Rešenja trećeg kolokvijuma iz Operativnih sistema 2, januar 2026.

1. (10 poena)

```
void wait(HANDLE mutex) {
        ++waiting_;
        SignalObjectAndWait(mutex, cond_, INFINITE, FALSE);
        WaitForSingleObject(mutex, INFINITE);
        --waiting_;
}
```

2. (10 poena)

(1*): `exit 1`

(2*): `shift`

(3*): `! -d "$dir"`

(4*): `tr -s ' ' | cut -d ' ' -f 3`

(5*): `-r`

3. (10 poena)

```
#define READER 1
#define WRITER_REQ 2
#define WRITER_DONE 3
#define KEY 1234L

enum {
    REQ_R = 1,     // reader request
    REQ_W = 2,     // writer request
    DONE_R = 3,    // reader finished
    DONE_W = 4,    // writer finished
    GRANT = 5      // permission granted
};

struct msg {
    long mtype;    // 1 -> from READER, 2 -> from WRITER start, 3 -> from
WRITER end
    int kind;
    pid_t pid; // process id of the sender, must be greater than 3
};

int main(int argc, char **argv) {
    key_t key = KEY;
    int qid = msgget(key, IPC_CREAT | 0666);
```

```c
    int active_readers = 0;
    int writer_active = 0;

    while (1) {
        if (active_readers > 0) {
            struct msg m;
            msgrcv(qid, &m, sizeof(m) - sizeof(long), READER, 0);
            if (m.kind == DONE_R) {
                active_readers--;
            } else {
                active_readers++;
                struct msg g = {m.pid, GRANT, 0};
                msgsnd(qid, &g, sizeof(g) - sizeof(long), 0);
                continue;
            }
        }

        if (writer_active > 0) {
            struct msg m;
            msgrcv(qid, &m, sizeof(m) - sizeof(long), WRITER_DONE, 0);
            writer_active = 0;
            continue;
        }

        struct msg m;
        msgrcv(qid, &m, sizeof(m) - sizeof(long), 0, 0);
        if (m.kind == REQ_R) {
            active_readers++;
            struct msg g = {m.pid, GRANT, 0};
            msgsnd(qid, &g, sizeof(g) - sizeof(long), 0);
        } else if (m.kind == REQ_W) {
            writer_active = 1;
            struct msg g = {m.pid, GRANT, 0};
            msgsnd(qid, &g, sizeof(g) - sizeof(long), 0);
        }
    }

    msgctl(qid, IPC_RMID, NULL);
    return 0;
}
```