Prvi kolokvijum iz Operativnih sistema 1
Jun 2017.

1. (10 poena)

```
interrupt void intNet () {
  if (bufTail==bufHead) {
    // Buffer full, reject the packet:
    *ioNetCtrl = PKT_REJECT;
    return;
  }
  *dmaAddr = &buffer[bufTail];
  *dmaCount = PKT_SIZE;
  *dmaCtrl = DMA_START;
}


interrupt void intDMA () {
  ++bufTail %= BUF_SIZE;
}
```

2. (10 poena) a)(7)

U klasu `Thread` dodati su sledeći privatni, nestatički podaci-članovi sa datim inicijalnim vrednostima:

```
Thread* Thread::parent = 0;
bool Thread::isActive = false;
bool Thread::isWaitingForAllChildren = false;
unsigned long Thread::activeChildrenCounter = 0;
Thread* Thread::isWaitingForChild = 0;


void Thread::created (Thread* par) {
  this->isActive = true;
  this->parent = par;
  if (par) this->parent->activeChildrenCounter++;
}

void Thread::completed () {
  this->isActive = false;
  if (!this->parent) return;
  this->parent->activeChildrenCounter--;
  if ((this->parent->isWaitingForAllChildren &&
       this->parent->activeChildrenCounter==0) ||
      this->parent->isWaitingForChild==this) {
    this->parent->isWaitingForAllChildren = false;
    this->parent->isWaitingForChild = 0;
    Scheduler::put(this->parent);
  }
}
```

```
void Thread::wait (Thread* forChild=0) {
  lock();
  jmp_buf old = Thread::running->context;

  if (forChild==0)
    if (Thread::running->activeChildrenCounter>0)
      Thread::running->isWaitingForAllChildren = true;
    else
      Scheduler::put(Thread::running);
  else
    if (forChild->parent==Thread::running && forChild->isActive)
      Thread::running->isWaitingForChild = forChild;
    else
      Scheduler::put(Thread::running);

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

3. (10 poena)

a)

```
class ThreadFnCaller : public Thread {
public:
  ThreadFnCaller (void (*fn)(void*), void* arg) : myFn(fn), myArg(arg) {}
  virtual void run () { myFn(myArg); }
private:
  void (*myFn)(void*);
  void* myArg;
};
```

b)

```
for (int i=0; i<N; i++) (new ThreadFnCaller(fn,args[i]))->start();
```