

Prvi kolokvijum iz Operativnih sistema 1

Avgust 2026.

1. (10 poena)

a)(5)

```
memcpy_usr2krnl:
    load    r4,#4
l1: load    r0,[r1]
    store   r0,[r3]
    add     r1,r1,r4
    add     r3,r3,r4
    dec     r2
    jnz     l1
    ret
```

b)(5)

```
void memcpy_usr2krnl (const uint32_t* usrbuf, size_t sz, uint32_t* krnlbuf) {
    while (sz-- > 0)
        *krnlbuf++ = v2p_addr(running->pmt,usrbuf++);
}
```

2. (10 poena)

```
const uint32_t offsetw = 10;
const uint32_t pagelw = 8;
const uint32_t page2w = 8;
const uint32_t PMT_size = 256;
const uint32_t SMT_size = 64;

typedef uint32_t PMT[PMT_SIZE];
typedef uint32_t SMT[SMT_SIZE][2];

void* v2pAddr (SMT smt, void* vaddr, uint8_t* rwx) {
    uint32_t va = (uint32_t)vaddr;
    static const uint32_t all1s = (uint32_t)-1;
    uint32_t seg = va >> (pagelw + page2w + offsetw);
    uint32_t page = (va >> offsetw) & ~(all1s<<(pagelw+page2w));
    uint32_t pagel = (va >> (page2w + offsetw)) & ~(all1s<<pagelw);
    uint32_t page2 = (va >> offsetw) & ~(all1s<<page2w);
    uint32_t offset = va & ~(all1s<<offsetw);
    if (rwx) *rwx = smt[seg][1] & 0b111;
    uint32_t* pmt = (uint32_t*)smt[seg][0];
    if (!pmt) return nullptr;
    uint32_t limit = smt[seg][1] >> 3;
    if (page>limit) return nullptr;
    pmt = pmt[pagel];
    if (!pmt) return nullptr;
    uint32_t frame = pmt[page2];
    if (!frame) return nullptr;
    uint32_t paddr = (frame<<offsetw) + offset;
    return (void*)paddr;
}
```

3. (10 poena)

```
Status MemContext::handlePF (void* va) {
    Page pg = this->getPage(va);
    MemRegion* mr = this->getMemRegion(pg);
    if (!mr) return ERR_INVALID_ADDR;
    void* fr = getFreeFrame();
    if (!fr) return ERR_NO_FREE_SPACE;
    PageDesc* pd = this->getPageDesc(pg);
    SwapBlk sb = pd->swapBlk;
    Status ret;
    if (sb)
        ret = mr->loadPageFromSwap(sb, fr);
    else
        ret = mr->loadInitPage(pg, fr);
    return ret;
}
```