

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

Студент: Барханоева Раяна Магометовна

Ст.билет: 1032252468

Группа: НКАбд-01-25

МОСКВА

2025 г

## Содержание

1. Цель работы.....	2
2. Выполнение работы.....	3
3. Самостоятельная работа.....	14
Вывод.....	17

## **1. Цель работы**

Приобретение практических навыков работы в Midnight Commander.

Освоение инструкций языка ассемблера `mov` и `int`.

## 2. Выполнение работы

Открыла Midnight Commander (рисунок 1).

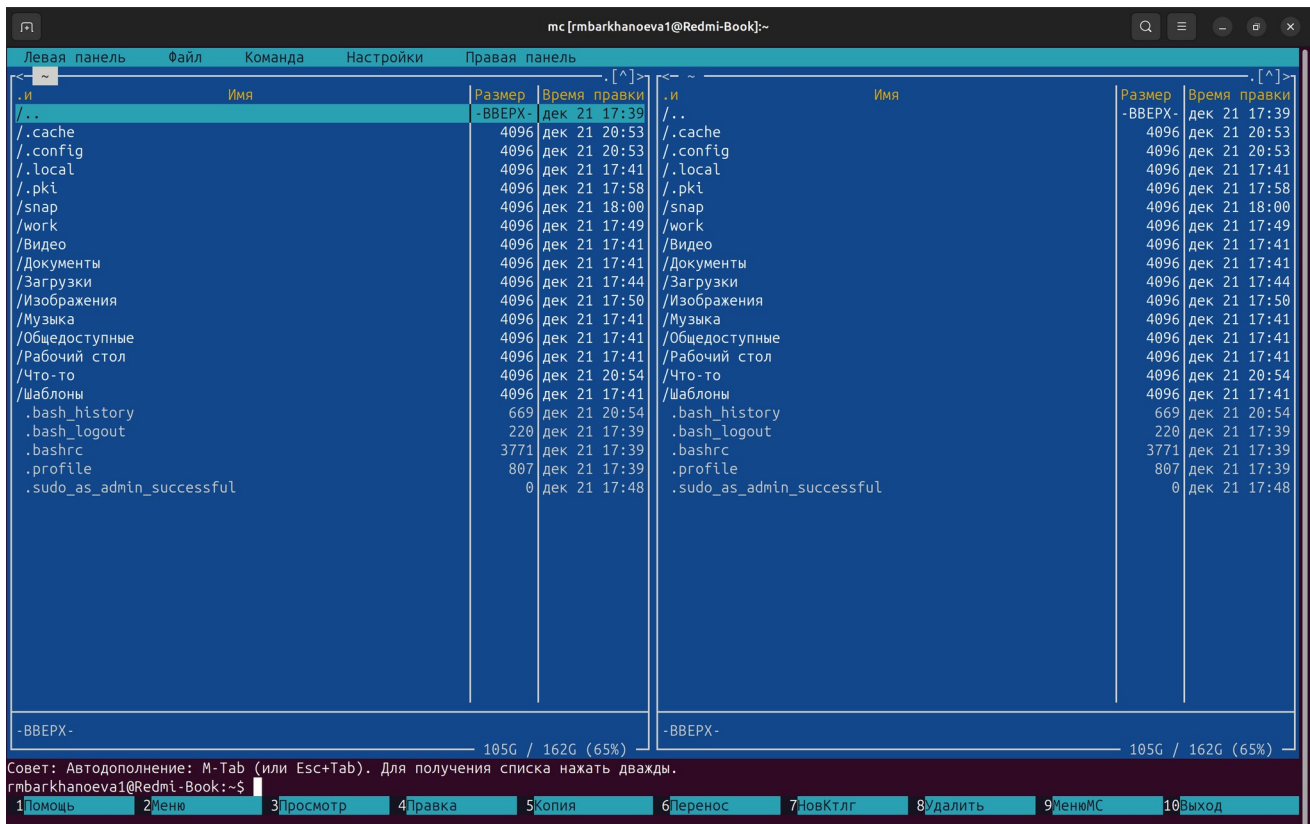


Рисунок 1.

С помощью команды `mc` открыла оболочку Midnight Commander.

Midnight Commander (или просто `mc`) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. `mc` является файловым менеджером.

Переход в каталог arch-rc (рисунок 2).

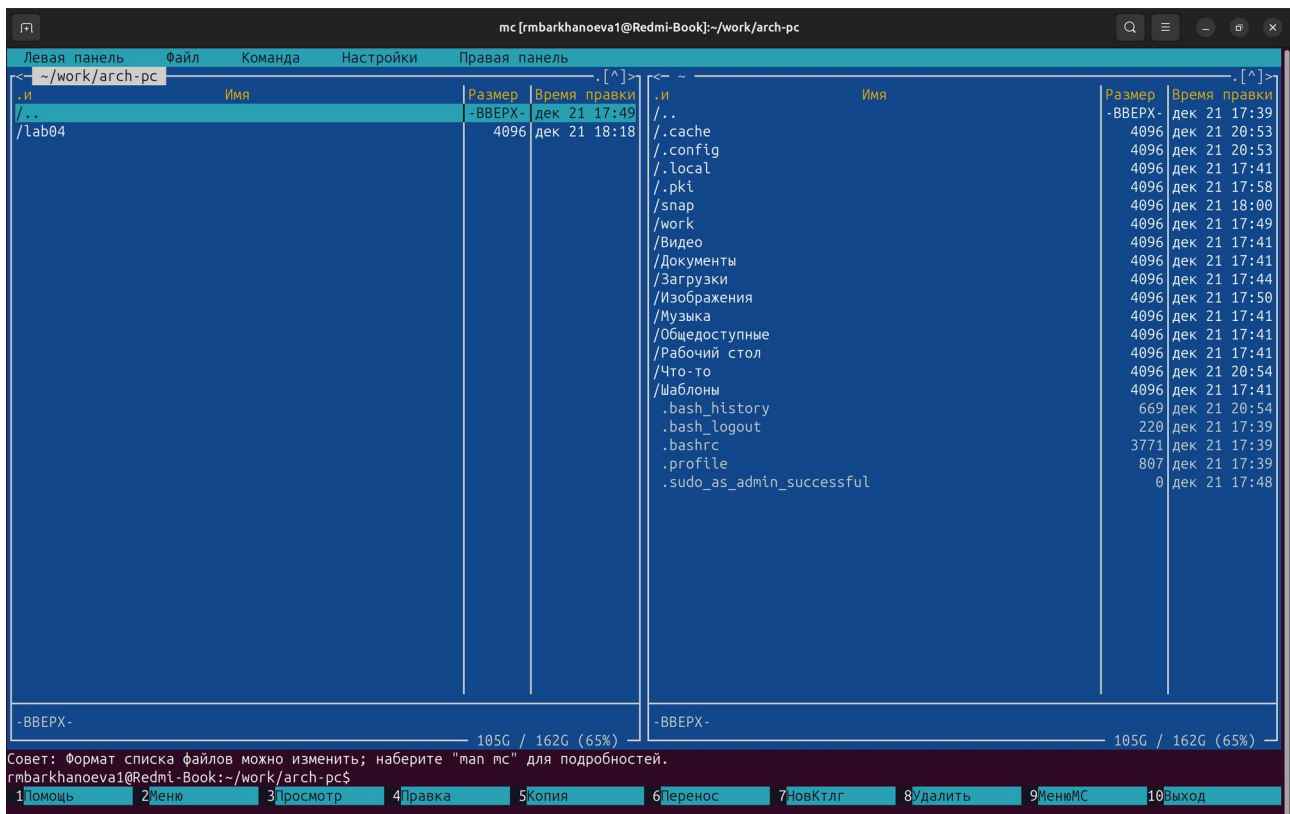


Рисунок 2.

Пользуясь клавишами  $\uparrow$ ,  $\downarrow$  и Enter перешла в каталог arch-rc.

Создание папки (рисунок 3).

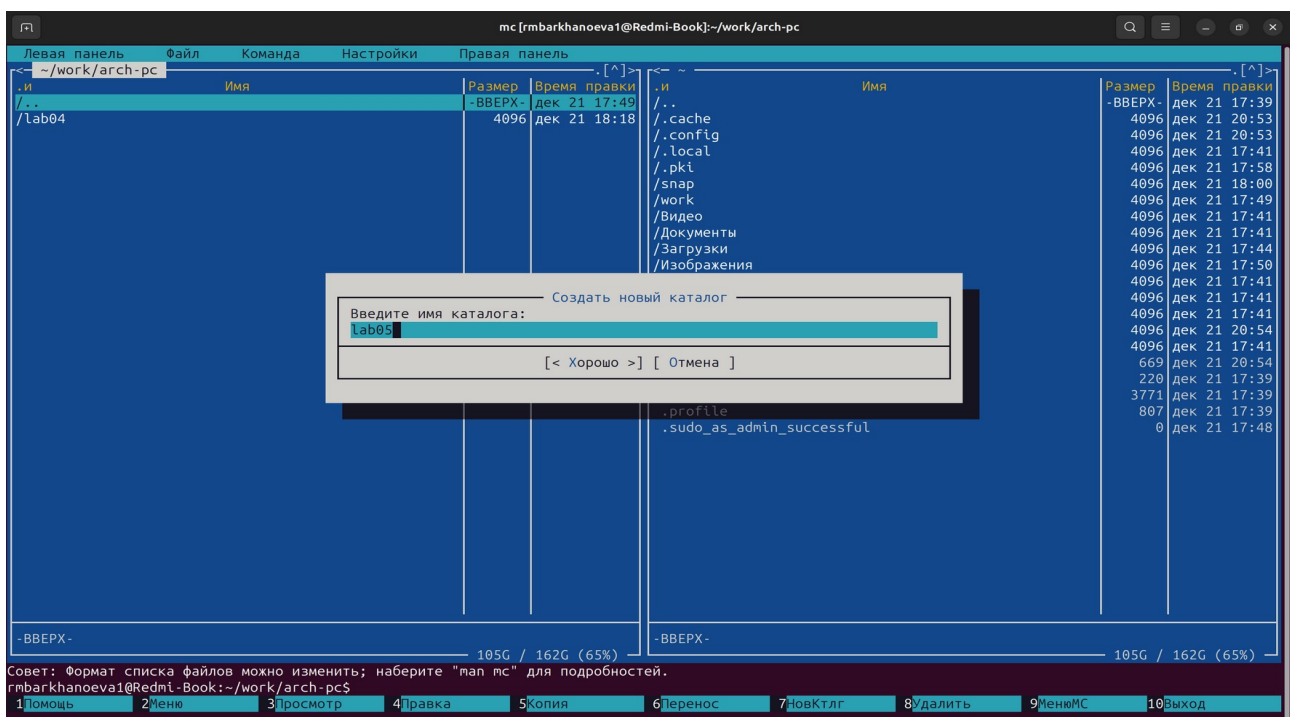


Рисунок 3.

С помощью функциональной клавиши F7 создала каталог lab05.

Создание файла (рисунок 4).

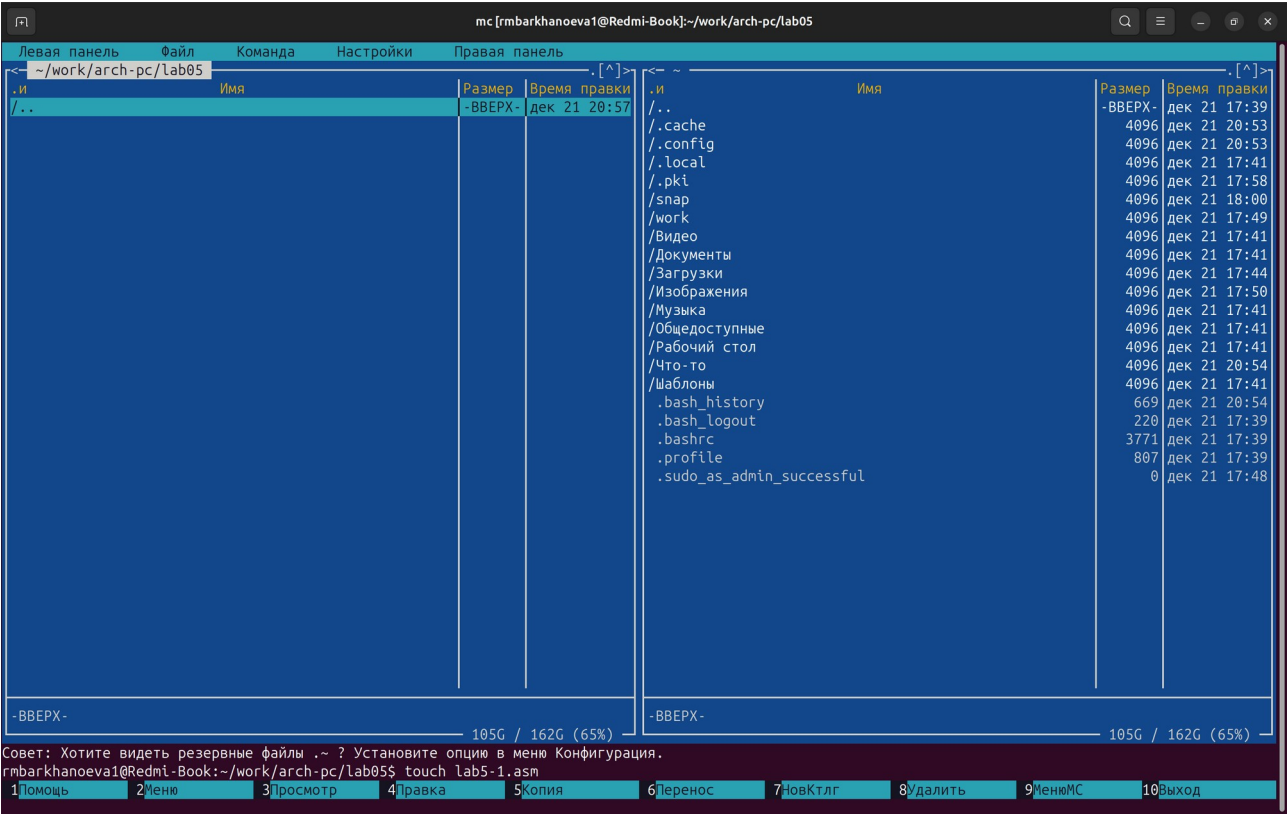
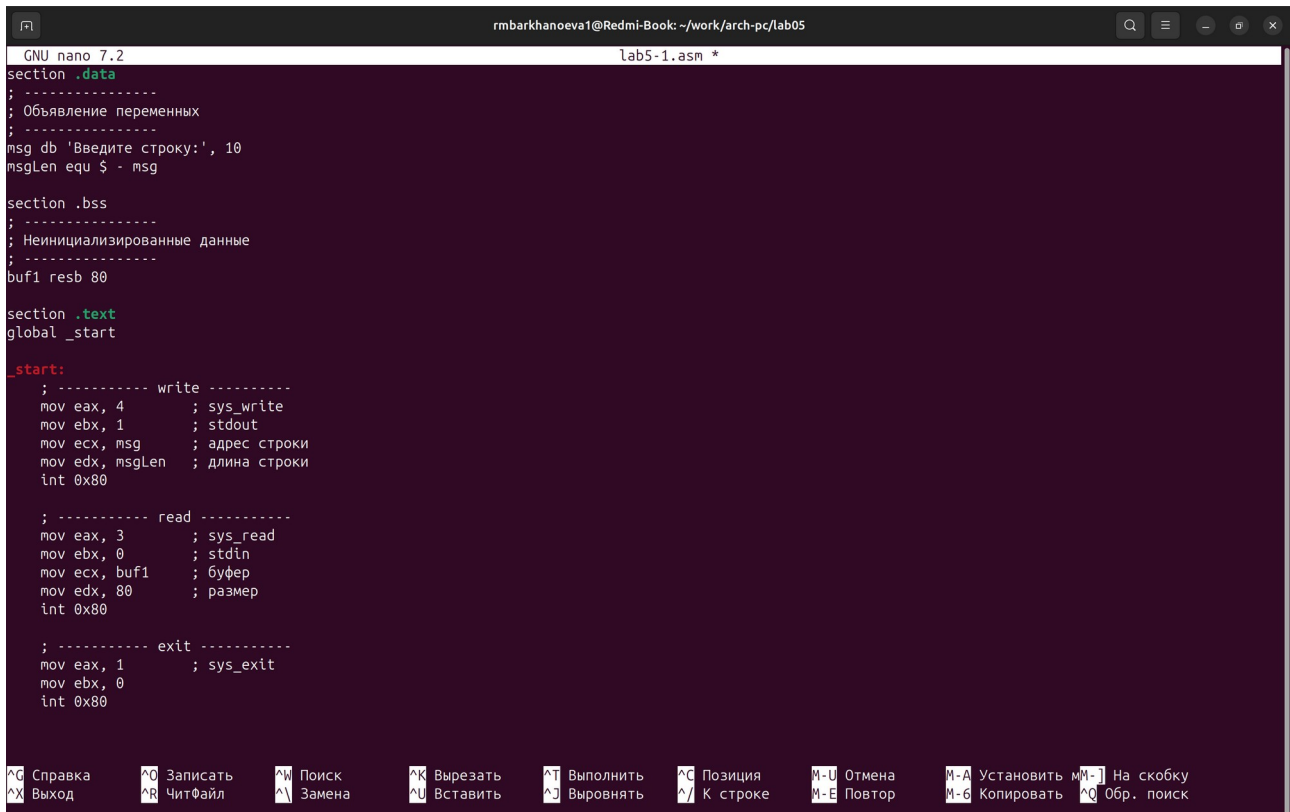


Рисунок 4.

Пользуясь командой touch создала файл lab5-1.asm.

Код программы (рисунок 5).



```
GNU nano 7.2 lab5-1.asm *
section .data
; -----
; Объявление переменных
; -----
msg db 'Введите строку:', 10
msgLen equ $ - msg

section .bss
; -----
; Неинициализированные данные
; -----
buf1 resb 80

section .text
global _start

_start:
; ----- write -----
mov eax, 4 ; sys_write
mov ebx, 1 ; stdout
mov ecx, msg ; адрес строки
mov edx, msgLen ; длина строки
int 0x80

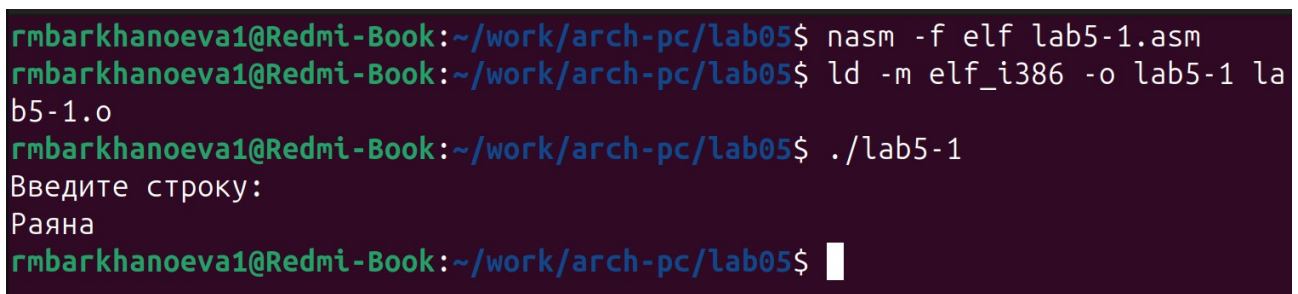
; ----- read -----
mov eax, 3 ; sys_read
mov ebx, 0 ; stdin
mov ecx, buf1 ; буфер
mov edx, 80 ; размер
int 0x80

; ----- exit -----
mov eax, 1 ; sys_exit
mov ebx, 0
int 0x80
```

Рисунок 5.

С помощью функциональной клавиши F4 открыла файл lab5-1.asm для редактирования во встроенном редакторе. И ввела текст программы из листинга 5.1

Проверка файла (рисунок 6).



```
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 la
b5-1.o
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Раяна
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$
```

Рисунок 6.

Сделала файл объектным и также выполнила его компоновку. Код запустился.

## 2.1. Подключение внешнего файла in\_out.asm

Скачивание in\_out.asm (рисунок 7).

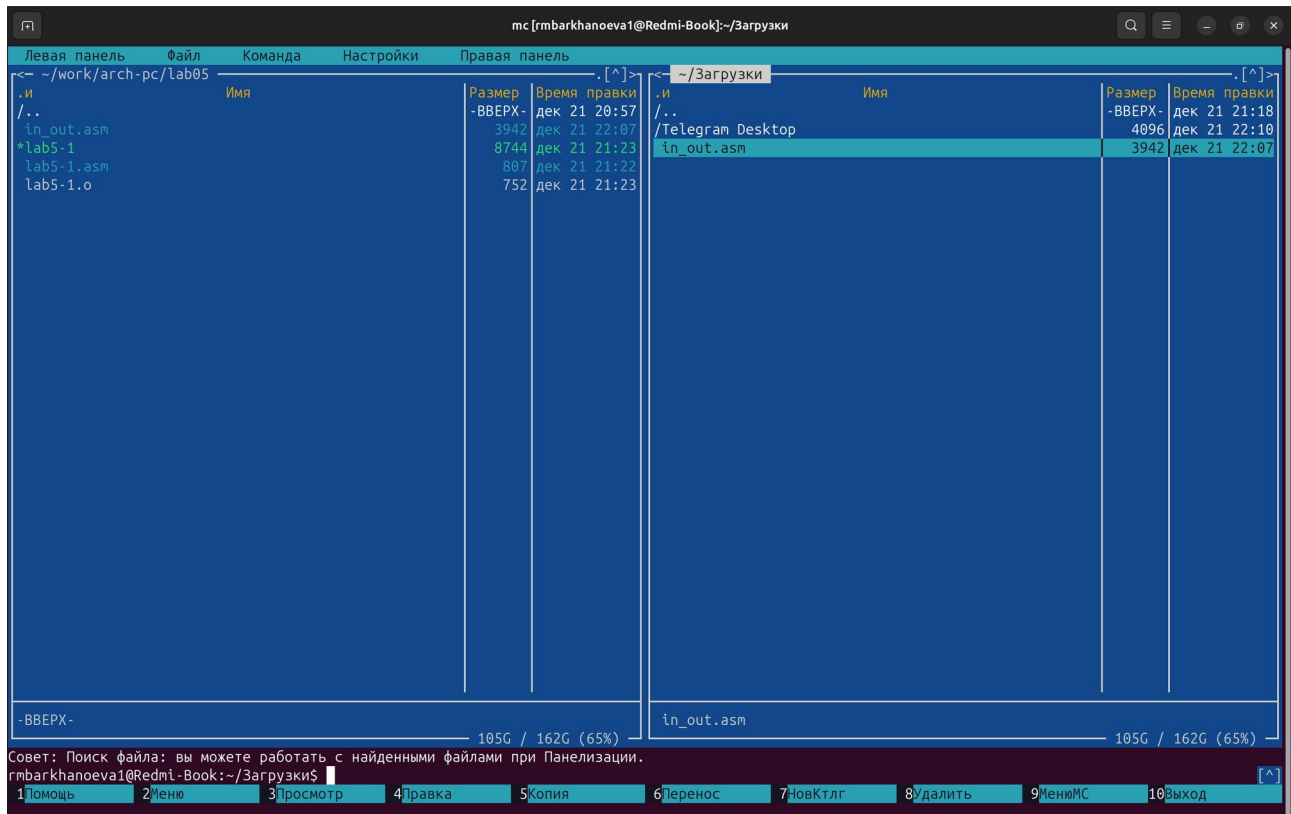


Рисунок 7.

Был скачен файл in\_out.asm со страницы курса в ТУИС в раздел  
Загрузки



Копирование в каталог (рисунок 8).

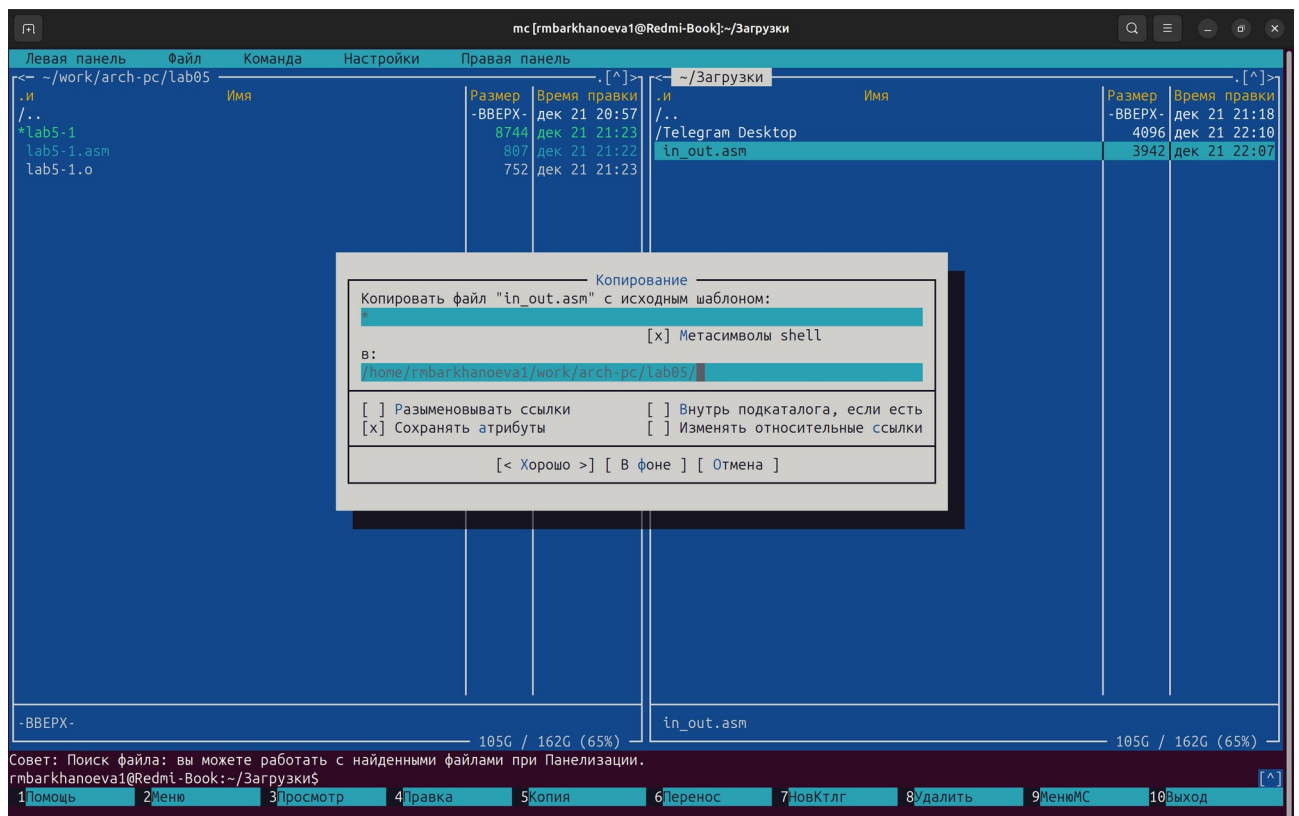


Рисунок 8.

Подключаемый файл `in_out.asm` должен лежать в том же каталоге, что и файл с программой, в которой он используется. На одной стороне нашей оболочки место, куда скачен файл, а на другой на каталог `lab05`. Нажатием клавиши `F5` мы копируем файл `in_out.asm` в каталог `lab05`.

## Создание копии (рисунок 9.)

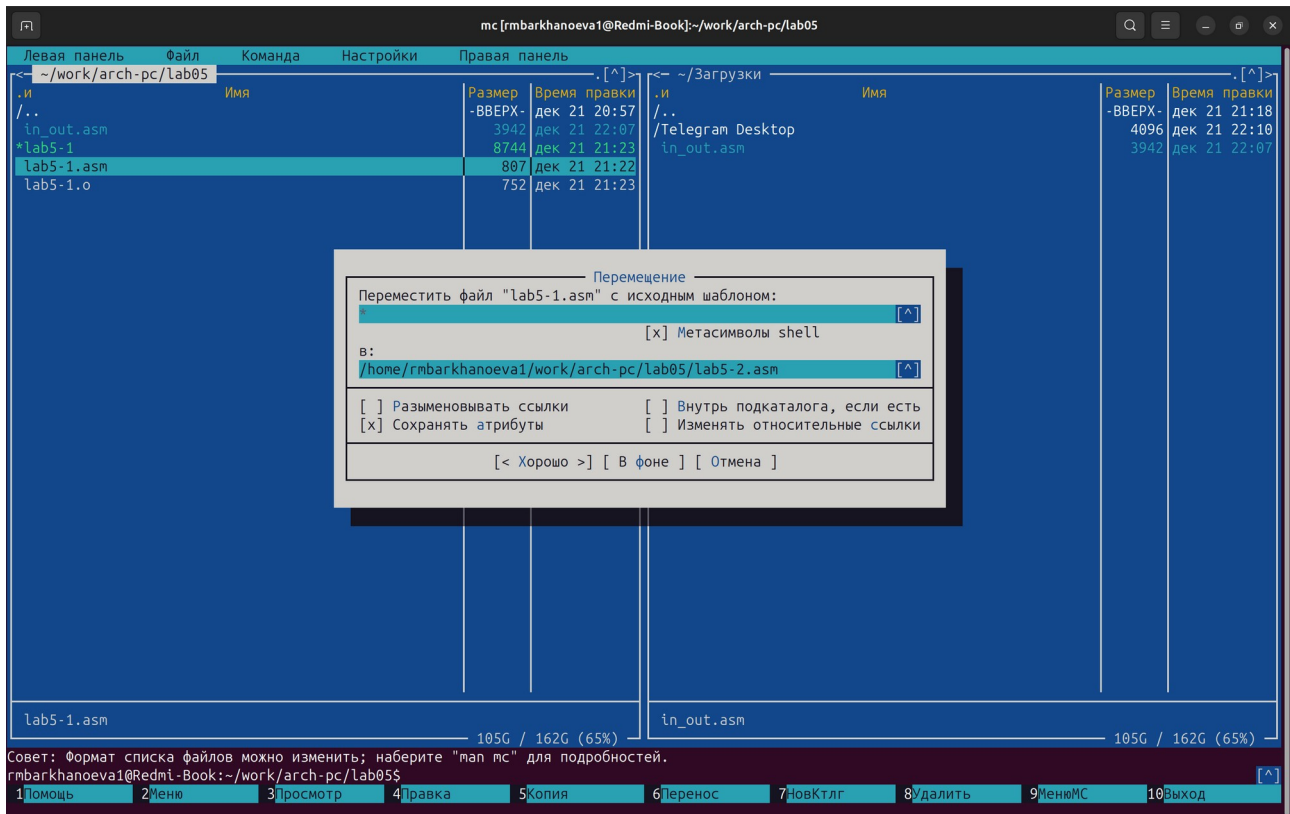


Рисунок 9.

Сделала копию файла lab5-1.asm с именем lab5-2.asm с помощью функциональной клавиши F6.

## Исправление текста программы (рисунок 10.)

```
GNU nano 7.2                                lab5-2.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

#include 'in_out.asm'          ; подключение внешнего файла

SECTION .data
msg:
    DB 'Введите строку: ', 0h

SECTION .bss
buf1:
    RESB 80                    ; буфер для ввода строки

SECTION .text
GLOBAL _start

_start:
    mov eax, msg                ; адрес сообщения
    call sprintf                ; вывести сообщение + перевод строки

    mov ecx, buf1               ; адрес буфера
    mov edx, 80                 ; максимальная длина
    call sread                  ; ввод строки

    call quit                   ; завершение программы
```

Рисунок 10.

Исправили текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (sprintf, sread и quit) в соответствии с листингом 5.2.

Тест программы (рисунок 11).

```
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o
lab5-2
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Раяна
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$
```

Рисунок 11.

Запуск программы прошел успешно.

Замена sprintLF на sprint (рисунок 12).

```
GNU nano 7.2 lab5-2.asm *
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

%include 'in_out.asm'      ; подключение внешнего файла

SECTION .data
msg:
    DB 'Введите строку: ', 0h

SECTION .bss
buf1:
    RESB 80                ; буфер для ввода строки

SECTION .text
GLOBAL _start

_start:
    mov eax, msg           ; адрес сообщения
    call sprint             ; вывести сообщение + перевод строки

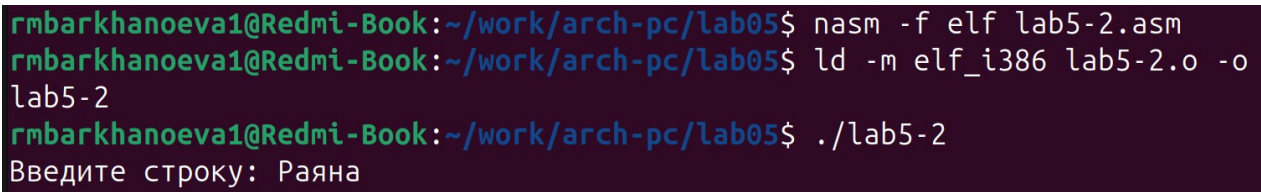
    mov ecx, buf1          ; адрес буфера
    mov edx, 80            ; максимальная длина
    call sread             ; ввод строки

    call quit              ; завершение программы
```

Рисунок 12.

Замена подпрограммы sprintLF на sprint.

Тест программы (рисунок 13).



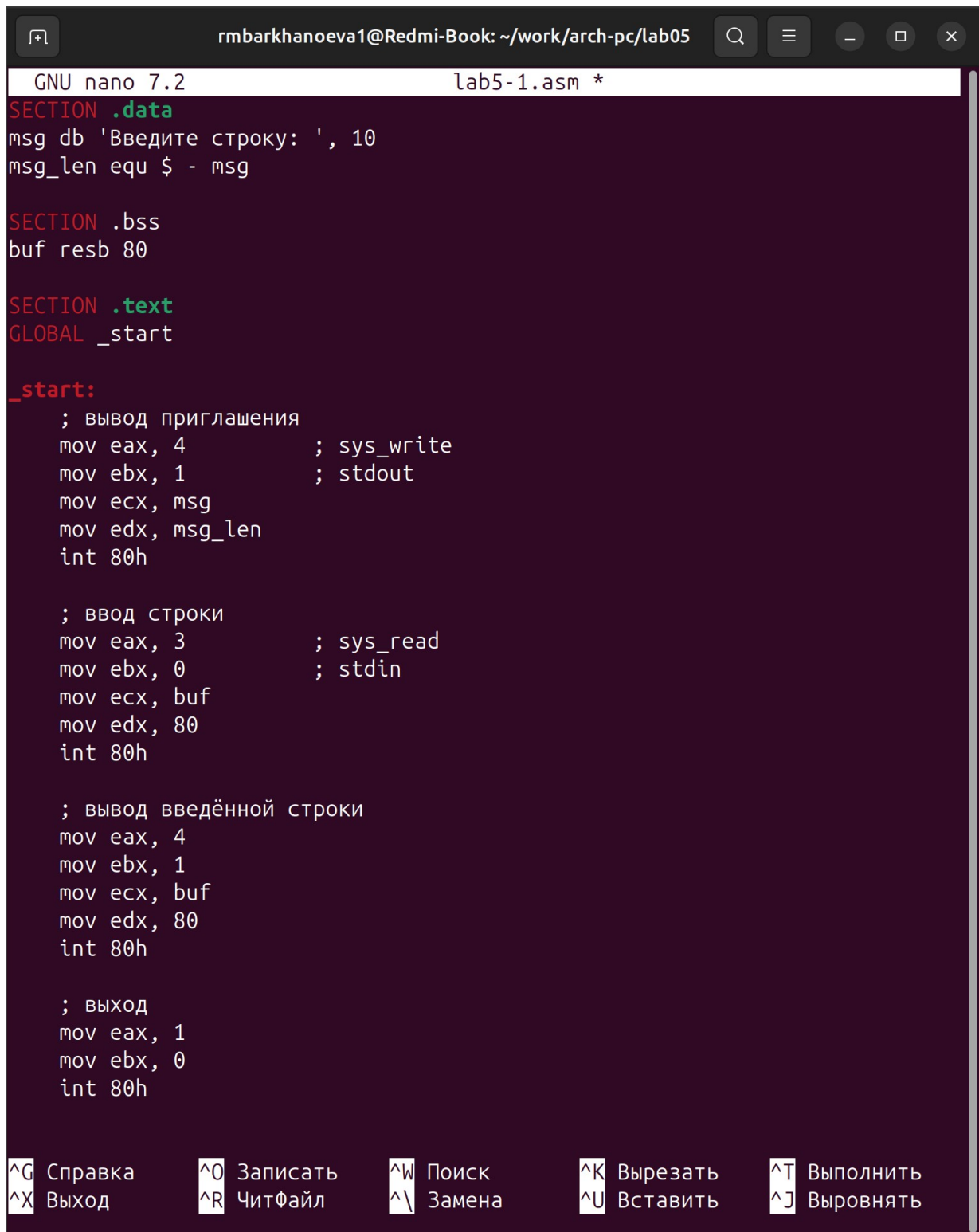
```
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o
lab5-2
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Раяна
```

Рисунок 13.

Разница в то, что `sprint` вывод на стандартный вывод, а `sprintLF` в конце добавляет перенос строки.

### 3. Самостоятельная работа.

Внесение изменений (рисунок 14).



```
GNU nano 7.2 lab5-1.asm *
SECTION .data
msg db 'Введите строку: ', 10
msg_len equ $ - msg

SECTION .bss
buf resb 80

SECTION .text
GLOBAL _start

_start:
    ; вывод приглашения
    mov eax, 4          ; sys_write
    mov ebx, 1          ; stdout
    mov ecx, msg
    mov edx, msg_len
    int 80h

    ; ввод строки
    mov eax, 3          ; sys_read
    mov ebx, 0          ; stdin
    mov ecx, buf
    mov edx, 80
    int 80h

    ; вывод введенной строки
    mov eax, 4
    mov ebx, 1
    mov ecx, buf
    mov edx, 80
    int 80h

    ; выход
    mov eax, 1
    mov ebx, 0
    int 80h
```

^G Справка    ^O Записать    ^W Поиск    ^K Вырезать    ^T Выполнить  
^X Выход    ^R ЧитФайл    ^\ Замена    ^U Вставить    ^J Выворнять

Рисунок 14.

Внесла изменения в программу (без использования внешнего файла in\_out.asm)

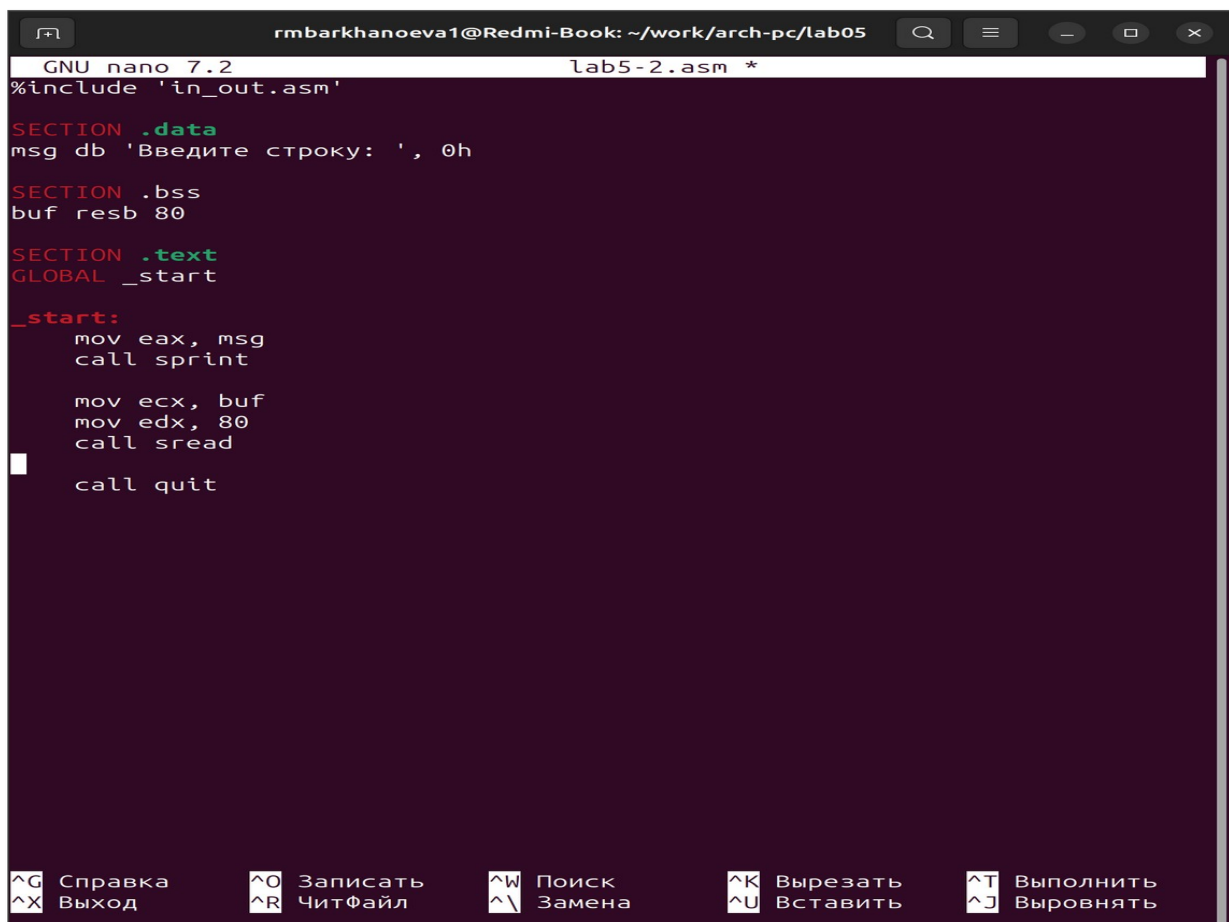
Тест (рисунок 15).

```
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1.o -o lab5-2
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Барханоева
rmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$
```

Рисунок 15.

Вывод корректный

Исправление кода (рисунок 16).



```
rmbarkhanoeva1@Redmi-Book: ~/work/arch-pc/lab05
GNU nano 7.2 lab5-2.asm *
%include 'in_out.asm'

SECTION .data
msg db 'Введите строку: ', 0h

SECTION .bss
buf resb 80

SECTION .text
GLOBAL _start

_start:
    mov eax, msg
    call sprint

    mov ecx, buf
    mov edx, 80
    call sread

    call quit
```

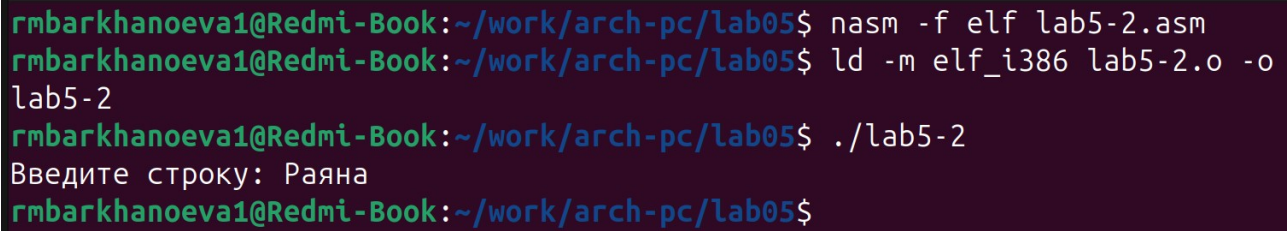
Legend:

<b>^G</b> Справка	<b>^O</b> Записать	<b>^W</b> Поиск	<b>^K</b> Вырезать	<b>^T</b> Выполнить
<b>^X</b> Выход	<b>^R</b> ЧитФайл	<b>^_</b> Замена	<b>^U</b> Вставить	<b>^J</b> Выводить

Рисунок 16.

Исправила текст программы с использованием подпрограммы из внешнего файла in\_out.asm.

Тест (рисунок 17).



```
gmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
gmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o
lab5-2
gmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Раяна
gmbarkhanoeva1@Redmi-Book:~/work/arch-pc/lab05$
```

Рисунок 17.

Вывод корректный



## **Вывод**

В ходе лабораторной работы были изучены основы работы с Midnight Commander и базовая структура программы на языке ассемблера NASM, а также получены начальные навыки использования инструкций `mov` и `int` и системных вызовов GNU/Linux.