Heidelberg University
Wintersemester 18/19

Group Parallel and Distributed Systems (PVS)
Artur Andrzejak, Diego Costa

Problem Set 1 for lecture Distributed Systems I (IVS1)

Due: 30.10.2018, 14:00 Uhr

Exercise 1 (1 Point)

Which of the following can be considered distributed systems? Justify your answer based on the properties of distributed systems discussed in the first lecture.

- The human brain
- A modern laptop
- A bee colony
- The entirety of all Android devices

Exercise 2 (2 Points)

What advantages and disadvantages has the connection-oriented protocol TCP compared to e.g. UDP? For which applications would you prefer UDP over TCP?

Exercise 3 (2 Points)

Suppose that the UDP receiver computes the Internet checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? Explain.

Exercise 4 (3 Points)

Using the TCP Client/Server Python code presented in Lecture 2, implement the `prepareReply` function on the server side to count the amount of words received by the client. Prepare and test your implementation on scenarios with messages containing up to one thousand words. **Submit your code and test cases into Moodle.**

Exercise 5 (8 Points)

Read the tutorial from RealPython[1] on socket programming in Python. Expand on the concepts of socket programming given in class and learn how to implement a server/client that handles/submits multiple simultaneous connections.

In this exercise you should develop, on top of a multi-connection server/client implementation, the following authentication scenario:

- The client reads from the command line the name and password, computes a hash value of the password (use the hashlib[2] library), and asks the server for the authentication (submit only the username and hashed password). Finally, the client prints the message returned by the server in the screen.

[1] https://realpython.com/python-sockets/
[2] https://docs.python.org/3/library/hashlib.html

- The server receives the credentials and validates the username and password. You can hard-code a set of valid credentials for this exercise, but keep the authentication logic on a separate function. The server should return one of two responses:
  - „Valid credentials“, if the username and password matches a pair in the database.
  - “Username and/or password are not in registered in the database“, otherwise.

Submit your client/server code in Moodle. If you are not familiar with Python, use other language (e.g. Java, Kotlin, C/C++).