Rešenja trećeg kolokvijuma iz Operativnih sistema 2, februar 2026.

1. (10 poena)

```
static uint64_t getPhysicalBlock(uint64_t logBlkNum) {
    return logBlkNum / DISK_NUM;
}

static uint64_t getDiskId(uint64_t logBlkNum) {
    return logBlkNum % DISK_NUM;
}

void seqRead(uint64_t blkNum, uint64_t size, void* output_buffer)
{
    Semaphore* done[DISK_NUM];
    bool used[DISK_NUM] = { false };

    for (int i = 0; i < DISK_NUM; i++) {
        done[i] = new Semaphore(0);
    }

    uint8_t* out = (uint8_t*)output_buffer;
    for (uint64_t i = 0; i < size; i++) {
        uint64_t pos = i % DISK_NUM;
        if (used[pos]) {
            done[pos]->wait();
        }
        used[pos] = true;
        uint64_t logBlkNum = blkNum + i;
        uint64_t diskId = getDiskId(logBlkNum);
        uint64_t phyBlkNum = getPhysicalBlock(logBlkNum);
        void* dst = (void*)(out + i * BLOCK_SIZE);
        read(diskId, phyBlkNum, done[pos], dst);
    }

    for (int i = 0; i < DISK_NUM; i++) {
        done[i]->wait();
        delete done[i];
    }
}
```

2. (10 poena)

```
#!/bin/bash

if [[ $# -ne 1 ]]; then
    echo "Usage: $0 <username>"
    exit 1
fi

username="$1"
```

```
sed -i "s|^${username}:[^:]*:|${username}:*:|" /etc/passwd
```

3. (10 poena)

```
#define N       5
#define ROUNDS  1000
#define KEY 5555

int semid;

union semun {
    int              val;
    struct semid_ds *buf;
    unsigned short  *array;
};

static void sem_wait_op(int sem_num)
{
    struct sembuf op = { (unsigned short)sem_num, -1, 0 };
    semop(semid, &op, 1);
}

static void sem_signal_op(int sem_num)
{
    struct sembuf op = { (unsigned short)sem_num, 1, 0 };
    semop(semid, &op, 1);
}

static void philosopher(int id)
{
    srand((unsigned)(time(NULL) ^ ((unsigned long)getpid() << 16)));

    int left  = id;
    int right = (id + 1) % N;

    for (int round = 0; round < ROUNDS; round++) {
        think(id);

        sem_wait_op(left);
        sem_wait_op(right);

        eat(id);

        sem_signal_op(left);
        sem_signal_op(right);
    }

    exit(0);
}

int main(void)
{
    union semun arg;
```

```
    unsigned short init_vals[N];
    pid_t pids[N];

    semid = semget(KEY, N + 1, IPC_CREAT | 0666);

    for (int i = 0; i < N; i++)
        init_vals[i] = 1;

    arg.array = init_vals;
    semctl(semid, 0, SETALL, arg);

    for (int i = 0; i < N; i++) {
        pids[i] = fork();
        if (pids[i] == 0) {
            philosopher(i);
        }
    }

    for (int i = 0; i < N; i++)
        waitpid(pids[i], NULL, 0);

    semctl(semid, 0, IPC_RMID, 0);

    return 0;
}
```