

# Rešenja prvog kolokvijuma iz Operativnih sistema 2

## Decembar 2016.

### 1. (10 poena)

a)(7) HP, MP, LP, MP, MP, LP, MP, LP, LP

b)(3)  $\tau = 6$

### 2. (10 poena)

```
public class Computer {

    public synchronized void writeX (double v) {
        while (this.state!=readyForX) wait();
        this.x = v;
        this.state = readyForY;
        notifyAll();
    }

    public synchronized void writeY (double v) {
        while (this.state!=readyForY) wait();
        this.y = v;
        this.state = readyToRead;
        notifyAll();
    }

    public synchronized double read () {
        while (this.state!=readyToRead) wait();
        double temp = this.x + this.y;
        this.state = readyForX;
        notifyAll();
        return temp;
    }

    private double x, y;
    private static final int readyForX=0, readyForY=1, readyToRead=2;
    private int state = readyForX;

};
```

### 3. (10 poena)

```
public class Posrednik {
    private static final int port = 6000;
    private class Nit extends Thread {
        private Socket klijentSocket;
        private Socket serverSocket;
        public Nit(Socket klijent) throws IOException {
            this.klijentSocket = klijent;
        }
        public void run() {
            try {
                Komunikacija klijent = new Komunikacija(klijentSocket);
                String ip = klijent.primi();
                String port = klijent.primi();
                serverSocket = new Socket(ip, Integer.parseInt(port));
                Komunikacija server = new Komunikacija(serverSocket);
                while (!klijent.kraj() && !server.kraj()) {
                    String poruka = klijent.primi();
                    if (poruka == null) {
                        break;
                    }
                    server.posalji(poruka);
                    poruka = server.primi();
                }
            }
        }
    }
}
```

```

        if (poruka == null) {
            break;
        }
        klijent.posalji(poruka);
    }
    klijent.zatvori();
    server.zatvori();
} catch (IOException e) {
    e.printStackTrace();
}
System.out.println("finish");
}
}
public void work () throws IOException {
    ServerSocket socket = new ServerSocket(port);
    while (true) {
        Socket klijent = socket.accept();
        new Nit(klijent).start();
    }
}
public static void main(String args[]) throws IOException {
    Posrednik posrednik = new Posrednik();
    posrednik.work();
}
}

public class Komunikacija {
    private Socket socket;
    private PrintWriter izlaz;
    private BufferedReader ulaz;
    public Komunikacija(Socket socket) throws IOException {
        this.socket = socket;
        izlaz = new PrintWriter(socket.getOutputStream(), true);
        ulaz = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
    }
    public void posalji(String poruka) {
        izlaz.println(poruka);
    }
    public String primi() throws IOException {
        String poruka = ulaz.readLine();
        return poruka;
    }
    public boolean kraj() {
        return socket.isClosed();
    }
    public void zatvori() throws IOException {
        izlaz.close();
        ulaz.close();
        socket.close();
    }
}
}

```