



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет
залізничного транспорту ім. академіка В. Лазаряна

Кафедра КІТ

ЛАБОРАТОРНА РОБОТА №11

З дисципліни: «Операційні системи»
на тему: «Створення серверних і клієнтських програм з використанням сокетів»

Виконав: студент групи ПЗ1712
Нікольський О.В.

Прийняла: Нежуміра О.І.

Тема. Створення серверних і клієнтських програм з використанням сокетів.

Мета:

- Ознайомитися з сокетами в UNIX;
- Отримати практичні навички програмування серверних та клієнтських застосунків;
- Отримати практичні навички використання сокетів.

Опис специфікацій

ServerClassTcp			
Методи			
Назва	Завдання	Вхід	Вихід
ServerClassTCP	Конструктор класу; ініціалізація полів класу	int pPort — номер порту для встановлення з'єднання	-
CreateSocket()	Створення серверного сокета на заданому порту	-	Логічне “так” при вдалому створенні або “ні” при помилці
ConnectionClient	Встановлення з'єднання з клієнтом	-	-
GetClientStreams	Отримання потоків вводу- виведення для з клієнтського сокета	-	-
SendArrayToClient	Передача клієнту відсортованого масиву через сокет	int[] array — масив чисел	-
GetArrayFromClient	Отримання масиву від клієнта через сокет	-	-
Sort	Сортування масиву	int[] array — масив чисел	int[] array — відсортований масив чисел
Поля			
Назва	Завдання		
final int PORT	Номер порту, з яким встановлюється з'єднання		
ServerSocket serverSocket	Серверний сокет		
Socket clientSocket	Клієнтський сокет		
InputStream clientInputStream	Поток вводу клієнтського сокета		
OutputStream clientOutputStream	Поток виведення клієнтського сокета		

Код класу ServerClassTCP

```
package com.company;
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Arrays;
public class ServerClassTCP {
    //Methods
```

```

ServerClassTCP(int pPort) {
    this PORT = pPort;
    serverSocket = null;
    clientSocket = null;
    clientInputStream = null;
    clientOutputStream = null;
}
public boolean CreateSocket(){
    try {
        serverSocket = new ServerSocket(PORT);
        System.out.println("Server ready");
        return true;
    } catch (Exception e){
        System.err.println("Port not free. Exception: " + e.getMessage());
        return false;
    }
}
public void ConnectionClient(){
    try{
        clientSocket = serverSocket.accept();
        System.out.println("Client connect");
    } catch (Exception e){
        System.err.println("Error client's connection. Exception: " + e.getMessage());
    }
}
public void GetClientsStreams(){
    try{
        clientInputStream = clientSocket.getInputStream();
        clientOutputStream = clientSocket.getOutputStream();
        System.out.println("Got client's streams");
    } catch (Exception e){
        System.err.println("Error getting client's input stream. Exception: " +
e.getMessage());
    }
}
private int[] Sort(int[] array){
    Arrays.sort(array);
    return array;
}
public void GetArrayFromClient() {
    try {
        ObjectInputStream inputStream = new ObjectInputStream(clientInputStream);
        int[] array = (int[])inputStream.readObject();
        SendArrayToClient(array);
    } catch (Exception e){
        System.err.println("Getting array error. Exception: " + e.getMessage());
    }
}
private void SendArrayToClient(int[] array){
    try{
        ObjectOutputStream outputStream = new ObjectOutputStream(clientSocket.getOutputStream());
        outputStream.writeObject(Sort(array));
    } catch (Exception e){
        System.err.println("Sending array error. Exception: " + e.getMessage());
    }
}
// Fields
private final int PORT;
private ServerSocket serverSocket;
private Socket clientSocket;

```

```

private InputStream clientInputStream;
private OutputStream clientOutputStream;
}

```

ClientClassTcp			
Методи			
Назва	Завдання	Вхід	Вихід
ClientClassTCP	Конструктор класу; ініціалізація полей класу	String pHost — хост, з яким встановлюватиметься з'єднання; int pPort — номер порту для встановлення з'єднання	-
CreateSocket()	Створення та з'єднання сокету з заданим хостом на заданому порту	-	Логічне “так” при вдалому створенні або “ні” при помилці
SendArrayToServer	Передача серверу масиву через сокет	int[] array — масив чисел	-
GetArrayFromServer	Отримання відсортованого масиву від сервера через сокет	-	-
Поля			
Назва	Завдання		
final int PORT	Номер порту, з яким встановлюється з'єднання		
final String HOST	Хост, з яким встановлюється з'єднання		
Socket socket	Клієнтський сокет		

Код класу ClientClassTCP

```

package com.company;
import java.io.*;
import java.net.Socket;
public class ClientClassTCP {
    //Methods
    ClientClassTCP(String pHost, int pPort){
        this.HOST = pHost;
        this.PORT = pPort;
    }
    public boolean CreateSocket(){
        try{
            socket = new Socket(HOST, PORT);
            return true;
        } catch (Exception e){
            System.err.println("Can't create socket. Exception: " + e.getMessage());
            return false;
        }
    }
}

```

```

public void SendArrayToServer(int []array){
    try {
        ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());
        outputStream.writeObject(array);
        GetArrayFromServer();
    } catch (Exception e){
        e.printStackTrace();
    }
}

private void GetArrayFromServer(){
    try {
        ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());
        int []array = (int [])inputStream.readObject();
        for(int i = 0; i < array.length; ++i){
            System.out.println(array[i]);
        }
    } catch (Exception e){
        e.printStackTrace();
    }
}

//Fields
private final String HOST;
private final int PORT;
private Socket socket = null;
}

```

Аналіз результатів та висновки

У ході виконання лабораторної роботи ознайомився з використанням сокетів UNIX.

Для реалізації завдання була обрана мова програмування Java, т.я. вона є кросплатформеною та має зручний набір інструментів для роботи з сокетами.