

## Računarske mreže – April Domaći zadaci

1.

HTTP protokol

- a) ima stanje, ima konekciju
- b) ima stanje, nema konekciju
- c) nema stanje, ima konekciju
- d) nema stanje, nema konekciju**

2.

```
header[9]=(byte)(Ssrc>>16);
```

- a) Izvlači treći bajt iz Ssrc varijable i smešta u RTP heder**
- b) Izvlači četvrti bajt iz Ssrc varijable i smešta u RTP heder
- c) Izvlači prvi bajt iz Ssrc varijable i smešta u RTP heder
- d) Izvlači drugi bajt iz Ssrc varijable i smešta u RTP heder

Zato što ga ">>" pomera za 16 bita tj. dva bajta desno, pa treći bajt dolazi tamo gde je ranije bio prvi.

Na primer:

```
Ssrc:  
00000000 11111111 11000000 11100000  
Ssrc>>16  
00000000 00000000 00000000 11111111
```

3.

Komande po RTSP protokolu su

- a) INIT, PLAY, PAUSE, STOP
- b) SETUP, PLAY, PAUSE, STOP
- c) SETUP, PLAY, PAUSE, TEARDOWN**

*Izvor:*

[http://en.wikipedia.org/wiki/Rtsp#RTSP\\_commands](http://en.wikipedia.org/wiki/Rtsp#RTSP_commands)

Sve RTSP komande: DESCRIBE, SETUP, PLAY, PAUSE, RECORD, TEARDOWN

4.

```
byte[] data = new byte[1024];  
InetAddress IPAddress = InetAddress.getByName("localhost");  
DatagramPacket packet = new DatagramPacket(data, data.length, IPAddress, 9876);
```

- a) deklarise varijablu packet kao TCP paket za prijem
- b) deklarise varijablu packet kao UDP paket za slanje**
- c) deklarise varijablu packet kao UDP paket za prijem
- d) deklarise varijablu packet kao TCP paket za slanje

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramPacket.html#DatagramPacket\(byte\[\],%20int,%20java.net.InetAddress,%20int\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramPacket.html#DatagramPacket(byte[],%20int,%20java.net.InetAddress,%20int))

5.

```
//fill by default header fields:  
Version = 2;  
header[0]=(byte)(header[0] | Version<<6);
```

- a) Shiftuje u levo za 6 pozicija vrednost Version polja
- b) Shiftuje u desno za 6 pozicija vrednost Version polja
- c) U polje Version, RTP hedera upisuje 2**
- d) U polje Version, RTP hedera upisuje 6

6.

Ako pokrenemo web server, čiji je kod dat, šta će biti poslato kao odgovor, na sledeći HTTP zahtev:  
POST /flower.gif HTTP 1.0

```
// kod servera
import java.io.*;
import java.net.*;
import java.util.*;
class WebServer {
    public static void main(String argv[]) throws Exception
    {
        System.out.println("Web Server up and running...");
        String requestMessageLine;
        String fileName;
        ServerSocket listenSocket = new ServerSocket(6789);
        while(true) {
            Socket connectionSocket = listenSocket.accept();
            System.out.println("Web server received request...");
            BufferedReader inFromClient =
                new BufferedReader(new InputStreamReader(
                    connectionSocket.getInputStream()));
            DataOutputStream outToClient =
                new DataOutputStream(
                    connectionSocket.getOutputStream());
            requestMessageLine = inFromClient.readLine();
            StringTokenizer tokenizedLine =
                new StringTokenizer(requestMessageLine);
            if (tokenizedLine.nextToken().equals("GET")){
                fileName = tokenizedLine.nextToken();
                if (fileName.startsWith("/") == true)
                    fileName = fileName.substring(1);
                File file = new File(fileName);
                int numOfBytes = (int) file.length();
                FileInputStream inFile = new FileInputStream(fileName);
                byte[] fileInBytes = new byte[numOfBytes];
                inFile.read(fileInBytes);
                outToClient.writeBytes("HTTP/1.0 200 Document Follows\r\n");
                if (fileName.endsWith(".jpg"))
                    outToClient.writeBytes("Content-Type: image/jpeg\r\n");
                if (fileName.endsWith(".gif"))
                    outToClient.writeBytes("Content-Type: image/gif\r\n");
                outToClient.writeBytes("Content-Length: " + numOfBytes + "\r\n");
                outToClient.writeBytes("\r\n");
                outToClient.write(fileInBytes, 0, numOfBytes);
                connectionSocket.close();
            }
            else System.out.println("Bad Request Message");
        }
    }
}
```

a) resurs flower.gif kao ASCII enkodiran sadržaj

**b) Bad Request Message**

c) resurs flower.gif kao binarni sadržaj

7.

Kod video striminga, na transportnom sloju se koristi

- a) UDP za RTSP i TCP za RTP
- b) TCP za RTP i RTSP
- c) UDP za RTP i RTSP
- d) UDP za RTP i TCP za RTSP**

*Izvor:*

<http://en.wikipedia.org/wiki/Rtsp>

[http://en.wikipedia.org/wiki/Real-time\\_Transport\\_Protocol](http://en.wikipedia.org/wiki/Real-time_Transport_Protocol)

Generalno ne mora da bude tako, ali se u praksi najčešće koriste na ovaj način

8.

Šta radi sledeća komanda (JAVA):

```
DatagramSocket clientSocket = new DatagramSocket();
```

```
DatagramPacket packet;
```

```
clientSocket.send(packet);
```

- a) prima paket preko TCP protokola
- b) prima paket preko UDP protokola
- c) šalje paket preko TCP protokola
- d) šalje paket preko UDP protokola**

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramSocket.html#send\(java.net.DatagramPacket\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramSocket.html#send(java.net.DatagramPacket))

9.

```
Socket connectionSocket = welcomeSocket.accept();
```

- a) isto što i nslookup localhost
- b) uspostavlja TCP konekciju
- c) započinje osluškivanje zahteva za TCP konekcijom**
- d) Inicijalizuje TCP serverski soket

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/ServerSocket.html#accept\(\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/ServerSocket.html#accept())

Ovde je greška što welcomeSocket nije definisan, ali pogledaj pitanje 12 pa ćeš videti da je on ServerSocket.

10.

Ako pokrenemo web server, čiji je kod dat, šta će biti poslato kao odgovor, na sledeći HTTP zahtev:

GET /flower.gif HTTP 1.0

```
// kod servera
import java.io.*;
import java.net.*;
import java.util.*;
class WebServer {
    public static void main(String argv[]) throws Exception
    {
        System.out.println("Web Server up and running...");
        String requestMessageLine;
        String fileName;
        ServerSocket listenSocket = new ServerSocket(6789);
        while(true) {
            Socket connectionSocket = listenSocket.accept();
            System.out.println("Web server received request...");
            BufferedReader inFromClient =
                new BufferedReader(new InputStreamReader(
                    connectionSocket.getInputStream()));
            DataOutputStream outToClient =
                new DataOutputStream(
                    connectionSocket.getOutputStream());
            requestMessageLine = inFromClient.readLine();
            StringTokenizer tokenizedLine =
                new StringTokenizer(requestMessageLine);
            if (tokenizedLine.nextToken().equals("GET")){
                fileName = tokenizedLine.nextToken();
                if (fileName.startsWith("/") == true)
                    fileName = fileName.substring(1);
                File file = new File(fileName);
                int numOfBytes = (int) file.length();
                FileInputStream inFile = new FileInputStream(fileName);
                byte[] fileInBytes = new byte[numOfBytes];
                inFile.read(fileInBytes);
                outToClient.writeBytes("HTTP/1.0 200 Document Follows\r\n");
                if (fileName.endsWith(".jpg"))
                    outToClient.writeBytes("Content-Type: image/jpeg\r\n");
                if (fileName.endsWith(".gif"))
                    outToClient.writeBytes("Content-Type: image/gif\r\n");
                outToClient.writeBytes("Content-Length: " + numOfBytes + "\r\n");
                outToClient.writeBytes("\r\n");
                outToClient.write(fileInBytes, 0, numOfBytes);
                connectionSocket.close();
            }
            else System.out.println("Bad Request Message");
        }
    }
}
```

**a) resurs flower.gif kao binarni sadržaj**

b) resurs flower.gif kao ASCII enkodiran sadržaj

c) neće biti nikakvog odgovora

d) Bad Request Message

Za ovo ne mogu 100% da garantujem. Sigurno se vraća resurs flower.gif, ali nisam siguran da li kao binarni ili ASCII enkodiran, nisam mogao nigde da nađem odgovor. Ja bih stavio pod a) kad bih dobio ovo pitanje, ali proveriti još negde :)

11.

Navedite tačan redoslet SMTP komandi

- a) HELO, DATA, MAIL FROM, QUIT, RCPT TO
- b) HELO, MAIL FROM, QUIT, RCPT TO, DATA
- c) DATA, HELO, MAIL FROM, QUIT, RCPT TO
- d) HELO, MAIL FROM, RCPT TO, DATA, QUIT**

*Izvor:*

[http://en.wikipedia.org/wiki/Smtp#Sample\\_communications](http://en.wikipedia.org/wiki/Smtp#Sample_communications)

12.

```
ServerSocket welcomeSocket = new ServerSocket(6789);
```

- a) Inicijalizuje TCP serverski socket**
- b) započinje osluškivanje zahteva za TCP konekcijom
- c) pokreće prijem podataka na serveru putem TCP protokola

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/ServerSocket.html#ServerSocket\(int\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/ServerSocket.html#ServerSocket(int))

13.

```
InetAddress IPAddress = InetAddress.getByName("localhost");
```

- a) isto što i nslookup localhost
- b) generiše ime hosta za datu IP adresu
- c) generiše DNS upit i dovlači IP adresu hosta localhost**
- d) kreira localhost interfejs

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/InetAddress.html#getByName\(java.lang.String\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/InetAddress.html#getByName(java.lang.String))

14.

```
DatagramSocket serverSocket = new DatagramSocket (9876);
```

- a) priprema UDP socket za slanje podataka
- b) priprema UDP socket za prijem podataka**
- c) priprema TCP socket za slanje podataka
- d) priprema TCP socket za prijem podataka

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramSocket.html#DatagramSocket\(int\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/DatagramSocket.html#DatagramSocket(int))

15.

```
Socket socket = new Socket("localhost", 6789);
```

- a) instancira TCP klijentski socket**
- b) instancira UDP klijentski socket
- c) instancira UDP serverski socket
- d) instancira TCP serverski socket

*Izvor:*

[http://java.sun.com/j2se/1.4.2/docs/api/java/net/Socket.html#Socket\(java.lang.String,%20int\)](http://java.sun.com/j2se/1.4.2/docs/api/java/net/Socket.html#Socket(java.lang.String,%20int))