

Rešenja trećeg kolokvijuma iz Operativnih sistema 2, septembar 2014.

1. (10 poena)

```
int mapRAIDBlock(unsigned long blk, unsigned long& d, unsigned long& b) {
    static const int dsks = getNumOfDisks(), blks = getNumOfBlocks();
    d = blk%dsks;
    b = blk/dsks;
    if (b>=blks) return -1; else return 0;
}
```

2. (10 poena)

```
#!/bin/bash
```

```
if [ $# -ne 2 ];then
    echo "Greska: broj argumenata neodgovarajuci"
    echo "Pozvati sa script.sh file address"
    exit 1
fi

file=$1
address=$2

if [ -f "$file" -a -r "$file" ]; then
    cat $file | grep "From:.*<$address>" > /dev/null
    if [ $? -eq 0 ]; then
        cat $file | grep "Content-Type:.*image" | \
            sed 's/.*name="\(.*\)"/\1/'
    fi
else
    echo "Greska: Fajlu ne moze da se pristupi"
    exit 2
fi
```

3. (10 poena)

```
#define N 3
#define key 123

void atomicOnTwoSems(int philId, int op) {
    int semId = semget(key, N, 0666 | IPC_CREAT );

    struct sembuf sems[2];
    sems[0].sem_num = philId;
    sems[1].sem_num = (philId+1)%N;
    sems[0].sem_op = sems[1].sem_op = op;
    sems[0].sem_flg = sems[1].sem_flg = SEM_UNDO;
    semop(semId, sems, (size_t)2);
}

void philosopher(int id) {

    while (1) {
        //request forks
        atomicOnTwoSems(id,-1);

        //eat
        sleep(rand()/RAND_MAX);

        //relese forks
        atomicOnTwoSems(id,1);

        //think
        sleep(rand()/RAND_MAX);
    }
}

int main() {

    int semId = semget(key, N, 0666 | IPC_CREAT );
    for (int var = 0; var < N; ++var) {
        semctl(semId, var, SETVAL, 1);
    }
    //philosophers
    int id;
    for (id = 0; id < N; id++) {
        if (fork() == 0) {
            philosopher(id);
        }
    }
    wait(0);
    return 0;
}
```