Rešenja kolokvijuma iz Operativnih sistema 2, februar 2021.

1. (10 poena)

```
public class Server {
    private static final int N = 50;
    private boolean[] cards = new boolean[N];
    private Deque<RequestHandler> users = new ArrayDeque<RequestHandler>();

    public Server () {
        for (int i = 0; i < N; i++) {
            cards[i] = true;
        }
    }

    public synchronized int[] getCards(RequestHandler user, int num) throws
InterruptedException {
        int[] ret;
        while (user != users.peekFirst() || (ret = tryGetCards(num)) ==
null) {
            wait();
        }

        users.getFirst();

        for (int i = 0; i < num; i++) {
            cards[ret[i]] = false;
        }

        notifyAll();

        return ret;
    }

    public synchronized void returnCards(int[] ret) throws
InterruptedException {
        for (int i = 0; i < ret.length; i++) {
            cards[ret[i]] = true;
        }

        notifyAll();
    }

    private int[] tryGetCards(int num) {
        int[] ret = new int[num];
        int n = 0;
        int x = 0;
        while (cards[x] && x < N - 1) {
            x++;
        }
        for (int i = 0; i < N; i++) {
            if (!cards[x]) {
                n = 0;
            } else {
                ret[n] = x;
                n++;
            }
            x = (x + 1) % N;
```

```
            if (n == num) {
                return ret;
            }
        }

        return null;
    }

    public void run() {
        ServerSocket serverSocket = null;

        try {
            serverSocket = new ServerSocket(5555);

            while (true) {
                Socket clientSocket = clientSocket = serverSocket.accept();

                ServerService clientService = new
ServerService(clientSocket);

                RequestHandler user = new RequestHandler(clientService,
this);
                users.addLast(user);
                user.start();
            }

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (serverSocket != null) {
                try {
                    serverSocket.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public static void main(String args[]) {
        Server server = new Server();

        server.run();
    }

    public String runExecutionOnCards(String commands, int[] cards) {
        ...
    }
}

public class RequestHandler extends Thread {
    private final ServerService service;

    private final Server server;
    public RequestHandler(ServerService service, Server server) {
        this.server = server;
        this.service = service;
    }

    public void run() {
        try {
```

```
            int numOfCards = service.getNumOfCards();
            String commands = service.getCommands();

            int cards[] = server.getCards(this, numOfCards);

            String result = server.runExecutionOnCards(commands, cards);

            service.sendResult(result);

        } catch (IOException e) {
            e.printStackTrace();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } finally {
            try {
                service.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

public class ServerService extends Service {
    public ServerService(Socket socket) throws IOException {
        super(socket);
    }

    public void sendResult(String msg) {
        sendMessage(msg.replaceAll("\n", "@"));
    }

    public String getCommands() throws IOException {
        return receiveMessage().replaceAll("@", "\n");
    }

    public int getNumOfCards() throws IOException {
        return Integer.parseInt(receiveMessage());
    }
}
```

Klasa Service je data na vežbama.

2. (10 poena) a)(5) Problem je potencijalno izgladnjivanje.

```
int allocate (unsigned res) {
  static const unsigned long sleep_time = ...;
  int stat = alloc(res);
  while (stat==0) {
    sleep(rnd(sleep_time)+1);
    stat = alloc(res);
  }
  return stat;
}
```

b)(5)

```
#include <stdlib.h>

int compare_ints (const void* p, const void* q) {
  int x = *(const int*)p;
  int y = *(const int*)q;
  return (x > y) - (x < y);
}

void sort_ints (unsigned* a, size_t n) {
  qsort(a,n,sizeof(*a),compare_ints);
}

int allocate (unsigned res[], size_t n) {
  sort_ints(res,n);
  for (int i=0; i<n; i++) {
    if (allocate(i)<0) {
      for (int j=0; j<i; j++) release(j);
      return -1;
    }
  }
  return 0;
}
```

3. (10 poena)

```
void clock () {
  while (true) {
    unsigned long pgDesc = *frames[clockHand].pgDesc;
    if (!(pgDesc & 1)) return;
    *frames[clockHand].pgDesc = pgDesc & ~1UL;
    clockHand = frames[clockHand].next;
  }
}
```

4. (10 poena)

```
#!/bin/bash

if [ $# -ne 1 -o ! -r $1 ]; then
      echo "Error: First parameter must be a readable file"
      exit 1
fi

OLD_IFS=$IFS

IFS=$'\n'

REGEX='.*(\([^\.]*\)\.\([^\,]*\)\,\(.*\)\,\(0x[0-9a-f]*\))$'
for i in $(cat $1); do

      name=$(echo $i | sed "s:$REGEX:\1.\2:")
      size=$(echo $i | sed "s:$REGEX:\3:")

      echo "{uint${size}_t __reg;"
      echo $i | sed "s:$REGEX"':read_reg("\1", "\2"'", \4, \&__reg);:"
      echo "printf(\"$name=0x\"PRIx$size\"\\n\", __reg);}"

done

IFS=$OLD_IFS
```