

Rešenja zadataka za treći kolokvijum iz Operativnih sistema 1 Jun 2016.

1. (10 poena)

```
typedef unsigned short Byte;
const int BlockSize = ...;
const int NumOfBlocks = ...;
const int BufferSize = NumOfBlocks*BlockSize;

class Buffer {
public:
    Buffer ();
    void put (Byte b);
    void read (Byte block[]);
private:
    Byte buffer[BufferSize];
    int rdCursor, wrCursor;
    Semaphore mutex, spaceAvailable, itemAvailable;
};

Buffer::Buffer () : rdCursor(0), wrCursor(0),
    mutex(1), spaceAvailable(BufferSize), itemAvailable(0) {}

void Buffer::put (Byte b) {
    spaceAvailable.wait();
    mutex.wait();
    this->buffer[this->wrCursor] = b;
    this->wrCursor = (this->wrCursor+1)%BufferSize;
    bool toSignal = (this->wrCursor%BlockSize == 0);
    mutex.signal();
    if (toSignal) itemAvailable.signal();
}

void Buffer::read (Byte block[]) {
    itemAvailable.wait();
    mutex.wait();
    for (int i=0; i<BlockSize; i++)
        block[i] = this->buffer[this->rdCursor+i];
    this->rdCursor = (this->rdCursor+BlockSize)%BufferSize;
    mutex.signal();
    spaceAvailable.signal(BlockSize);
}
```

2. (10 poena)

```
Node* Node::getNode (PCB* pcb, const char* path) {
    static const char delimiter = '/';
    if (pcb==0 || path==0) return 0; // Exception!
    Node* node = 0;
    const char* pStart = path;
    if (*pStart==delimiter) {
        node = rootNode;
        pStart++;
    } else {
        node = pcb->curDir;
    }
    while (node && *pStart) {
        const char* pEnd = pStart+1;
        while (*pEnd && *pEnd!=delimiter) pEnd++;
        node = node->getSubnode(pStart,pEnd);
        pStart = (*pEnd)?(pEnd+1):pEnd;
    };
    return node;
}
```

3. (10 poena)

```
unsigned long append (FCB* fcb) {
    if (fcb==0 || freeHead==0) return 0;
    unsigned long last=0, next=fcb->head, ret=freeHead;
    while (next) last=next, next=fat[next];
    if (last)
        fat[last] = freeHead;
    else
        fcb->head = freeHead;
    unsigned long oldHead = freeHead;
    freeHead = fat[freeHead];
    fat[oldHead] = 0;
    return ret;
}
```