Exercise 2 – RFC 868 time protocol

Provide extensive explanations of the considerations and problems that you found in this question.

I only encountered one difficulty:

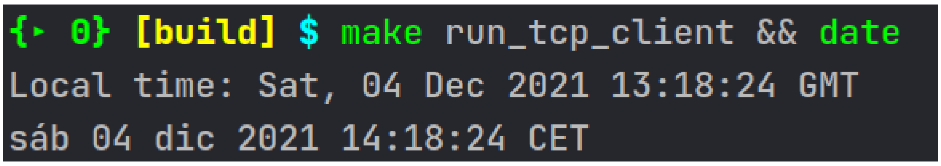
- In UDP part, when reading the number of seconds from 1900 until now, in the Java client I receive a number that does not correspond to the one sent by the server.

  The solution I provide is tricky because I pass the number to its binary representation and store it in a string. After that, the **Long.parseLong(String s)** Java method returns the correct number of seconds which are going to be added to the RFC 868 base date.

Explain what role marshalization plays in this exercise.

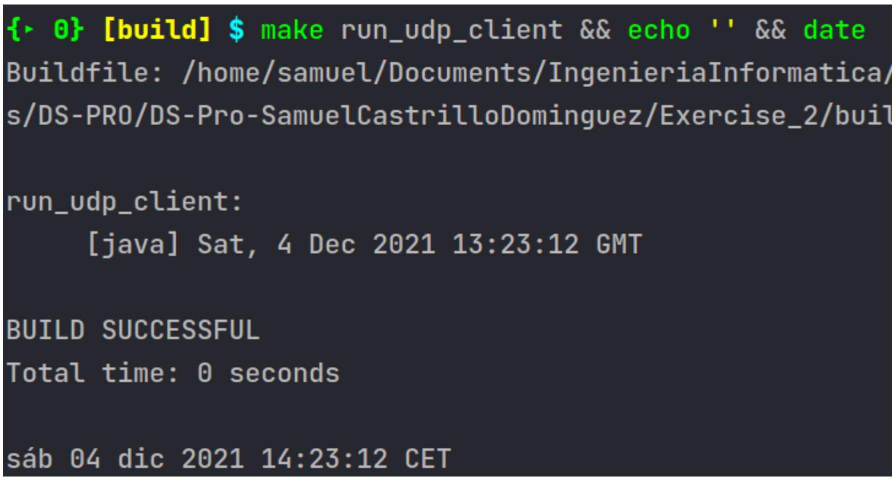
I did not do the part of JRMP.

Explain the tests that you designed to check the code.



```
{▸ 0} [build] $ make run_tcp_client && date
Local time: Sat, 04 Dec 2021 13:18:24 GMT
sáb 04 dic 2021 14:18:24 CET
```

Figure 1: Time obtanied with TCP



```
{▸ 0} [build] $ make run_udp_client && echo '' && date
Buildfile: /home/samuel/Documents/IngenieriaInformatica/
s/DS-PRO/DS-Pro-SamuelCastrilloDominguez/Exercise_2/buil

run_udp_client:
     [java] Sat, 4 Dec 2021 13:23:12 GMT

BUILD SUCCESSFUL
Total time: 0 seconds

sáb 04 dic 2021 14:23:12 CET
```

Figure 2: Time obtained with UDP

NOTE: Extensive documentation in the source code.

In TCP (Figure 1): The server waits for a client in the accept() call.

In UDP (Figure 2): The server waits until receiving the message “GET_DATE” from the client.

When the server establishes the connection with the client, it calculates the number of seconds from 1900 and send them to the client.

In order to show better results, the execution of the programs is compared with the *date* function provided by Linux. As we can see, the time obtained differs in one hour. It is due to the timezone. The RFC 868 establishes that the timezone must be GMT (Greenwich Mean Time).