Prvi kolokvijum iz Operativnih sistema 1
Odsek za računarsku tehniku i informatiku
Maj 2012.

1. (10 poena)

```
static int dmaCompleted = 0;

void transfer (OutputRequest* ioHead) {
  while (ioHead) {
    dmaCompleted = 0;  // initialize transfer
    *dmaBlkAddress = ioHead->buffer;
    *dmaBlkSize = ioHead->size;
    *dmaCtrl = 1;  // start transfer
    while (!dmaCompleted); // wait for DMA to complete
    ioHead->callBack(ioHead);  // signal completion
    ioHead = ioHead->next;  // take next
  }
}

interrupt void dmaInterrupt () {
  dmaCompleted = 1;
}
```

2. (10 poena)

a)(5) VA: Segment(8):Page(16):Offset(8); PA: Frame(20):Offset(8).

b)(5) FF0015h

3. (10 poena)

```
void yield (jmp_buf old, jmp_buf new) {
  if (setjmp(old)==0)
    longjmp(new,1);
}

void dispatch () {
  lock();
  jmp_buf old = Thread::running->context;
  Scheduler::put(Thread::running);
  Thread::running = Scheduler::get();
  jmp_buf new = Thread::running->context;
  yield(old,new);
  unlock();
}
```

4. (10 poena)

```
// Helper recursive function: traverse the tree, compute its size,
// and store its size in sz (sz is external to the thread’s stack):

void size_ (Node* node, int& sz) {
  Node* ln = node->getLeftChild();
  int lsz = 0; // left subtree size
  Node* rn = node->getRightChild();
  int rsz = 0; // right subtree size

  if (rn) {
    if (fork()==0) {
      size_(rn,&rsz);
      exit();
    }
  }

  if (ln) size_(ln,&lsz);

  wait(null); // wait for all descendants to complete
  *sz = lsz+rsz+1;  // compute the cumulative size and return to the caller
}


int size (Node* node) {
  if (node==0) return 0;
  int sz = 0;
  size_(node,&sz);
  return sz;
}
```