

Universidad de León
School of Industrial, Computer and Aerospace Engineering
Course on Distributed Systems and Networks

Homework #1: Introduction to Distributed Systems

Submit a single .zip file containing your solutions to the HW exercises. Only .pdf and .c files are accepted. Submit via agora by 21:00 on Friday, 27th-September-2024.

Exercises

1. Introduction to Distributed Systems

- 1. What is a distributed system made of?**
- 2. Interpret the meaning of this phrase: “The TCP protocol makes omission faults transparent to applications”**

2. UDP

- 1. Download the script to the practical that we did on week #39, titled “[Practical on CS with datagram sockets in C](http://paloalto.unileon.es/ds/lab/udpcsscript.pdf)” (The link that points to it is <http://paloalto.unileon.es/ds/lab/udpcsscript.pdf>) and redo exercise 1 in your own Linux host, at your home and having no Internet connectivity. Explain each step you take alongside the final tcpdump trace.**
- 2. Extend the echoServerBase.c and the echoClientBase.c from the practical mentioned in the previous exercise such that the server returns the actual loop’s *iteration number* (An int, which in C, under a mainstream platform is represented with 32 bits) along with the data that should be echoed back to the client.**

The client should, as well, access that integer when received and have it properly printed out. The loop’s source code is located at line number 62, and the counter is represented in an int variable which name is counter. To simplify your work, the following *empty* function has been included in the source code:

```
void appendCounter(char *buffer, unsigned int counter, unsigned int *nbytes){  
}
```

Function appendCounter(), receives all of the parameters necessary for appending the value of counter to the buffer that is to be returned to the client. Consequently, you must code your solution within function appendCounter(), exclusively. In Computer Networks we reviewed how to send an integer over a socket (Functions htonl() and ntohl()).

Extend the client’s code so that it prints the iteration number out every time it sends a new request to the server.

