

# Prvi kolokvijum iz Operativnih sistema 1

## Odsek za softversko inženjerstvo

### Mart 2015.

#### 1. (10 poena)

```
static int dir = 0; // current transfer direction
static REG* ptr = 0; // pointer to current data item
static int count = 0; // counter

void startIO () { // Helper: start a new transfer
    ptr = ioHead->buffer;
    count = ioHead->size;
    dir = ioHead->dir;
    *ioCtrl = 1 | (dir<<1); // Start I/O
}

void transfer () {
    if (ioHead==0) return; // Exception - no requests
    startIO();
}

interrupt void ioInterrupt () {
    if (*ioStatus&2)
        ioHead->status = -1; // Error in I/O
    else { // Transfer the next data item
        if (dir)
            *ioData = *ptr++;
        else
            *ptr++ = *ioData;
        if (--count)
            return;
        else
            ioHead->status = 0; // Transfer completed successfully
    }

    ioHead = ioHead->next; // Remove the request from the list
    if (ioHead==0)
        *ioCtrl = 0; // No more requests
    else
        startIO(); // Start a new transfer
}
```

## 2. (10 poena)

```
dispatch:    ; Save the current context
             load  rx,running
             store r0,#offsR0[rx] ; save regs
             store r1,#offsR1[rx]
             ...
             store r31,#offsR31[rx]
             pop   r0 ; save pc
             store r0,#offsPC[rx]
             pop   r0 ; save psw
             store r0,#offsPSW[rx]
             pop   r0 ; save original sp
             store r0,#offsSP[rx]

             ; Select the next running process
             call  scheduler

             ; Restore the new context
             load  rx,running
             load  r0,#offsSP[rx] ; restore original sp
             push  r0
             load  r0,#offsPSW[rx] ; restore original psw
             push  r0
             load  r0,#offsPC[rx] ; restore pc
             push  r0
             load  r0,#offsR0[rx] ; restore regs
             load  r1,#offsR1[rx]
             ...
             load  r31,#offsR31[rx]
             ; Return
             iret
```

## 3. (10 poena)

a)(5) Problem je što su promenljive `flag` i `c` koje bi trebalo da budu deljene (zajedničke) između niti, jer preko njih treba da razmenjuju podatke i sinhronizuju se, definisane kao automatske (alociraju se na steku), što znači da zapravo neće biti deljene, već će svaka nit imati svoju instancu ovih promenljivih (na svom steku), pošto svaka nit ima svoj zaseban stek, pa razmene zapravo neće ni biti. Rešenje je prosto deklarirati ih kao statičke (`static`).

b)(5)

```
void pipe () {
    static char c1, c2;
    static int flag1 = 0, flag2 = 0;

    if (fork())
        writer(&c1,&flag1);
    else
        if (fork())
            reader(&c1,&flag1);
        else
            if (fork())
                writer(&c2,&flag2);
            else
                reader(&c2,&flag2);
}
```