Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

**Лабораторна робота № 3**

по дисципліні «Програмування компютерних та віртуальних мереж інформаційно-комунікаційних систем»

Тема: «Знайомство з протоколом UDP»

|  |  |
| --- | --- |
| Виконав:  студенти групи ІТ-91мн  Давиденко І. В.  Свириденко О. А.  Дата здачі \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Захищено з балом \_\_\_\_\_\_\_\_\_\_\_ | Перевірено:  ас. Галушко Д. О. |

Київ 2020

Завдання

**Хід роботи**

1. Ознайомитись з API для роботи з UDP на Java: DatagramSocket, DatagramPacket (https://www.baeldung.com/udp-in-jav<https://www.baeldung.com/udp-in-java>)
2. Реалізувати наступний клієнт-серверний застосунок, що працює через протокол UDP:

Сервер:

Сервер має вміти «відповідати» на такі запити клієнта:

- сьогоднішня дата;

- поточний час;

- номер та прізвища усіх студентів у бригаді.

Клієнт:

Консольний застосунок, який демонструє функціональність сервера.

**Контрольні питання**

1. Як співвідносяться модель ISO OSI та стек протоколів TCP/IP?
2. Чим TCP відрізняється від UDP?
3. Що таке порт? Що таке сокет?
4. В чому полягає різниця між Socket та DatagramSocket?
5. Що таке блокуючий та неблокуючий ввід-вивід даних?
6. Якими факторами обмежений максимальний розмір UDP-пакету?

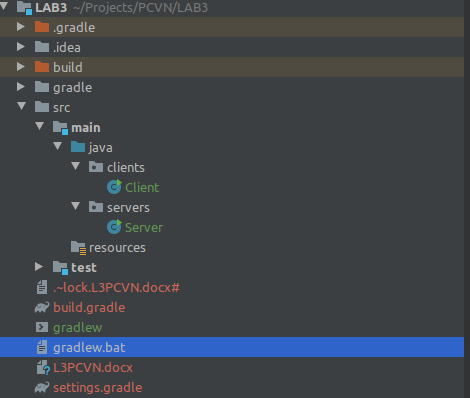
Короткі теоретичні відомості

UDP (англ. User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в RFC 768.

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. При необходимости исправления ошибок на сетевом уровне интерфейса приложение может задействовать TCP или SCTP, разработанные для этой цели.

Хід роботи

Створю проект “LAB3”, котрий має наступну структуру:



Файл Client.java:

package clients;

import servers.Server;

import java.io.Console;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

public class Client {

private DatagramSocket socket;

private InetAddress address;

private byte[] buf;

public Client() throws IOException {

socket = new DatagramSocket();

address = InetAddress.getByName("localhost");

}

public String sendEcho(String msg) {

try {

buf = msg.getBytes();

DatagramPacket packet

= new DatagramPacket(buf, buf.length, address, 4445);

socket.send(packet);

buf = new byte[512];

packet = new DatagramPacket(buf, buf.length);

socket.receive(packet);

String received = new String(

packet.getData(), 0, packet.getLength());

return received;

} catch (IOException exception) {

exception.printStackTrace();

return null;

}

}

public void close() {

socket.close();

}

public static void main(String[] args) {

try {

Client client = new Client();

System.out.println(client.sendEcho("time"));

System.out.println(client.sendEcho("date"));

System.out.println(client.sendEcho("info"));

client.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

Файл Server.java:

package servers;

import javax.imageio.IIOException;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Server extends Thread {

private DatagramSocket socket;

private boolean running;

private byte[] buf = new byte[512];

public Server(int port) throws IOException {

socket = new DatagramSocket(port);

}

public void run() {

running = true;

while (running) {

try {

DatagramPacket packet

= new DatagramPacket(buf, buf.length);

socket.receive(packet);

InetAddress address = packet.getAddress();

int port = packet.getPort();

packet = new DatagramPacket(buf, buf.length, address, port);

String received

= new String(packet.getData(), 0, packet.getLength());

if (received.indexOf("time") == 0) {

SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss z");

Date date = new Date(System.currentTimeMillis());

byte [] buffer = formatter.format(date).getBytes();

packet = new DatagramPacket(buffer, buffer.length, address, port);

}

if (received.indexOf("date") == 0) {

SimpleDateFormat formatter= new SimpleDateFormat("yyyy-MM-dd");

Date date = new Date(System.currentTimeMillis());

byte [] buffer = formatter.format(date).getBytes();

packet = new DatagramPacket(buffer, buffer.length, address, port);

}

if (received.indexOf("info") == 0) {

String info = "IT-91mn Davidenko";

byte [] buffer = info.toString().getBytes();

packet = new DatagramPacket(buffer, buffer.length, address, port);

}

if (received.equals("end")) {

running = false;

continue;

}

socket.send(packet);

} catch (IOException exception) {

exception.printStackTrace();

break;

}

}

socket.close();

}

public static void main(String[] args) {

int port = 4445;

if (args.length >= 1) {

port = Integer.parseInt(args[0]);

}

try {

Server server = new Server(port);

server.run();

} catch (IOException exception) {

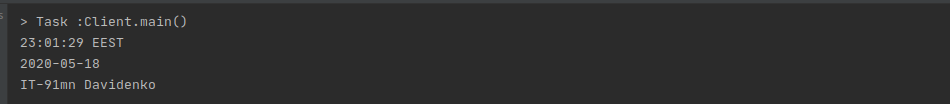
exception.printStackTrace();

}

}

}

В результаті виконання програми отримаю наступне:



Висновок: Під час лабораторної роботи я навчився створювати проект

котрий за допомогою протоколу UDP здійснює відправку повідомлень в реальному часі. Також створив консольну програму котра отримує повідомлення та виводить їх на екран.