

Prvi kolokvijum iz Operativnih sistema 1

Avgust 2023.

1. (10 poena)

Osnovno rešenje bez optimizacije:

```
f:
    load r1,#1
    load r2,#8
    sub sp,sp,r2 ; struct S s
    load r2,[sp+4*4] ; r2 = n
    and r2,r2,r2 ; r2 == 0?
    jnz f_1001
    store r1,[sp+0] ; s.a = 1
    store r1,[sp+4] ; s.b = 1
    jmp f_1002
f_1001:
    push r0
    load r0,sp ; s = f(n-1)
    sub r2,r2,r1 ; r2 = n-1
    push r2
    call f
    pop r2
    pop r0
f_1002:
    load r2,[sp+0] ; return s
    store r2,[r0+0]
    load r2,[sp+4]
    store r2,[r0+4]
    load r2, #8
    add sp, sp, r2
    ret
```

Rešenje sa tzv. *Named Return Value* optimizacijom (NRVO, za objašnjenje videti <http://afrodita.rcub.bg.ac.rs/~dmilicev/publishing/OOP%20predavanja%202018>, slajdovi počev od 362, konkretno slajd 373):

```
f:
    load r1,#1
    load r2,[sp+2*4] ; r2 = n
    and r2,r2,r2 ; r2 == 0?
    jnz f_1001
    store r1,[r0+0] ; s.a = 1
    store r1,[r0+4] ; s.b = 1
    ret
f_1001:
    sub r1,r2,r1 ; r1 = n-1
    push r1
    call f
    pop r1
    ret
```

2. (10 poena)

```
inline void getMemCtxt (PCB* pcb, uint32& base, uint32& limit) {
    base = pcb->baseBlk * BLK_SIZE;
    limit = pcb->numOfBlks * BLK_SIZE - 1;
}
```

```

inline bool isMemBlkFree (size_t num) {
    return freeMemBlks[num/32] & (1<<(num%32));
}

inline void allocMemBlk (size_t num) {
    freeMemBlks[num/32] &= ~(uint32) (1<<(num%32));
}

int expand (PCB* pcb) {
    size_t newBlk = pcb->baseBlk + pcb->numOfBlocks;
    if (!isMemBlkFree(newBlk)) return -1;
    allocMemBlk(newBlk);
    pcb->numOfBlocks++;
    return 0;
}

```

3. (10 poena)

```

void handlePageFault (PCB* pcb, uint32 page) {
    SegDsc* sd = getSegDesc(pcb, page);
    if (!sd) {
        pcb->handleMemSegFault(pcb, page);
        return;
    }
    void* frame = allocFrame();
    PageDsc* pd = getPageDesc(pcb, page);
    if (isFirstAccess(pd)) {
        setAccessed(pd);
        sd->coldLoad(sd, pcb, pd, page, frame);
    } else
        sd->hotLoad(sd, pcb, pd, page, frame);
}

```