Rešenja prvog kolokvijuma iz Operativnih sistema 2
Novembar 2012.

1. (10 poena)

```
const int timesliceHP = ..., timesliceMP = ..., timesliceLP = ...;
extern PCB * const idle;

class Scheduler {
public:
  Scheduler ();
  PCB* get ();
  void put (PCB*);
private:
  PCB* head[3];  // 0-HP, 1-MP, 2-LP
  PCB* tail[3];
  static const int timeslice[3];
};

const int Scheduler::timeslice[] = {timesliceHP,timesliceMP,timesliceLP};

Scheduler::Scheduler () {
  for (int i=0; i<2; i++)
    head[i]=tail[i]=0;
}

PCB* Scheduler::get () {
  for (int i=0; i<2; i++)
    if (head[i]) {
      PCB* ret = head[i];
      head[i] = head[i]->next;
      if (head[i]==0) tail[i]=0;
      ret->next = 0;
      ret->timeslice=timeslice[i];
      return ret;
    }
  return idle;
}

void Scheduler::put (PCB* pcb) {
  if (pcb==0) return; // Exception!
  int i=0;
  if (pcb->tau<=5) i=0;
  else
  if (pcb->tau<=10) i=1;
  else i=2;
  pcb->next = 0;
  if (tail[i]==0)
    tail[i] = head[i] = pcb;
  else
    tail[i] = tail[i]->next = pcb;
}
```

2. (10 poena)

```
monitor Semaphore;
export wait, signal;

var val : integer;

procedure wait ();
begin
  while val<=0 do wait();
  val := val - 1;
end;

procedure signal ();
begin
  val := val + 1;
  notifyAll();
end;

begin
  val := 1;
end; (* Semaphore *)
```

3. (10 poena)

```
public class NetResource extends Usluga {
 public NetResource(String host, int port, int resId) {
  super(host, port);
  id = resId;
 }

 private int id;
 public static final int RN = ...;
 private static boolean[] allocatedResources = new boolean[RN];

 public void alloc() throws DeadlockPreventionException {
  for (int i = id; i < allocatedResources.length; i++) {
   if(allocatedResources[i]) throw new DeadlockPreventionException(id);
  }
  String message = "#alloc#" + id + "#";
  sendMessage(message);
  receiveMessage();
  allocatedResources[id]=true;
 }

 public void dealloc() {
  String message = "#dealloc#" + id + "#";
  sendMessage(message);
  receiveMessage();
  allocatedResources[id]=false;
 }
}
```

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

```
public final int RN;
public Resource[] resources;

public Server(int port, int RN){
 this.RN=RN;
 resources = new Resource[RN];
 for (int i = 0; i < resources.length; i++) {
  resources[i]=new Resource();
 }
 ...

//poziv konstruktora new RequestHandler(clientSocket, resources);
```

`RequestHandler` treba izmeniti na sledeći način:

```
public class RequestHandler extends Thread {
 ...
 Resource[] resources;
 ...

public RequestHandler(Socket clientSocket, Resource[] resources) {
  this.sock = clientSocket;
  this.resources = resources;
  ...
}

protected void processRequest(String request) {
 StringTokenizer st = new StringTokenizer(request, "#");
 String functionName = st.nextToken();
 int resId = Integer.parseInt(st.nextToken());

 if (functionName.equals("alloc")) {
      resources[resId].alloc();
      sendMessage("allocated");
 } else if (functionName.equals("dealloc")){
      resources[resId].dealloc();
      sendMessage("deallocated");
 }
}
```