Rešenja prvog kolokvijuma iz Operativnih sistema 2 Novembar 2013.

1. (10 poena)

```
void Scheduler::put (PCB* pcb, int wasBlocked) {
  if (pcb==0) return; // Exception!
  if (pcb==idle) return; // Idle process is not put in the queue
  int i=0;
  if (wasBlocked) {
    pcb->tau=(pcb->tau+pcb->time)>>1;
    pcb->time = 0;
    if (pcb->tau<=tauHP) i=0;
    else
    if (pcb->tau<=tauMP) i=1;
    else
    i=2;
  } else {
    i = pcb->lastQueue;
    if (i<2) i++;
  }
  pcb->next = 0;
  if (tail[i]==0)
    tail[i] = head[i] = pcb;
  else
    tail[i] = tail[i]->next = pcb;
}
```

2. (10 poena)

```
public class Buffer {
 protected List l = new LinkedList();

 public synchronized void put(Object o){
  l.add(o);
  notify();
 }

 public synchronized Object get(){
  while (l.isEmpty())try {wait();} catch (InterruptedException e) {}
  return l.remove();
 }
```

3. (10 poena)

Na serverskoj strani u klasi `Server` treba dodati sledeće atribute:

```
 protected File rootDir;

public Server(int port, String rootDir) {
 this.rootDir = new File(rootDir);
 ...

//poziv konstruktora new RequestHandler(clientSocket, rootDir);

public static void main(String args[]) {
 Server s = new Server(6001, "/home/...");
 s.start();
 ...
```

Klasu `RequestHandler` treba izmeniti na sledeći način:

```
public class RequestHandler extends Thread {
 ...
 protected File rootDir;
```

```
 ...
public RequestHandler(Socket clientSocket, File rootDir) {
  this.sock = clientSocket;
  this.rootDir = rootDir;
  ...
}

protected void processRequest(String request) {
  StringTokenizer st = new StringTokenizer(request, "#");
  String functionName = st.nextToken();
  String filePath = st.nextToken();
  try {
   if (functionName.equals("index")) sendIndex(filePath);
   else if (functionName.equals("get")) sendFile(filePath);
  } catch (Exception e) {//...}
}

protected void sendIndex(String filePath) throws Exception {
 File file = new File(rootDir+"/"+filePath);
 if ((!file.exists()) || !file.isDirectory()) sendMessage("#error#");
 else {
  String[] fileList = file.list();
  for (int i = 0; i < fileList.length; i++) sendMessage(fileList[i]);
 }
 sendMessage("#done#");
}

protected void sendFile(String filePath) throws Exception {
 File file = new File(rootDir+"/"+filePath);
 if ((!file.exists()) || file.isDirectory()) sendMessage("#error#");
 else {
  BufferedReader fileIn = new BufferedReader(new FileReader(file));
  while (true) {
   String line = fileIn.readLine();
   if (line == null) { sendMessage("#done#");break; }
   else sendMessage(line);
  }
 }
}
```

Na klijentskoj strani:

```
public class FileClient extends Usluga {
 public FileClient(String host, int port) {
  super(host, port);
 }

 public String getFile(String file) {
  String message = "#get#" + file;
  sendMessage(message);
  StringBuffer content = new StringBuffer();
  while (true) {
   message = receiveMessage();
   if(message.equals("#done#") || message.equals("#error#")) break;
   content.append(message+"\r\n");
  }
  return content.toString();
 }

 public ArrayList<String> getIndex(String dir) {
  String message = "#index#" + dir;
  sendMessage(message);
  ArrayList<String> l = new ArrayList<String>();
  while (true) {
   message = receiveMessage();
   if(message.equals("#done#") || message.equals("#error#")) break;
```

```
    l.add(message);
   }
   return l;
  }
 }
```