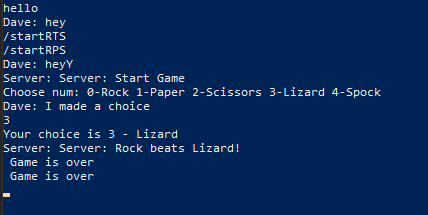
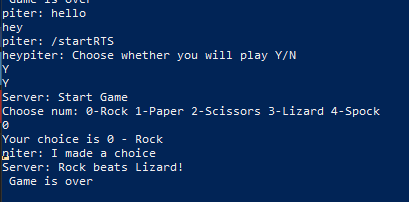
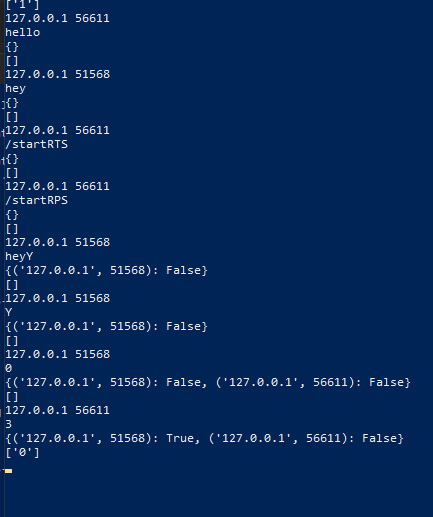
Курсовая работа по Инфокоммуникационные системы и сети



Server.py

import socket  
from enum import IntEnum  
  
  
class Action(IntEnum):  
 Rock = 0  
 Paper = 1  
 Scissors = 2  
 Lizard = 3  
 Spock = 4  
  
  
victories = {  
 Action.Scissors: [Action.Lizard, Action.Paper],  
 Action.Paper: [Action.Spock, Action.Rock],  
 Action.Rock: [Action.Lizard, Action.Scissors],  
 Action.Lizard: [Action.Spock, Action.Paper],  
 Action.Spock: [Action.Scissors, Action.Rock]  
}  
  
  
def determine\_winner(user1\_action, user2\_action):  
 defeats = victories[user1\_action]  
 if user1\_action == user2\_action:  
 data = (f"Both players selected {user1\_action.name}. It's a tie!")  
 elif user2\_action in defeats:  
 data = (f"{user1\_action.name} beats {user2\_action.name}!")  
 else:  
 data = (f"{user2\_action.name} beats {user1\_action.name}!")  
 return data  
  
  
sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
sock.bind(('localhost', 6925))  
clients = {}  
game = False  
players = {}  
choice = []  
  
print('Start Server')  
while True:  
 data, addres = sock.recvfrom(1024)  
 print(addres[0], addres[1])  
 data = data.decode('utf-8')  
 if addres not in clients.keys():  
 clients.update({addres: data})  
 print("new connected")  
 print(type(data))  
 data += " connected to this chat"  
 print(data)  
 try:  
 for client in clients.keys():  
 if client == addres:  
 continue  
 print(players)  
 print(choice)  
 if (data == "exit"):  
 print(clients.pop(addres))  
 print('deleted')  
 if (len(players) < 2):  
 if (game and data == "Y" and not (client in players.keys())):  
 players.update({client: False})  
 data = "Start Game \nChoose num: 0-Rock 1-Paper 2-Scissors 3-Lizard 4-Spock"  
 for player in players.keys():  
 data = 'Server: ' + data  
 sock.sendto(data.encode('utf-8'), player)  
 continue  
 elif (game and data == "N"):  
 data = "Game is over"  
 game = False  
 players.clear()  
  
 if (data == "/startRPS" and len(clients) > 1):  
 game = True  
 data = "Сhoose whether you will play Y/N"  
 players.update({client: False})  
 elif (len(players) == 2 and game and not players[addres]):  
 choice.append(data)  
 sock.sendto((f"Your choice is {data} - {Action(int(data)).name}").encode('utf-8'), addres)  
 players[addres] = True  
 data = "I made a choice "  
  
 elif (data == "/stop"):  
 game = False  
 players.clear()  
 choice.clear()  
  
 sock.sendto((str(clients[addres]) + ': ').encode('utf-8') + data.encode('utf-8'), client)  
  
 if (not False in players.keys() and len(choice) == 2):  
 data = determine\_winner(Action(int(choice[0])), Action(int(choice[1])))  
 for player in players.keys():  
 data = 'Server: ' + data + "\n Game is over"  
 sock.sendto(data.encode('utf-8'), player)  
 game = False  
 players.clear()  
 choice.clear()  
 except:  
 continue

Client.py

import socket  
import sys  
import threading  
  
  
def read\_sok():  
 while True:  
 data = sor.recv(1024)  
 print(data.decode('utf-8'))  
  
  
server = 'localhost', 6925  
alias: str = input() # Вводим наш псевдоним  
sor = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
sor.bind(('', 0)) # Задаем сокет как клиент  
sor.sendto((alias).encode('utf-8'), server) # Уведомляем сервер о подключении  
potok = threading.Thread(target=read\_sok)  
potok.start()  
while True:  
 mensahe = input()  
 sor.sendto((mensahe).encode('utf-8'), server)  
 if (mensahe == "exit"):  
 sys.exit()

Tracert IP

