

Rešenja drugog kolokvijuma iz Operativnih sistema 2, septembar 2015.

1. (10 poena)

```
int ResourceAllocator::request (int p, ResourceVector req) {
    if (p<0 || p>=np) return -1; // Exception
    if (alloc[p]+req>max[p]) return -2; // Request beyond announcement
    if (req>free) return -3; // Not enough free resources
    // Try to allocate it and check if it leads to a safe state
    alloc[p]+=req;
    free-=req;
    if (!isSafe()) {
        alloc[p]-=req;
        free+=req;
        return -4; // Cannot be acquired because it leads to an unsafe state
    } else
        return 0; // Resources acquired
}
```

2. (10 poena)

```
unsigned getLRUPage (PCB* pcb) {
    if (pcb==0) return; // Exception!
    unsigned minPage = 0, minRef = 0, first = 1;
    for (unsigned page=0; page<PMTSIZE; page++) {
        unsigned frame = pcb->pmt[page];
        if (!frame) continue;
        unsigned ref = pcb->pageRefHash.getValue(page);
        if (first || ref<=minRef) {
            first = 0;
            minRef = ref;
            minPage = page;
        }
    }
    return minPage;
}
```

3. (10 poena)

```
void incPageFaultCounter (PCB* pcb) {
    if (pcb==0) return; // Exception!
    pcb->pageFaultCounters[pcb->pageFaultCursor]++;
}

void shiftPageFaultCounters (PCB* pcb) {
    if (pcb==0) return; // Exception!
    pcb->pageFaultCursor++;
    if (pcb->pageFaultCursor>=PFLTCOUNTERS)
        pcb->pageFaultCursor = 0;
    pcb->pageFaultCounters[pcb->pageFaultCursor] = 0;
}

unsigned getNumberOfPageFaults (PCB* pcb) {
    if (pcb==0) return; // Exception!
    unsigned sum = 0;
    for (int i=0; i<PFLTCOUNTERS; i++)
        sum += pcb->pageFaultCounters[i];
    return sum;
}
```