

Rešenja zadataka za kolokvijum iz Operativnih sistema 1 Rok 1/2025.

1. (10 poena)

```
const short THR_STACK = 5;
const short THR_PRIV_DATA = 6;

void switchMemContext (Thread* toRun) {
    MPU::setRegion(THR_STACK, toRun->getStackStart(), toRun->getStackSize(),
                   O_URW|O_KRW);
    void* dataStart = toRun->getPrivDataStart();
    if (dataStart)
        MPU::setRegion(THR_PRIV_DATA, dataStart, toRun->getPrivDataSize(),
                       O_URW|O_KRW);
    else
        MPU::invalidateRegion(THR_PRIV_DATA);
}
```

2. (10 poena)

```
sys_call:    push    sp
             push    r0
             push    r1
             ...
             push    r31
             load    r0,running
             store    ssp,[r0+offsSP]

             load    ssp,kernelSP
             call    handle_sys_call

             load    r0,running
             load    ssp,[r0+offsSP]
             pop     r31
             ...
             pop     r1
             pop     r0
             pop     sp
             rti
```

3.

```
class BlockDevice {
public:
    BlockDevice (IVTNo ivte, BlkDeviceDriver* drv);

    int read  (BlkNo blkNo, void* buffer);
    int write (BlkNo blkNo, void* buffer);

protected:
    int perform (BlkNo blkNo, void* buffer, int rdwr);

private:
    friend void blkDevISR (void*);
    Semaphore queue, eop;
    BlkDeviceDriver* myDrv;
    static const int RD = 0, WR = 1;
};
```

```

inline int BlockDevice::read (BlkNo blkNo, void* buffer) {
    return this->perform(blkNo,buffer,RD);
}

inline int BlockDevice::write (BlkNo blkNo, void* buffer) {
    return this->perform(blkNo,buffer,WR);
}

interrupt void blkDevISR (void* dev) {
    BlockDevice* d = (BlockDevice*)dev;
    d->eop.signal();
}

BlockDevice::BlockDevice (IVTNo ivte, BlkDeviceDriver* drv)
: queue(1), eop(0), myDrv(drv) {
    Interrupts::initIVT(ivte, blkDevISR, this);
    myDrv->setIVTE(ivte);
}

int BlockDevice::perform (BlkNo blkNo, void* buffer, int rdwr) {
    this->queue.wait();
    myDrv->start(blkNo,buffer,rdwr);
    this->eop.wait();
    int status = myDrv->getStatus();
    this->queue.signal();
    return status;
}

```

4. (10 poena)

```

cat ../adoc > b/bdoc
ln b/bdoc ../hdoc
ln -s ../hdoc a/sdoc
rm ../hdoc
rm ../adoc

```

Komanda `cat a/sdoc` će ispisati sadržaj iskopiran u `bdoc` iz obrisanoj fajla `adoc`.