Prvi kolokvijum iz Operativnih sistema 1
Odsek za softversko inženjerstvo
Mart 2018.

1. (10 poena)

```
static unsigned *io1Ptr = 0, *io2Ptr = 0;
static int io1Count = 0, io2Count = 0;
static const unsigned timeout = 50;

void transfer (unsigned* blk1, int count1, unsigned* blk2, int count2) {
  // I/O 1:
  io1Ptr = blk1;
  io1Count = count1;
  *io1Ctrl = 1; // Start I/O 1
  *timer = timeout; // Start timer

  // I/O 2:
  io2Ptr = blk2;
  *io2Ctrl = 1; // Start I/O 2
  for (io2Count=count2; io2Count>0; io2Count--) {
    while (*io2Stat&1 == 0); // Busy-wait for I/O 2 to be ready
    *io2Ptr++ = *io2Data;
  }
  *io2Ctrl = 0; // Stop I/O 2

  // Busy wait for I/O 1 completion:
  while (io1Count);
}

interrupt void timerInterrupt () {
  *io1Ptr++ = *io1Data;
  if (--io1Count)
    *timer = timeout; // Restart timer
  else
    *io1Ctrl = 0; // Stop I/O 1
}
```

2. (10 poena)

```
      dispatch:   ; Save the current context
                  push  r0    ; save regs
                  push  r1
                  ...
                  push  r31
                  load  r0, running
                  store sp, #savedSP[r0] ; save sp

                  ; Select the next running process
                  call  scheduler

                  ; Restore the new context
                  load  r0, running
                  load  r1, #timeSlice[r0]; restart timer
                  store r1, [Timer]
                  load  sp, #savedSP[r0] ; restore sp
                  pop   r31
                  pop   r30 ; restore regs
                  ...
                  pop   r0

                  ; Return
                  iret
void scheduler () {
  do {
    running = processes + (running - processes + 1) % NUM_OF_PROCESSES;
  } while (running->status!=ready);
}
```

3. (10 poena)

```
#include <stdio.h>

char command[33];

void main () {

  while (true) {

    scanf(“%32s“,&command);
  if (strcmp(command,"q")==0) break;

    int pid = fork();
    if (pid<0)
      printf(“Error executing %s.\n“,command);
    else
    if (pid>0)
      wait(pid);
    else
      execlp(command);
  }
}
```