

# Rešenja prvog kolokvijuma iz Operativnih sistema 2

## Oktober 2014.

1. (10 poena) HP, MP, HP, HP, MP, HP, MP, LP, MP, LP

2. (10 poena)

```
monitor TaxiDispatcher;
  export userRequest, taxiAvailable;

  var waitingUsers, availableTaxis : integer;
      waitForUser, waitForTaxi : condition;

  procedure userRequest ();
  begin
    if (availableTaxis>0)
    begin
      availableTaxis:=availableTaxis-1;
      waitForUser.signal;
    end
  else
  begin
    waitingUsers:=waitingUsers+1;
    waitForTaxi.wait;
  end;
end;

  procedure taxiAvailable ();
  begin
    if (waitingUsers>0)
    begin
      waitingUsers:=waitingUsers-1;
      waitForTaxi.signal;
    end
  else
  begin
    availableTaxis:=availableTaxis+1;
    waitForUser.wait;
  end;
end;

begin
  waitingUsers:=0; availableTaxis:=0;
end;
```

3. (10 poena)

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

```
public static boolean kraj = false;
protected QueryExecutor executor;

public Server(int port, QueryExecutor executor) {
  this.executor = executor;
  ...

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

public static void main(String args[]) {
  QueryExecutor executor = new QueryExecutor();
  Server s = new Server(6001,executor);
  s.start();
  executor.start();
}
```

...

**klasu RequestHandler treba izmeniti na sledeći način:**

```
public class RequestHandler extends Thread {
    ...
    protected QueryExecutor executor;
    ...

    public RequestHandler(Socket clientSocket, QueryExecutor executor) {
        this.sock = clientSocket;
        this.executor = executor;
        ...
    }

    protected void processRequest(String request) {
        StringTokenizer st = new StringTokenizer(request, "#");
        String sqlQuery = st.nextToken();
        int waitTime = Integer.parseInt(st.nextToken());
        executor.put(new SQLWorkerThread(sqlQuery, waitTime, out));
    }

    public class QueryExecutor extends Thread {
        protected Queue<SQLWorkerThread> fifoQueue = new Queue<SQLWorkerThread>();
        protected DelayQueue<SQLWorkerThread> delayQueue = new
        DelayQueue<SQLWorkerThread>();

        private synchronized SQLWorkerThread get() {
            while (delayQueue.isEmpty() && fifoQueue.isEmpty()) wait();
            SQLWorkerThread nextWT = delayQueue.peek();
            if (nextWT.getDelay() > 0)
                nextWT = fifoQueue.peek();
            delayQueue.remove(nextWT);
            fifoQueue.remove(nextWT);
            return nextWT;
        }

        public synchronized void put(SQLWorkerThread wt) {
            fifoQueue.add(wt);
            delayQueue.add(wt);
            notifyAll();
        }

        public void run() {
            while (!Server.kraj) {
                SQLWorkerThread nextWT = get();
                nextWT.start();
            }
        }
    }
}
```