Prvi kolokvijum iz Operativnih sistema 1
Odsek za računarsku tehniku i informatiku
April 2013.

1. (10 poena)

```
const REG ESC = 0;

void main () {
  REG data = ESC;

  *io1Ctrl = 1;
  *io2Ctrl = 1;
  *io3Ctrl = 1;

  while (1) {
    while (*io1Status==0 && *io2Status==0);
    if (*io1Status) data = *io1Data;
    else data = *io2Data;
  if (data==ESC) break;
    while (*io3Status==0);
    *io3Data = data;
  }

  *io1Ctrl = 0;
  *io2Ctrl = 0;
  *io3Ctrl = 0;
}
```

2. (10 poena)

a)(5) VA: Page(16):Offset(14); PA: Frame(18):Offset(14).

b)(5) 64K 32-bitnih reči.

Pošto virtuelni prostor ima 64K = 2^{16} stranica, toliko ulaza ima i PMT.

Svaki ulaz je veličine najmanje jedne adresibilne jedinice (32-bitne reči), što je i dovoljno za smeštanje broja okvira i eventualnih dodatnih bita, pa PMT zauzima 64 K reči.

3. (10 poena)

```
sys_call:   ; Save current context
            push  r0
            load  r0,[running]
            store r1,#offsR1[r0] ; save r1
            pop   r1  ; save r0 through r1
            store r1,#offsR0[r0]
            store r2,#offsR2[r0] ; save other regs
            store r3,#offsR3[r0]
            ...
            store r31,#offsR31[r0]
            store sp,#offsSP[r0] ; save sp

            ; Switch to kernel code
            load  sp,[kernelStack] ; switch to kernel stack
            inte ; enable interrupts
            call  kernel ; go to kernel code
            intd ; disable interrupts

            ; Restore new context
            load  r0,[running]
            load  sp,#offsSP[r0] ; restore sp
            load  r31,#offsR31[r0] ; restore r31
            ...  ; restore other regs
            load  r1,#offsR1[r0]
            load  r0,#offsR0[r0] ; restore r0
            ; Return
            iret
```

4. (10 poena)

```
class Thread {
public:
  void start ();
  virtual ~Thread ();
protected:
  Thread () : myPID(0) {}
  virtual void run () {}
private:
  static void thread (void*);
  int myPID;
};

void Thread::thread (void* p) {
  Thread* thr = (Thread*)p;
  if(thr) thr->run();
}

void Thread::start () {
  if (myPID==0) myPID = thread_create(thread,this);
}

Thread::~Thread () {
  if (myPID>0) wait(myPID);
}
```