Computer Networks

Homework#4 Network Programming Experiences

Due: Friday, December 21th, 2018

1. Controlling Drones

Supposing that you are controlling a drone. A drone (a server program) can receive four types of commands from a remote controller (a client program):

- Move_UP
- Move_DOWN
- Move_LEFT
- Move_RIGHT

The moving area of the drone is restricted in a rectangle(10×10). At the beginning, the drone is at the position$(5,5)$. Whenever the drone receives a control command from the remote controller, it makes a move corresponding to the type of command.

For example, the current position of the drone is$(5,5)$. It receives the command **Move_UP**, so it moves up to the position $(5,4)$ and it reports its new position to the remote controller. If the drone is going to move out of the moving area, it cannot make the move.

Task: Your task is using Socket programming with TCP/IP protocol to simulate the drone (server program) and the remote controller (client program).

Input: The sequence of commands is read from the text file "moves.txt" in which each command is on one line. Drone controller (client program) reads commands line by line from this text file and sends these commands to the drone (server program) one by one. Example of the "moves.txt" file is as follows:

moves.txt
Move_UP Move_UP Move_DOWN Move_LEFT

Output: Whenever the controller (client program) receives a new position from the drone (server program), it saves the new position into the text file "*position.txt*". Consequently, the "*position.txt*" contains only one position which is the newest position of the drone. Example of the "*positions.txt*" file is as follows:

position.txt
(4,4)

2. Online Calculator (different with exam)

You are going to develop an Online Calculator which receives the expression from clients, then calculate the expression and return the result to clients. For simplicity, the expression is a string containing integer numbers and only two operators: addition (+) and subtraction (-). We assume that all expressions are correct. Examples of expressions are as follows:

- Expressions: "5+7–3+2", "1+1"

Task: Your task is using Socket programming with TCP/IP protocol to:

a. Develop the Online Calculator which is a server application

b. Develop a client program which sends expressions to the Online Calculator and receive results.

Input: The expression is read from the text file "*expression.txt*". Client program reads the expression from this text file and sends to the Online Calculator (server program). At a time, the client can send only one expression. Example of the "*expression.txt*" file is as follows:

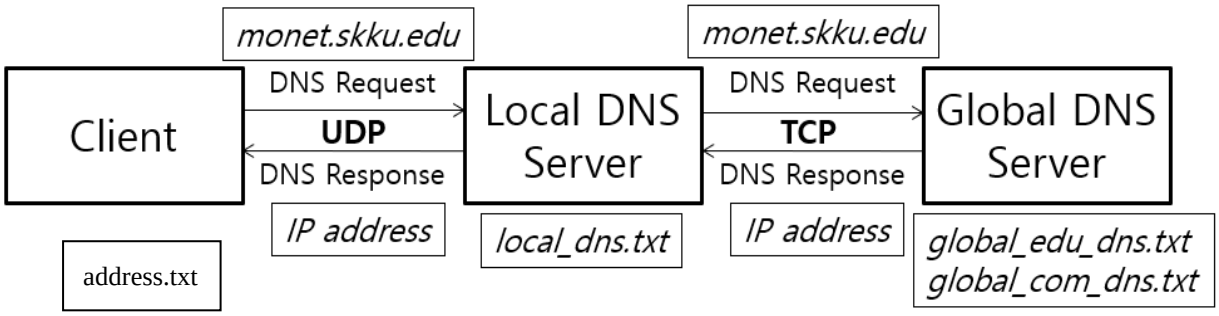
expression.txt
1+5-2+7-3

Output: Whenever the client program receives the result from Online Calculator, it saves the result into the text file "*result.txt*". Example of the "*result.txt*" file is as follows:

result.txt
8

3. Simple DNS simulation

You are going to simulate how DNS servers work that is described in the figure below:



As can be seen from the Figure above, there are one client and two DNS servers. When the Client wants to get the IP address of a domain, the following steps are carried out:

1. The Client reads the domain name from **address.txt** and sends the domain name to the Local DNS server
2. The Local DNS server checks if the domain name is in its database (the text file **local_dns.txt**) or not. If YES, it returns the corresponding IP address of the domain name to the Client, otherwise, the Local DNS forward the domain name to the Global DNS and ask for the IP address
3. At the Global DNS Server, it searches IP address of the given domain name in two text files **global_edu_dns.txt** and **global_com_dns.txt** and returns the corresponding IP address. If there is no matching domain name, it returns the message "Not found" to Local DNS server
4. After receiving result from Global DNS Server, the Local DNS Server forwards the data to the Client
5. The Client write the result to the text file **result.txt**

Task: Your task is using Socket programming with TCP/IP and UDP protocols to:

a. Develop the Client
b. Develop the Local DNS Server which is listening for incoming connections at the port 8000
c. Develop the Global DNS Server which is listening for incoming connections at the port 9000
d. The Client communicates with the Local DNS Server by UDP protocol
e. The Local DNS Server communicate with Global DNS Server by TCP/IP protocol
f. All Client, Local DNS Server, Global DNS Server run on localhost address (127.0.0.1)

Input: The domain name is read from the text file **domain.txt**. The Client program reads the domain name from this text file and sends to the Local DNS Server (server program). At a time, the client can send only one domain name. Example of the **domain.txt** file is as follows:

address.txt
www.skku.edu

Database of the Local DNS server and Global DNS servers is a text file with multiple lines. One line has two parts: domain name and IP address **which are separated by a space**. Example of the databases is as follows:

local_dns.txt	global_edu_dns.txt	global_com_dns.txt
www.skku.org 112.124.11.23	www.skku.edu 212.124.11.20 www.seoul.edu 212.124.11.14	www.skku.com 112.24.11.19 www.kaist.com 112.24.11.12

Output: Whenever the client program receives the result from Local DNS Server, it saves the result into the text file **result.txt**. Example of the **result.txt** file is as follows:

result.txt
212.124.11.20

Description:

This homework was designed to improve your network programming experiences. Students will focus on functional aspects of development. There is no exact method or solution for programming. You can use open source or you can create your own functions. You can get help from colleagues around you, but copying is not allowed. Coding yourself is the key to this homework.

What you have to turn in:

- A Project Report:
  - File named HW4_ID_NAME.pdf
  - Declare your development environment (platform and compiler); and how to COMPILE/BUILD and RUN your applications step by step.
  - Declare the folder structure of the source code and the meaning of each file inside
  - Declare how to set up and how to use extra non-standard applications/components you have used for your implementation, if any.
  - Describe in detail about your implementation of functionalities that are considered very important (based on your perspective).
  - DO NOT include the whole source code in the report.
  - Show the screenshots along with explanation for each of functionalities you have implemented.
  - Upload the the i-campus Homework#4-report

- Source codes:
  - A compressed file named HW4_ID_NAME.zip
  - The executable source code of your programs.
  - The executable files of the application on a specific platform (Linux)
  - Configuration files and related files, if any.
  - Upload to the i-campus Homework#4-source codes

NOTICE:

✓ BOTH ORIGINAL AND COPY WILL GET –30 POINTS EACH INSTEAD OF 0S.

✓ ANY SOURCE CODE WITH COMPILE OR RUNTIME ERROR WILL GIVE YOU 0 POINTS.

✓ ALL THE HOMEWORK MATERIALS SHOULD BE MADE IN **ENGLISH**.

Good luck!