observe the timestamps used for performing the adjustment).
- d. Explain what tests you will perform to demonstrate that the program functions correctly. Particularly, we are interested in your building a program that continually prints out the current server time along with the client -local- time so that the user can easily observe the progress of **adjtime()** (Every 2 seconds, for example). Set an initial time difference between the two clocks not greater than 15 seconds, otherwise the time length taken by the **adjtime()** system call will be too long.

2. [4] RFC 868 time protocol

Skim the RFC 868 time protocol and implement the client/server pairs appearing on the adjoining table. In each C/S pair, the client will request the current time to the RFC 868 time server and will print out the response. Note, in this case we are not seeking to synchronize any clocks, but only to print out the server time in the client.

Implemented over	Client	Server
TCP	C	Java
UDP	Java	C
JRMP	Java RMI	Java RMI

- a. Deliver each C/S pair in a separate folder containing the corresponding sources, executable code and documentation
- b. Provide extensive explanations about the considerations you made about this question
- c. Since the RFC 868 time format must be the same across the three requested implementations, were you able to reuse some of the data structures or the algorithms involved? If so, explain how you did it.
- d. Explain the tests that you designed to check the correctness of the implementations

3. [3] RMI C/S and concurrency

Write an RMI server that keeps a remote counter of certain events. Your task consists of building a Java RMI Server that implements the three following remote methods:

```
public void reset(); // Reset the counter to 0
public long int increment(); //Increments the counter (counter++)
public long int decrement(); //Decrements the counter (counter--)
```

Upon server initialization, the counter must be initialized to 0 by using method `reset()`. You must use only simple Java synchronization primitives such as synchronized blocks and methods:

- a. Explain the general strategy that you devised for guaranteeing atomic writes on the counter variable

- b. Write an RMI client program that permits testing the server
- c. Provide an extensive discussion of your software design and the tests that demonstrate that it functions correctly

4. [2] TCP timestamps option in RFC 7323

- a. Compose a summary about the TCP timestamps based on section no. 3 of RFC 7323 titled "TCP Timestamps Option". Focus on the usefulness of this option, particularly on the hardest challenges posed by it.
- b. Can this TCP option be used for having the client synchronize its clock with the clock of the server?

5. [1] RMI lab practice script activities

In this exercise, we seek to improve the RMI client/server practice that we did in the lab according to the following outline:

- a. Clients don't know the location of the RMI server. Clients find out the location of the RMI server by sending a request to the Local Network broadcast address and UDP port 1234. This entails the RMI server to listen for these requests when it boots. When the server receives such a broadcast request, it reacts by providing the client a response string containing the IP address and the TCP port on which the server's RMI registry is located. Explain the technical considerations necessary for this solution to work out well.

6. [2] Analysis of Request/Response protocols in Distributed Systems

Sending an e-mail message using the SMTP protocol requires numerous short, back-and-forth messages between the user agent (The client) and the e-mail server:

- a) Why is this problem particularly troublesome when the client is connected to a wireless network?
- b) What would you suggest to do to resolve it without affecting the SMTP server?

7. [7] IP Network Design

A university has 4 internal networks, A, B, C and D which maximum sizes are 100, 400, 800 and 1000 hosts.

- a) Choose a private IP addressing space that may encompass all the networks and calculate each of the network numbers, broadcast address and number of usable addresses.
- b) Explain the considerations that you have taken into account for making the choice of IP addresses. Specifically, which RFC documents the Private Addressing spaces available in IPv4?
- c) What conditions apply to two IP addressing spaces for them to be correctly coalesced into a single resulting IP space?
- d) Produce an internetwork design assuming that network A and B belong to the same physical location whereas C and D are located each at different cities. List the contents of the forwarding tables of each router for your design.
- e) Which type of routing protocol would you choose for your internetwork, an interior gateway protocol or an exterior one?
- f) Choose a concrete routing protocol and explain your choice.