

## Michał Szczepaniak

### Zadanie 1

Do serwera dodałem linijki służące do pobrania adresu klienta oraz wysłanie na ten adres odpowiedzi:

```
25 InetSocketAddress clientAddress = receivePacket.getAddress();
26 int clientPort = receivePacket.getPort();
27
28 byte[] responseBuffer = "UDP Java server response".getBytes();
29 DatagramPacket responsePacket = new DatagramPacket(responseBuffer, responseBuffer.length, clientAddress, clientPort);
30 socket.send(responsePacket);
```

Do klienta dodałem linijki które odbierają wiadomość od serwera:

```
21 byte[] receiveBuffer = new byte[1024];
22
23 DatagramPacket serverResponse = new DatagramPacket(receiveBuffer, receiveBuffer.length);
24 socket.receive(serverResponse);
25
26 String msg = new String(serverResponse.getData());
27 System.out.println("received msg: " + msg);
```

Działanie serwer:

```
C:\Michał\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z1JavaUdpServer.java
JAVA UDP SERVER
received msg: Ping Java Udp
```

Działanie klient:

```
C:\Michał\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z1JavaUdpClient.java
JAVA UDP CLIENT
received msg: UDP Java server response
```

### Zadanie 2

W serwerze dodałem do zamieniania wiadomości na string standard UTF 8:

```
23 String msg = new String(receivePacket.getData(), StandardCharsets.UTF_8);
```

W kliencie zmieniłem port na 9008 oraz przy zamienianiu na bajty dodałem kodowanie UTF 8:

```
6 serverPort = 9008
7
8 msg = "żółta gęś"
9
10 print('PYTHON UDP CLIENT')
11 client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
12 client.sendto(bytes(msg, 'utf-8'), (serverIP, serverPort))
```

Działanie serwer:

```
C:\Michal\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z2JavaUdpServer.java
JAVA UDP SERVER
received msg: żółta gęś
```

Działanie klient:

```
C:\Michal\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>python Z2PythonUdpClient.py
PYTHON UDP CLIENT
```

## Zadanie 3

W serwerze dodałem zamienianie wiadomości na int z uwzględnieniem kolejności bajtów oraz wysyłam z powrotem do klienta liczbę podwyższoną o jeden ( $nb + 1$ ):

```
20 while(true) {
21     Arrays.fill(receiveBuffer, (byte)0);
22     DatagramPacket receivePacket = new DatagramPacket(receiveBuffer, receiveBuffer.length);
23     socket.receive(receivePacket);
24
25
26     int nb = ByteBuffer.wrap(receivePacket.getData()).order(java.nio.ByteOrder.LITTLE_ENDIAN).getInt();
27     System.out.println(nb);
28
29     InetAddress clientAddress = receivePacket.getAddress();
30     int clientPort = receivePacket.getPort();
31
32     byte[] b = ByteBuffer.allocate(4).putInt(nb + 1).array();
33
34     DatagramPacket responsePacket = new DatagramPacket(b, b.length, clientAddress, clientPort);
35     socket.send(responsePacket);
36 }
```

W kliencie przy odebraniu wiadomości zamieniam z na int z uwzględnieniem kolejności bajtów, gdzie Java przesłała z kodowaniem Big Endian:

```
3 serverIP = "127.0.0.1"
4 serverPort = 9008
5 # msg = "Ping Python Udp!"
6 msg_bytes = (300).to_bytes(4, byteorder='little')
7
8 print('PYTHON UDP CLIENT')
9 client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10 client.sendto(msg_bytes, (serverIP, serverPort))
11
12 buff, address = client.recvfrom(1024)
13 print(int.from_bytes(buff, byteorder='big'))
```

Działanie serwer:

```
C:\Michał\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z3JavaUdpServer.java
JAVA UDP SERVER
300
```

Działanie klient:

```
C:\Michał\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>python Z3PythonUdpClient.py
PYTHON UDP CLIENT
301
```

## Zadanie 4

W serwerze pierwsza litera decyduje czy wiadomość przyszła z serwera Python czy Java, jeżeli pierwsza litera wiadomości to "P" to jest to wiadomość z serwera Python, jeżeli pierwsza litera to "J" to jest wiadomość z serwera Java:

```
25         InetAddress address = receivePacket.getAddress();
26         int port = receivePacket.getPort();
27         byte[] responseBuffer;
28
29         if (msg.length() > 0 && msg.charAt(0) == 80) { // Python
30             responseBuffer = "Ping Python".getBytes();
31         } else if (msg.length() > 0 && msg.charAt(0) == 74) { // Java
32             responseBuffer = "Ping Java".getBytes();
33         } else { // Unknown
34             responseBuffer = "Ping Unknown".getBytes();
35         }
36
37         DatagramPacket responsePacket = new DatagramPacket(responseBuffer, responseBuffer.length, address, port);
38         socket.send(responsePacket);
```

Klient Java - odbieranie wiadomości jak wcześniej:

```
21         byte[] receiveBuffer = new byte[1024];
22
23         DatagramPacket serverResponse = new DatagramPacket(receiveBuffer, receiveBuffer.length);
24         socket.receive(serverResponse);
25
26         String msg = new String(serverResponse.getData());
27         System.out.println("received msg: " + msg);
```

Klient Python - odbieranie wiadomości jak wcześniej:

```
11     buff, address = client.recvfrom(1024)
12     print("received msg: " + str(buff, 'cp1250'))
```

#### Działanie serwera:

```
C:\Michal\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z4JavaUdpServer.java
JAVA UDP SERVER
received msg: Ping Java Udp

received msg: Ping Python Udp!
```

#### Działanie klienta Java:

```
C:\Michal\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>java Z4JavaUdpClient.java
JAVA UDP CLIENT
received msg: Ping Java
```

#### Działanie klienta Python:

```
C:\Michal\studia\Semestr_6\Systemy_Rozproszone\Lab_1\lab_gniazda>python Z4PythonUdpClient.py
PYTHON UDP CLIENT
received msg: Ping Python
```