

Prvi kolokvijum iz Operativnih sistema 1

Jun 2018.

1. (10 poena)

```
const REG ESC = 0;

void transfer () {
    REG data = ESC;

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

    while (1) {
        while (*io1Status==0);
        data = *io1Data;

        if (data==ESC) break;

        ind done = 0;
        while (!done) {
            if (*io2Status!=0)
                *io2Data = data, done = 1;
            else
                if (*io3Status!=0)
                    *io3Data = data, done = 1;
        }
    }

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

2. (10 poena) a)(7)

```
dispatch:    ; Save the current context
             push r0    ; save regs
             push r1
             ...
             push r31
             store ssp, #offsSSP[rpid] ; save ssp

             ; Select the next running process and store its PCB* in rpid
             call scheduler

             ; Restore the new context
             load ssp, #offsSSP[rpid] ; restore ssp
             pop r31
             pop r30 ; restore regs
             ...
             pop r0
             ; Return
             iret
```

b)(3) Ne treba. Ova prekidna rutina pristupa samo registrima procesora, strukturi PCB tekućeg procesa i njegovom sistemskom steku. Kako su sve tri stvari korišćene isključivo od strane tog procesora (jer je taj proces raspoređen samo tom procesoru), nema potrebe za međusobnim isključenjem.

3. (10 poena)

```
const int N = ...;
int mat[N][N];
int sums[N];
pthread_t pid[N];
typedef int Row[N];

void sum (Row* row) {
    int s = 0;
    for (int i=0; i<N; i++)
        s += (*row)[i];
    sums[row-&mat[0]] = s;
}

// Wrapper, for type-casting only:
void* _sum (void* row) {
    sum((Row*)row);
    return NULL;
}

int par_sum () {
    for (int i=0; i<N; i++)
        pthread_create(&pid[i], &_sum, &a[i]);
    int s = 0;
    for (int i=0; i<N; i++) {
        pthread_join(pid[i], NULL);
        s += sums[i];
    }
    return s;
}
```