

Prvi kolokvijum iz Operativnih sistema 1

Odsek za softversko inženjerstvo

Mart 2012.

1. (10 poena)

```
static unsigned* io2Ptr = 0;
static int io2Count = 0;
static int io2Completed = 0;

void transfer (unsigned* blk1, int count1, unsigned* blk2, int count2) {
    // I/O 2:
    io2Ptr = blk2;
    io2Count = count2;
    io2Completed = 0;
    *io2Ctrl = 1; // Start I/O 2

    // I/O 1
    *io1Ctrl = 1; // Start I/O 1
    while (count1>0) {
        while (!(*io1Status&1)); // busy wait
        *io1Data = *blk1++;
        count1--;
    }
    *io1Ctrl = 0; // Stop I/O 1

    // Wait for I/O 2 completion:
    while (!io2Completed);
}

interrupt void io2Interrupt() {
    *io2Data = *io2Ptr++;
    if (--io2Count == 0) {
        io2Completed = 1;
        *io2Ctrl = 0; // Stop I/O 2
    }
}
```

2. (10 poena)

a)(5) VA: Page(24):Offset(8); PA: Frame(20):Offset(8). Nalazi se na adresi 109BCh.

b)(5)

```
void setPageDescr(unsigned* pmtpr, unsigned page, unsigned frame){
    pmtpr[page] = frame | ~((unsigned int)~0 / 2);
}
```

3. (10 poena)

a)(5)

```
f:   load  r0,[n] ; if (n==0)
      cmp   r0,#0
      jne  else
      ret   ; r0==0, return 0
else: dec  r0    ; f(n-1)
      push(n)
      store [n],r0
      call  f
      pop  (n)
      inc  r0    ; return f(n-1)+1
      ret
```

b)(5) Problem je to što je svakoj lokalnoj promenljivoj i argumentu pridružen jedan i samo jedan globalni i statički alocirani stek. Zbog toga taj stek može da „prati“ samo instance lokalnih promenljivih samo jedne niti, a ne više njih. Na primer, jedna nit bi mogla da pozove funkciju f sa datim argumentom n i druga učini to isto i uporedo, pokvarivši i tekuću vrednost i stek starih vrednosti za n prve niti. Za potrebe uporednih niti neophodno je imati zaseban skup instanci lokalnih promenljivih i argumenata pridružen svakoj niti. Prema tome, ceo skup statički alociranih lokalnih promenljivih i argumenata, zajedno sa njima pridruženim LIFO strukturama (pojedinačnim stekovima), mora da bude deo konteksta niti, što znači da se mora čuvati i restaurirati iz PCB prilikom promene konteksta niti, na sličan način kako se čuvaju i restauriraju registri procesora, odnosno analogno odvajanju zasebnog kontrolnog steka za svaku nit.

4. (10 poena)

```
void visit (void* nd) {
    TreeNode* node = (TreeNode*)nd;
    if (node==0) return;
    TreeNode* rn = getRightChild(node);
    if (rn) create_thread(&visit,rn);
    process(node);
    visit(getLeftChild(node));
}

void main () {
    TreeNode* root = ...;
    create_thread(&visit,root);
}
```