

Rešenja zadataka za kolokvijum iz Operativnih sistema 1 - Rok 2/2025.

1. (10 poena)

a)(5)

b)(3) 3 po 16 ulaza c)(2) $0x100 = 256$

Stranica # (hex)	RWX (bin)
0	001
1	001
2	001
4	100
5	001
6	001
7	001
8	001
17	110
18	110
EE	110
EF	110

2. (10 poena)

```
class PeriodicThread : public Thread {
public:
    PeriodicThread (Time period, Time start = 0)
        : myPeriod(period), myStart(start), toStop(false) {}

    void stop () { toStop = true; };
    virtual void activate () = 0;

protected:
    virtual void run ();

private:
    Time myPeriod, myStart;
    bool toStop;
};

void PeriodicThread::run () {
    if (myStart!=0) Timing::sleep(myStart);
    Time nextPeriod = Timing::now();
    while (!toStop) {
        nextPeriod += myPeriod;
        activate();
        Timing::sleep(nextPeriod);
    }
}
```

3. (10 poena)

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

    static const int RD = 0, WR = 1;
    int perform (BlkNo blkNo, void* buffer, int rdwr);
};
```

```

protected:
    virtual void run ();

private:
    struct Request {
        BlkNo blkNo; void* buffer; int rdwr; int status; Semaphore* sem;
    };

    BoundedBuffer<Request*,50> buffer;
    BlkDeviceDriver* myDrv;
    Semaphore eop; friend void blkDevISR (void*);
};

inline int BlockDevice::perform (BlkNo blkNo, void* buffer, int rdwr) {
    Semaphore eop = 0;
    Request req{blkNo,buffer,rdwr,0,&eop};
    this->buffer.append(&req);
    eop.wait();
    int status = req.status;
    return status;
}

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

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

void BlockDevice::run () {
    while (true) {
        Request* req = this->buffer.take();
        myDrv->start(req->blkNo,req->buffer,req->rdwr);
        this->eop.wait();
        req->status = myDrv->getStatus();
        req->sem->signal();
    }
}

```

4. (10 poena)

a)(5) $n+1$.

Objašnjenje: najpre se učitava prvi blok zapisa korenog direktorijuma sa unapred poznate fiksne lokacije. U njegovom prvom ulazu pronalazi se FCB za dir_1 i tu se nalazi broj prvog bloka zapisa njegovog sadržaja. Zatim se učitava taj blok i u njegovom prvom ulazu određuje broj prvog bloka zapisa sadržaja dir_2 itd. U n -tom učitanoj bloku pronalazi se broj prvog bloka zapisa sadržaja dir_n . U tom, $n+1$ -om učitanoj bloku, u prvom ulazu, pronalazi se broj prvog bloka sadržaja fajla.

b) $2(n+1)$, jer se za svaki element staze mora učitati dva bloka, pošto se on nalazi u drugom bloku sadržaja (jedan blok sadrži 16 ulaza), a pretraga ulaza je sekvencijalna po nizu ulaza, dok se pozicija drugog bloka sa sadržajem dobija iz FAT koji je keširan.