

Rešenja zadatka za treći kolokvijum iz Operativnih sistema 1 Jun 2023.

1. (10 poena)

```
class DiskDeviceDriver : public IBlockDeviceDriver {
public:
    virtual int  init (Semaphore* complete);
    virtual void startTransfer (DiskRequest*);
private:
    uint32* dmaCtrlRegion;
    Semaphore* semComplete;
    friend void transferComplete (void*);
};

int DiskDeviceDriver::init (Semaphore* sc) {
    semComplete = sc;
    dmaCtrlRegion = requestDMAChannel();
    if (!dmaCtrlRegion) return -1;
    uint32 ivte = requestIVTEntry(transferComplete,this);
    if (ivte<0) return -1;
    *(dmaCtrlRegion+0) = ivte;
    return 0;
}

void DiskDeviceDriver::startTransfer (DiskRequest* req) {
    *(dmaCtrlRegion+1) = req->buffer;
    *(dmaCtrlRegion+1) = req->startBlockNo;
    *(dmaCtrlRegion+1) = req->blockCount;
    *(dmaCtrlRegion+1) = req->dir;
}

interrupt void transferComplete (void* obj) {
    ((DiskDeviceDriver*)obj)->semComplete->signal();
}
```

2. (10 poena)

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>

#define handle_error(msg) \
    do { fprintf(stderr,"Error: %s\n",msg); exit(-1); } while(0)

int main (int argc, const char* argv[]) {
    if (argc!=2) handle_error("Missing argument.");

    int pipefd[2];
    if (pipe(pipefd)<0) handle_error("Cannot create a pipe.");

    pid_t pid = fork();
    if (pid<0) handle_error("Cannot create a child process.");

    if (pid>0) {

        write(pipefd[1],argv[1],strlen(argv[1]));

    } else {

        char c;
        while (read(pipefd[0],&c,1)>0)
            putchar(c);

    }

    close(pipefd[0]);
    close(pipefd[1]);
    exit(0);
}
```

3. (10 poena)

```
void truncate (FCB* fcb) {
    if (fcb==0) return;
    if (!fcb->head) return; // File already empty

    uint32 last = fcb->head, cnt = 1;
    while (fat[last])
        last = fat[last], cnt++;
    fat[last] = freeHead;
    freeHead = fcb->head;
    freeCount += cnt;
    fcb->head = 0;
    fcb->size = 0;
}
```