

# Rešenja zadataka za treći kolokvijum iz Operativnih sistema 1 Jun 2026.

## 1. (10 poena)

```
class KeyboardBuffer {
public:
    KeyboardBuffer() : head(0), tail(0), count(0), itemAvailable(0) {}
    char getc ();
    void putc (char c);

    static KeyboardBuffer* getCurrent () { return cur; }
    static void setCurrent (KeyboardBuffer* kb) { cur = kb; }

private:
    static const size_t KB_SIZE = 256;
    char buffer[KB_SIZE];
    size_t head, tail, count;
    Semaphore itemAvailable;

    static KeyboardBuffer* cur;
};

static KeyboardBuffer::KeyboardBuffer* cur = nullptr;

char KeyboardBuffer::getc () {
    itemAvailable.wait();
    kbint_mask();
    char c = buffer[head];
    head = (head+1)%KB_SIZE;
    count--;
    kbint_unmask();
    return c;
}

void KeyboardBuffer::putc (char c) {
    if (count<KB_SIZE) {
        buffer[tail] = c;
        tail = (tail+1)%KB_SIZE;
        count++;
        itemAvailable.signal();
    }
}

interrupt void keyb_int () {
    KeyboardBuffer* kb = KeyboardBuffer::getCurrent();
    while (*ioStatus & 1) {
        char c = *ioData;
        if (kb) kb->putc(c);
    }
}
```

2. **(10 poena)** Rešenje je dato sa obradom grešaka (koja nije zahtevana u zadatku). Funkcija `fgetc` vraća EOF ako je učitavanje stiglo do kraja fajla ili je došlo do greške; ukoliko se stiglo do kraja fajla, funkcija `feof` vraća 1, a ukoliko je došlo do greške, ona vraća 0.

```
#include <stdio.h>
```

```
#define handle_error(msg) \
    do { perror(msg); exit(-1); } while (0)
```

```
int main (int argc, const char* argv[]) {
    if (argc<3) handle_error("Syntax: redirect program_name input_file");

    FILE* in = fopen(argv[2],"r");
    if (!in) handle_error("Cannot open input file.");

    FILE* out = popen(argv[1],"w");
    if (!out) { fclose(in); handle_error("Cannot execute the program."); }

    for (char c=fgetc(in)); c!=EOF; c=fgetc(in))
        fputc(c,out);

    if (!feof(in)) {
        fclose(in) pclose(out); handle_error("Error reading file.");
    }

    fclose(in);
    pclose(out);
    return 0;
}
```

3. **(10 poena)**

```
BlockNo Dir::getFCBBlock (const char name[FNAME_LEN]) const {
    size_t entry = this->hash(name);
    size_t slot = this->hashTbl[entry];
    DirEntry* dentry = this->getDentry(slot);
    while (dentry) {
        if (dentry->cmpName(name))
            return dentry->getFCBBlockNo();
        slot = dentry->getNext();
        dentry = this->getDentry(slot);
    }
    return 0;
}
```