Homework #1

Fall 2015

Due date: 2015/10/11

Guess Number

There are two UDP servers (S1 and S2) with the same IP address but different ports , and you only know the port of S1. In order to connect to S2, you need to send requests to S1 to “guess” the port of S2.
After sending the request, S1 will response a message back and tell you whether you guess the correct port number or not.
After you have guessed the correct port, send a message to S2.

Requirements

1. Write a UDP client program with 2 arguments “ server_ip “ and “ S1_port”
2. Using these two arguments to send a message to S1 and the message is a String in JSON form - {“guess” : Number }.
   The Number is between 3000 and 60000.
3. And then, S1 will send message back and the message is a String in JSON form – {“result” : String }
   The contents of result have the following case:
   “larger” : S2’s port is larger than you guessed.
   “smaller” : S2’s port is smaller than you guessed.
   “bingo!” : You have guessed the correct S2’s port.
   “Wrong JSON content” : The content in JSON form is not correct.
   “parse_error: XXXX” : S1 can’t parse your message.
4. After you have guessed the correct port, send a message to S2 and the message is a String in JSON form – {“student_id” : String (your student ID) }
5. Finally, S2 will send a message back and the message is a String in JSON form – {“result” : String }
   The contents of result have the following case:
   “Congrats! XXX”: You have connect to S2 correctly.
   “Wrong JSON content” : The content in JSON form is not correct.
   “parse_error: XXXX” : S2 can’t parse your message.
6. You need to print out all messages including sending and receiving one.

Demo

- Program runs correctly. (60%)
- Executing time less than 1 second. (10%)
- Oral defense (30%)

Note

1. We would provide a sample UDP server code for you.
   This sample code is written by Node.js
   Hence, you need to set up system environment to run server.
   Node.js : https://nodejs.org/en/
2. In demo, we would ask you to connect to UDP server provided by us.
3. You could use any programming language to write your own code.
4. Reference of JSON : http://www.json.org/

Sample run

Client side	Server side
#set up UDP Client listening on 127.0.0.1:5567 #1 send {"guess" : 30500 } receive {"result":"larger"} #2 send {"guess" : 57234 } receive {"result":"smaller"} #3 send {“guess” : “not a number”} receive {"result":" Wrong JSON content "} #4 send “sdfsdfg” receive {"result":"parse_error: SyntaxError: Unexpected token s"}	#set up S1 listening on 127.0.0.1:5566 S2 listening on 127.0.0.1:56666 #1 S1 receive: {"guess" : 30500 } S1 send back: {"result":"larger"} #2 S1 receive: {"guess" : 57234 } S1 send back: {"result":"smaller"} #3 S1 receive: {"guess" : "not a number" } S1 send back: {"result":"Wrong JSON content"} #4 S1 receive: sdfsdfg [SyntaxError: Unexpected token s] S1 send back: {"result":"parse_error: SyntaxError: Unexpected token s"}

#5 send {"guess" : 56666 } receive {"result":"bingo!"} #6 send {"student_id" : "0286030" } receive {"result":"Congrats! 0286030"}	#5 S1 receive: {"guess" : 56666 } S1 send back: {"result":"bingo!"} #6 S2 receive: {"student_id" : "0286030" } S2 send back: {"result":"Congrats! 0286030"}