

# **РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук Кафедра  
прикладной информатики и теории вероятностей**

## **ОТЧЕТ**

### **ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

дисциплина: Архитектура компьютера

Студент: Джумабаев Дамир

Группа: НКАбд-02-25

**МОСКВА**

# **Содержание**

**1 Цель работы**

**2 Задание**

**3 Теоретическое введение**

**4 Выполнение лабораторной работы**

**4.1 Основы работы с Midnight Commander**

**4.2 Работа с NASM**

**4.3 Подключение внешнего файла**

**4.4 Задание для самостоятельной работы**

**1 Цель работы**

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

## **2 Задание**

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

## **3 Теоретическое введение**

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. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается входе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции. Data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: -DB(define byte) — определяет переменную размером в 1 байт; - DW(define word)— определяет переменную размером в 2 байта (слово); - DD (define double word)—определяет переменную размером в 4 байта (двойное слово); - DQ(define quad word)—определяет переменную размером в 8 байт(учетверённое слово); -DT(define ten bytes)—определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. mov dst,src Здесь операнд dst — приёмник, asrc

—источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения(const).

Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером. `int n` Здесь `n`—номер прерывания, принадлежащий диапазону 0 – 255. При программировании в Linux с использованием вызовов ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления).

## 4 Выполнение лабораторной работы

### 4.1 Основы работы с Midnight Commander

Введя соответствующую команду в терминале(рис.4.1),я открываю Midnight Commander (рис.4.2)

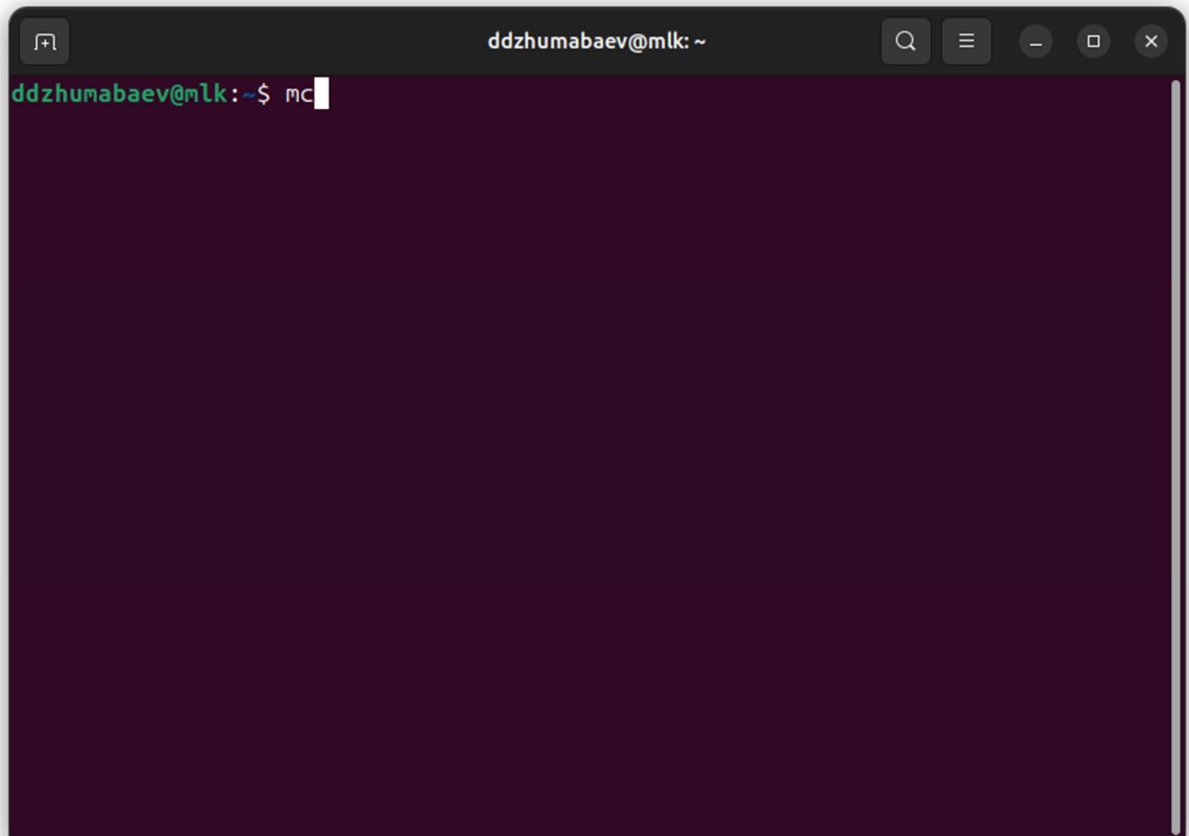


Рис 4.1 Открытие Midnight Commander

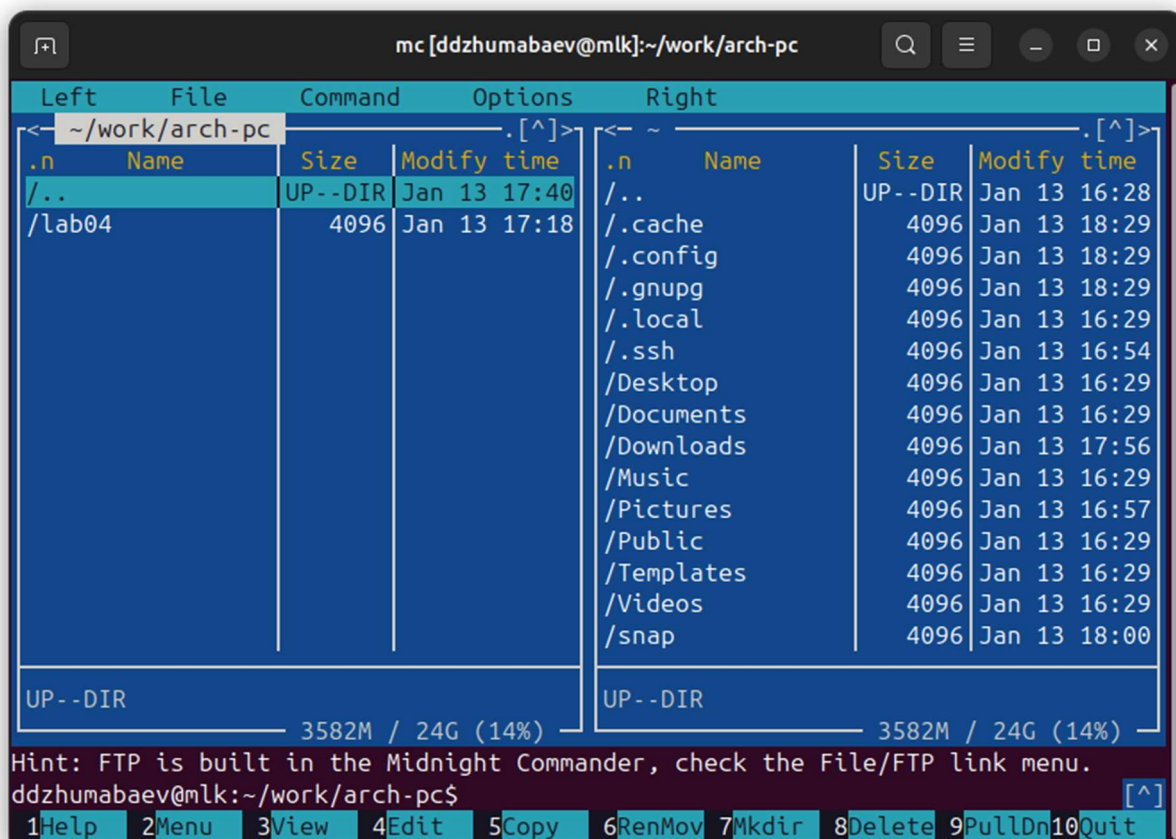


Рис. 4.2: Интерфейс Midnight Commander

Перехожу в созданный каталог в предыдущей лабораторной работе(рис.4.3).

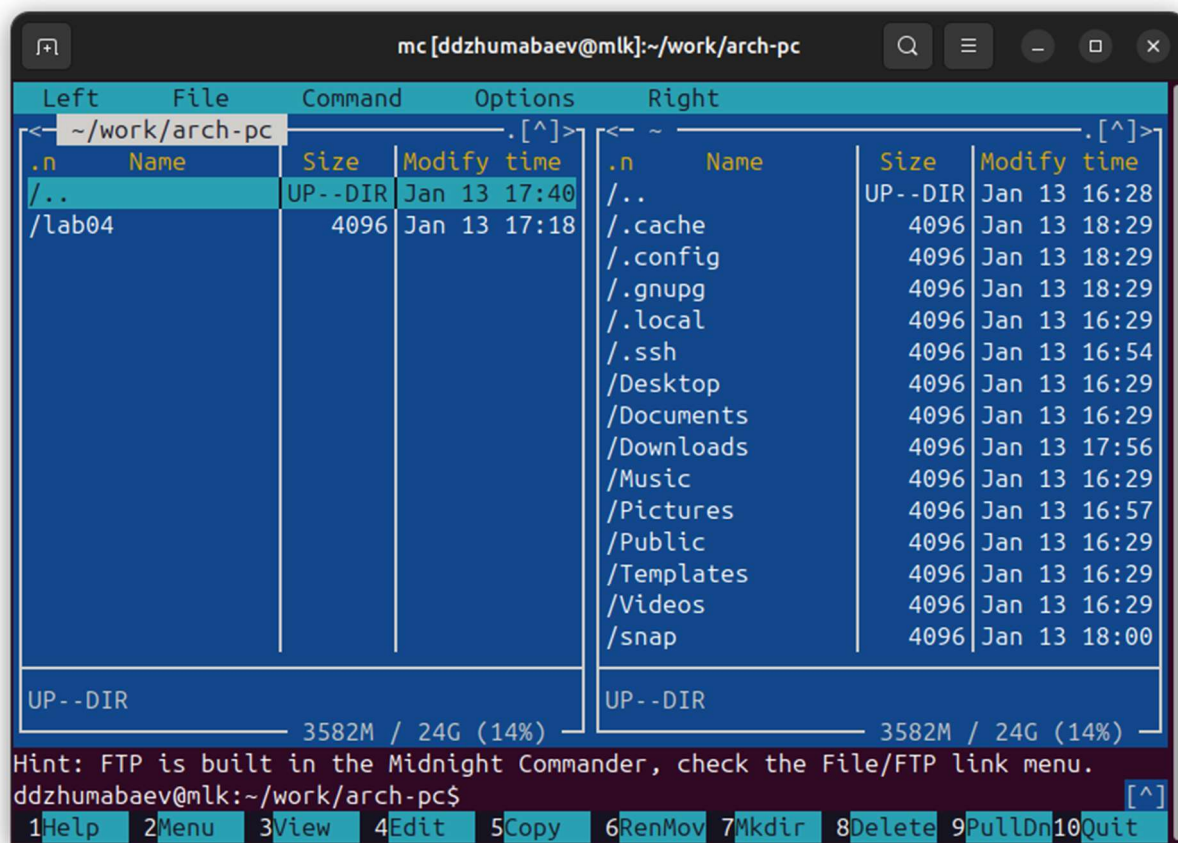


Рис 4.3 Открытый каталог arch-pc

С помощью функциональной клавиши, я создаю под каталог lab05, в котором буду работать (рис. 4.4).

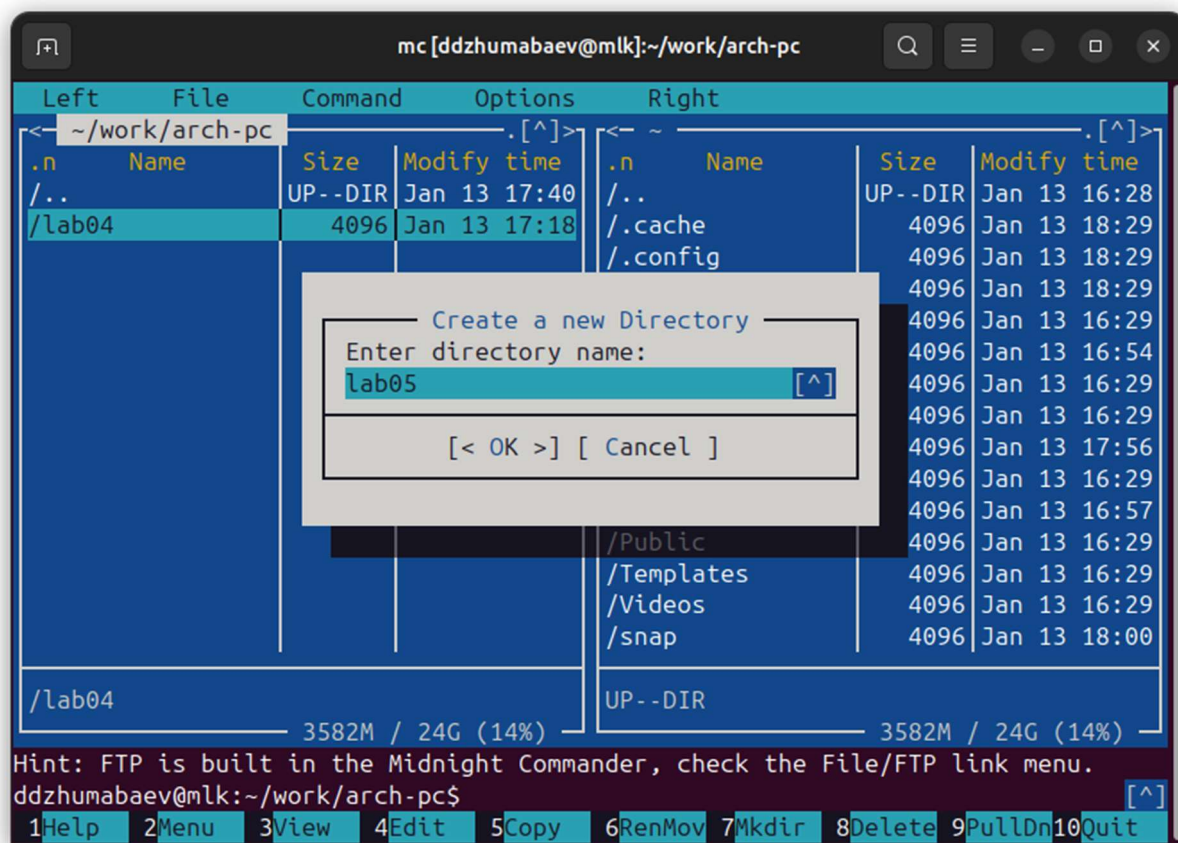


Рис. 4.4: Создание рабочего подкаталога

В строке ввода ввожу команду touch и создаю файл(рис.4.5).

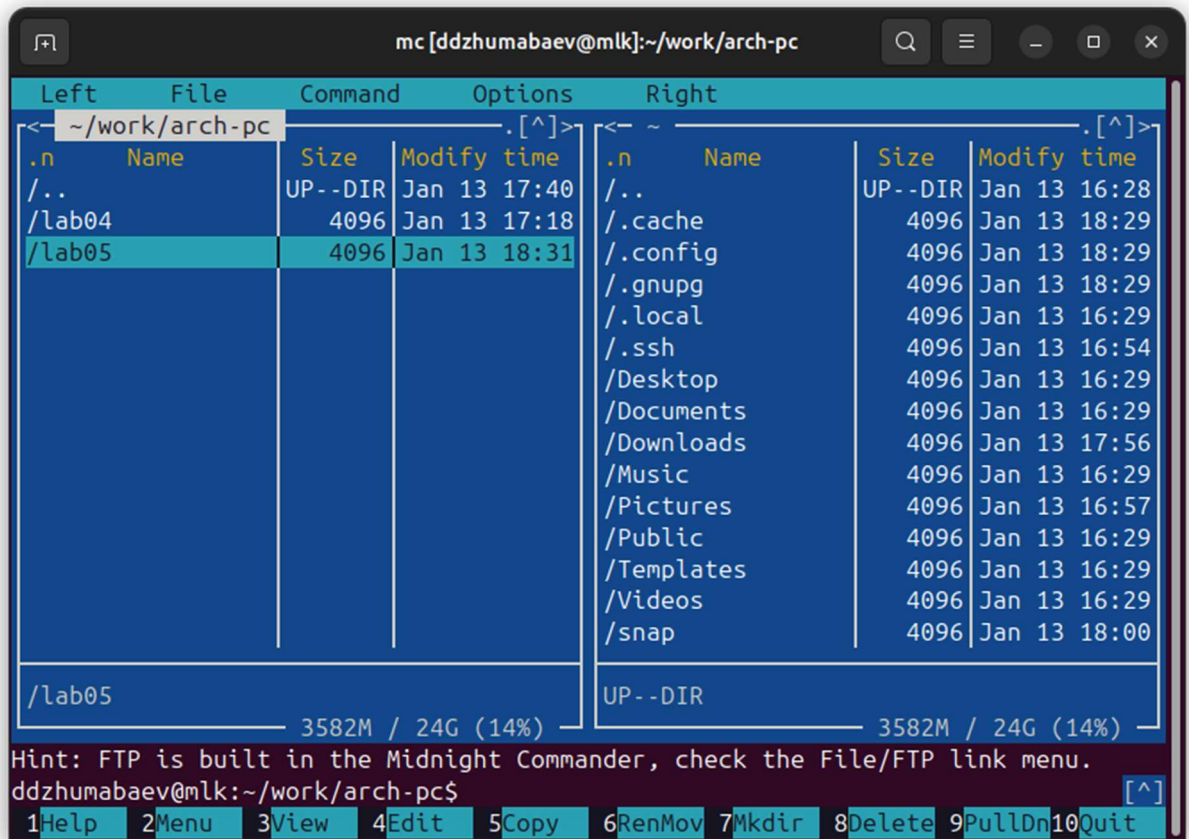
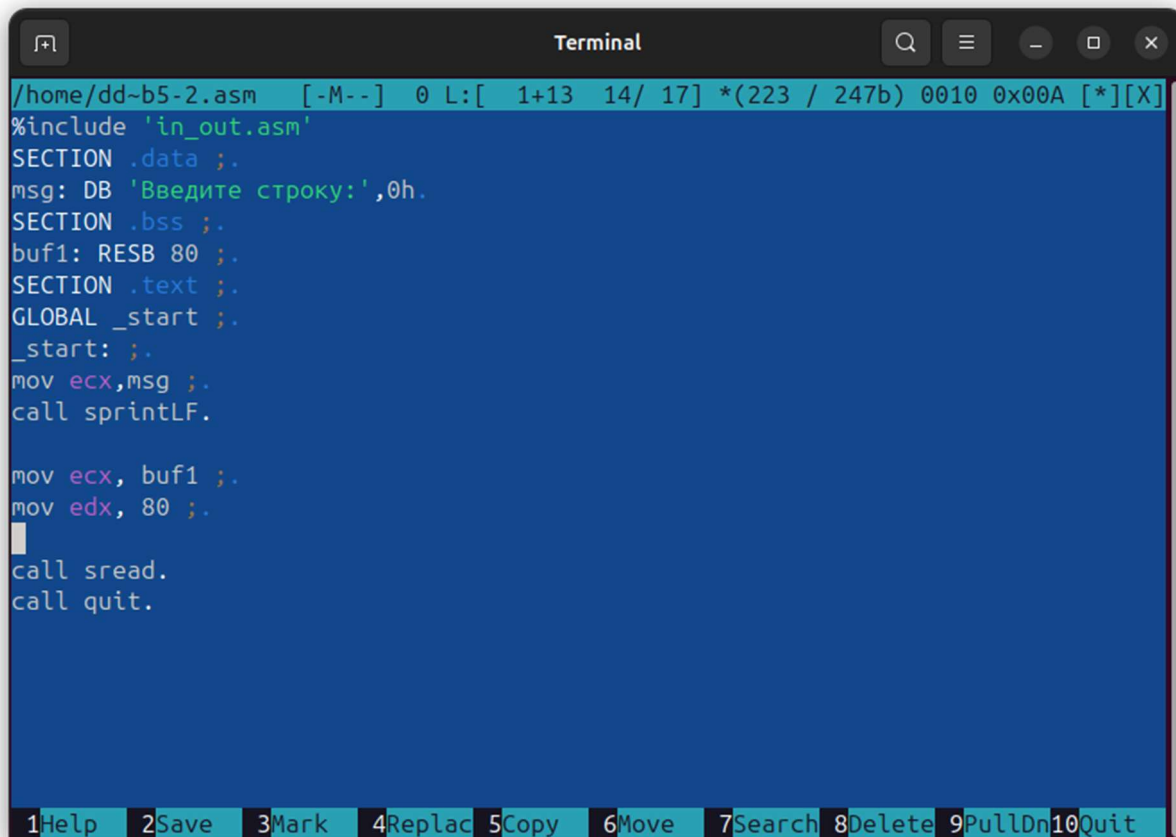


Рис. 4.5: Создание файла в Midnight Commander

## 4.2 Работа в NASM

С помощью F4 открываю только что созданный файл и вношу код с листинга (рис. 4.6).



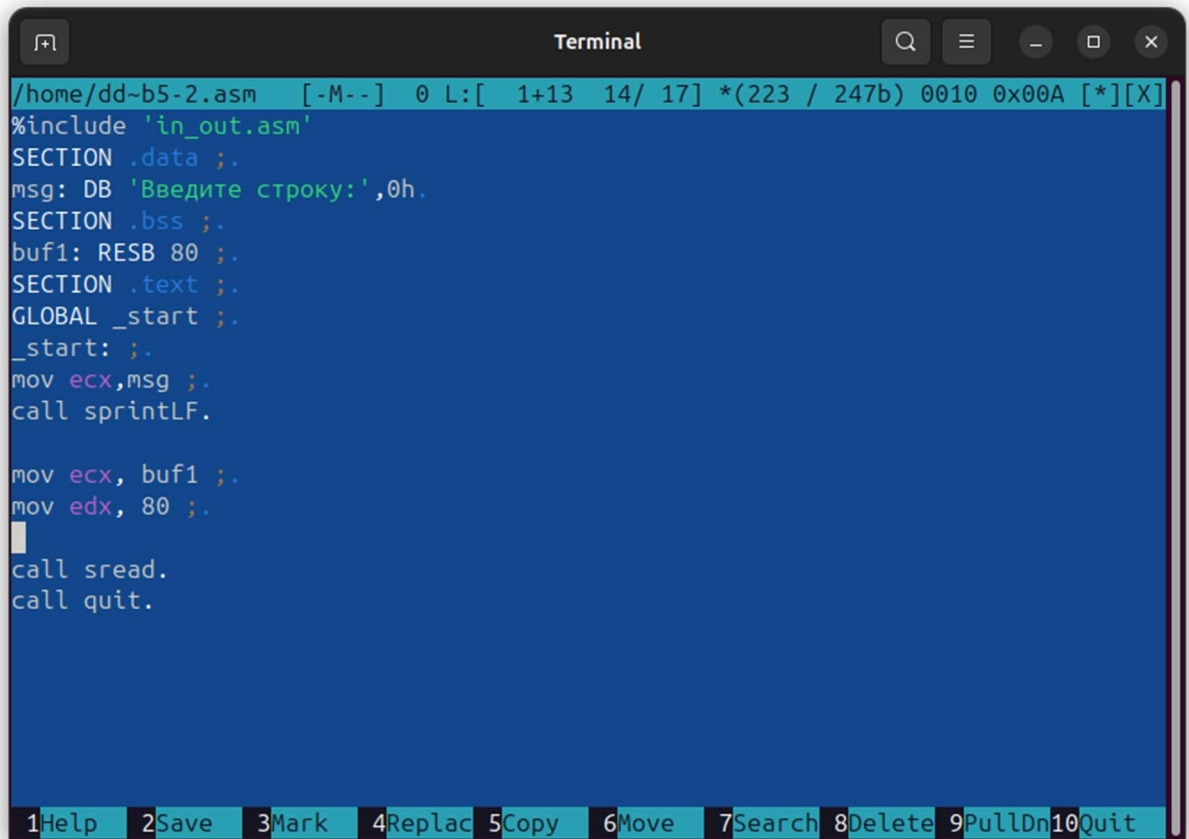
```
/home/dd~b5-2.asm  [-M--]  0 L:[ 1+13 14/ 17] *(223 / 247b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .data ;.
msg: DB 'Введите строку:',0h.
SECTION .bss ;.
buf1: RESB 80 ;.
SECTION .text ;.
GLOBAL _start ;.
_start: ;.
mov ecx,msg ;.
call sprintLF.

mov ecx, buf1 ;.
mov edx, 80 ;.
call sread.
call quit.
```

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit

Рис 4.6 Редактирование файла в Midnight Commander

Проверяю сохраненные изменения с помощью клавиши F3(рис.4.7).



The image shows a terminal window titled "Terminal" with a dark blue background. At the top, a status bar displays file information: `/home/dd~b5-2.asm`, `[-M--]`, `0 L:[ 1+13 14/ 17]`, and `*(223 / 247b) 0010 0x00A [*][X]`. The main area contains assembly code: `%include 'in_out.asm'`, `SECTION .data ;.`, `msg: DB 'Введите строку:',0h.`, `SECTION .bss ;.`, `buf1: RESB 80 ;.`, `SECTION .text ;.`, `GLOBAL _start ;.`, `_start: ;.`, `mov ecx,msg ;.`, `call sprintLF.`, `mov ecx, buf1 ;.`, `mov edx, 80 ;.`, `call sread.`, and `call quit.`. A menu bar at the bottom lists: `1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit`.

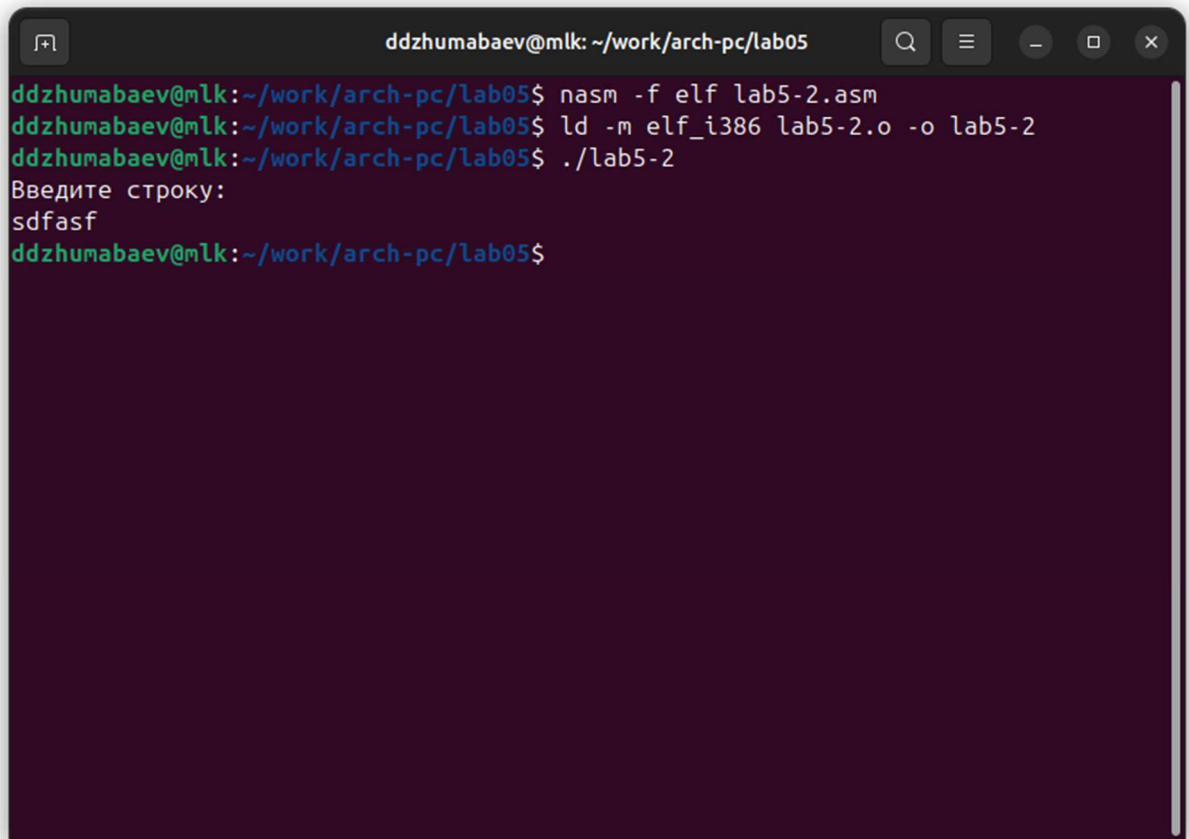
```
/home/dd~b5-2.asm  [-M--]  0 L:[ 1+13 14/ 17] *(223 / 247b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .data ;.
msg: DB 'Введите строку:',0h.
SECTION .bss ;.
buf1: RESB 80 ;.
SECTION .text ;.
GLOBAL _start ;.
_start: ;.
mov ecx,msg ;.
call sprintLF.

mov ecx, buf1 ;.
mov edx, 80 ;.
call sread.
call quit.
```

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit

Рис. 4.7: Проверка сохранения сделанных изменений

Транслирую и компоную измененный файл, запускаю(рис.4.8).

A terminal window with a dark background and light green text. The window title is 'ddzhumabaev@mlk: ~/work/arch-pc/lab05'. The terminal shows the following commands and output:

```
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
sdfasf
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

Рис. 4.8: Трансляция, компоновка и последующий запуск программы

### 4.3 Подключение внешнего файла

Скачанный с ТУИС файл сохраняю в общую папку на своем компьютере, на виртуальной машине в интерфейсе Midnight Commander перехожу в директорию общей папки, копирую файл в рабочий подкаталог.(рис.4.9).

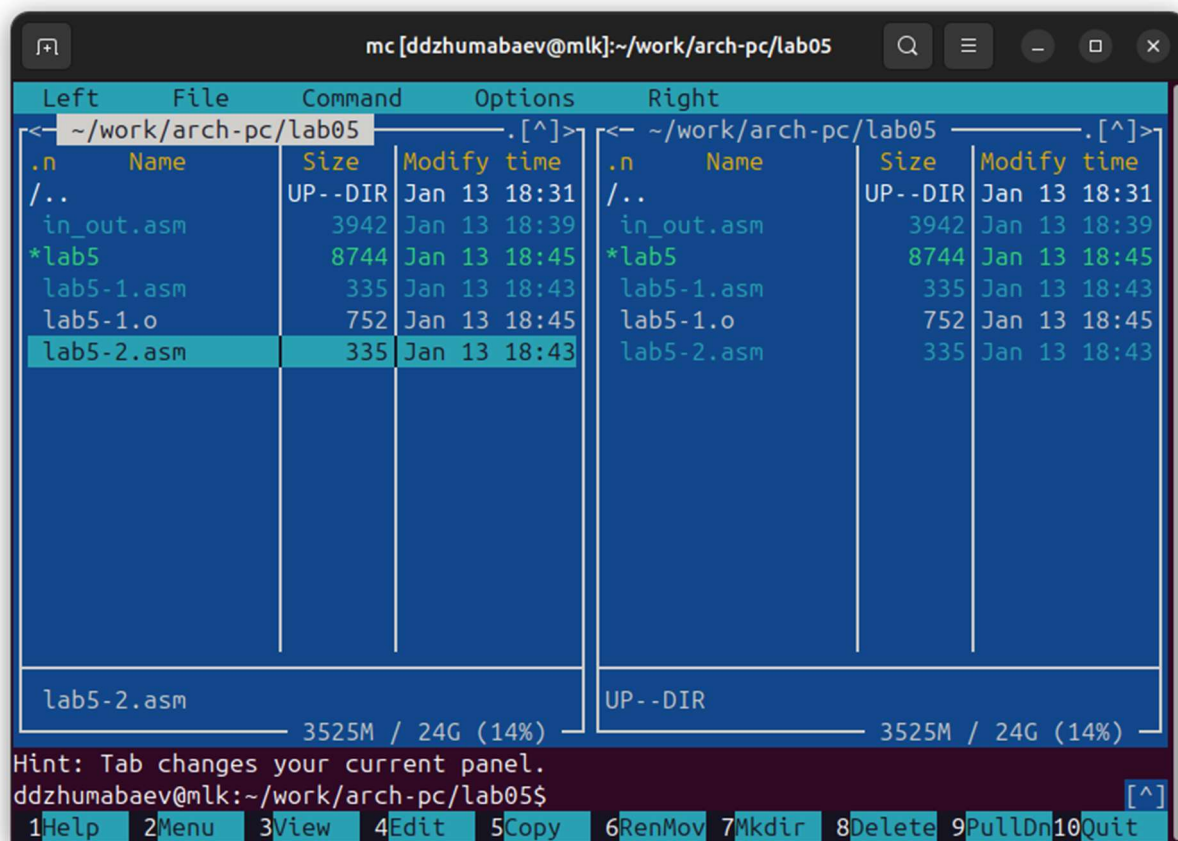


Рис. 4.9: Копирование файла в рабочий каталог

Создаю копию файла для последующей работы с ним(рис.4.10).

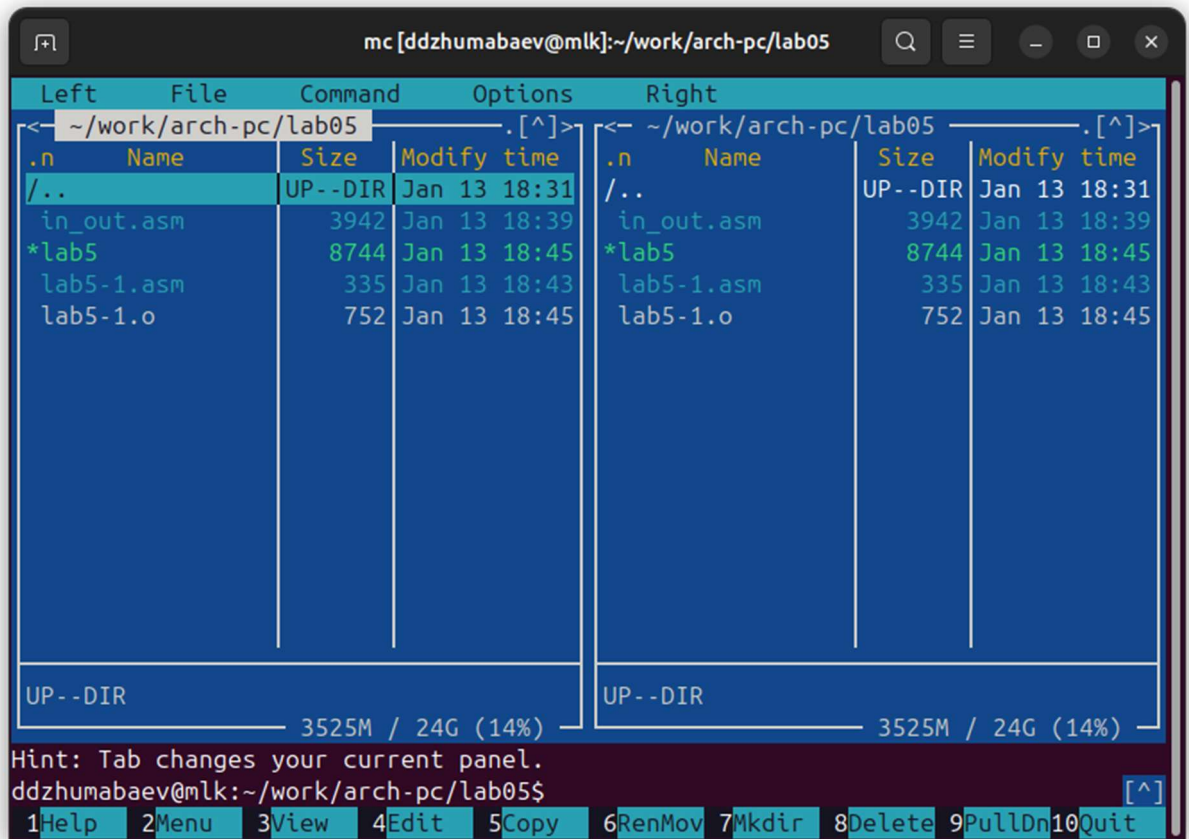
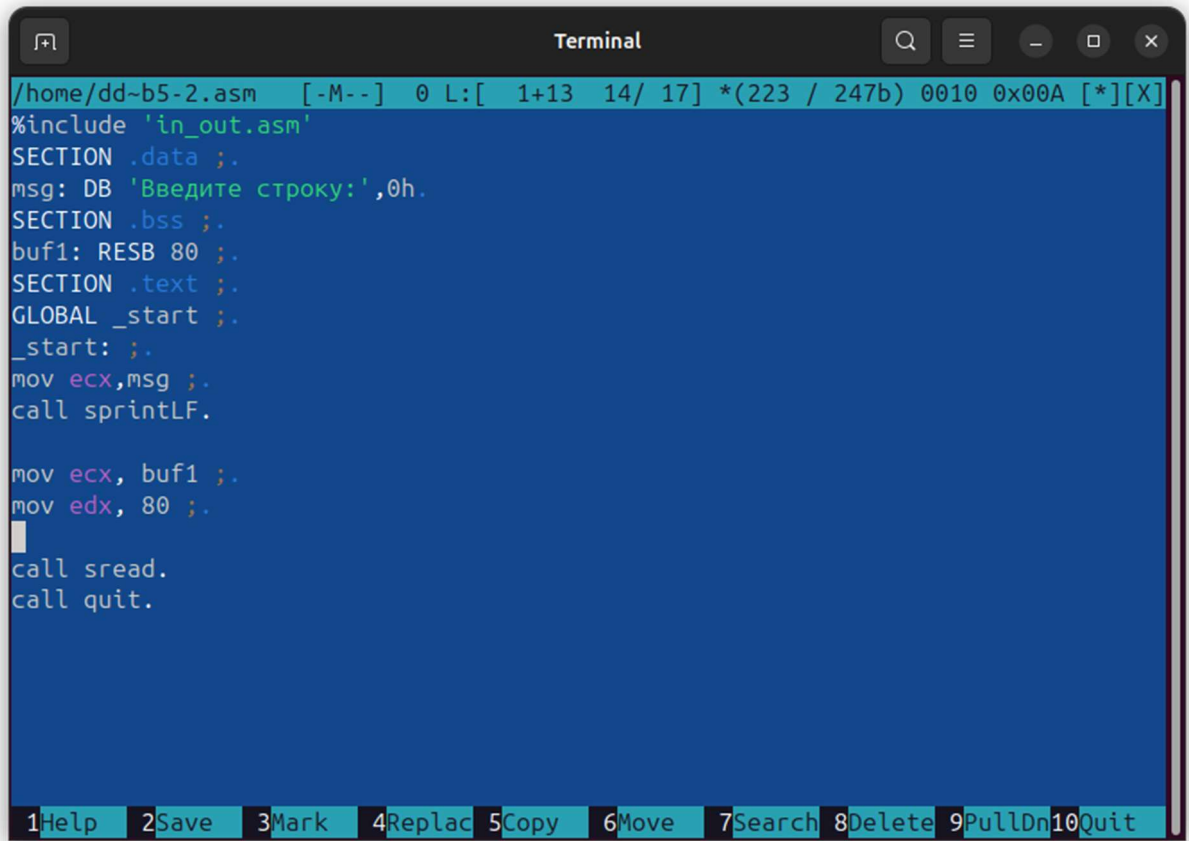


Рис. 4.10: Создание копии файлов Midnight Commander

В копии файла подключаю подпрограмм из подключенного файла(рис.4.11).



The image shows a terminal window titled "Terminal" with a dark background and light blue text. The terminal displays assembly code for a program. The code includes a header line with file path and metadata, followed by include directives, section declarations for .data, .bss, and .text, and various assembly instructions for memory allocation, string handling, and system calls. The code is as follows:

```
/home/dd~b5-2.asm [-M--] 0 L:[ 1+13 14/ 17] *(223 / 247b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .data ;.
msg: DB 'Введите строку:',0h.
SECTION .bss ;.
buf1: RESB 80 ;.
SECTION .text ;.
GLOBAL _start ;.
_start: ;.
mov ecx,msg ;.
call sprintLF.

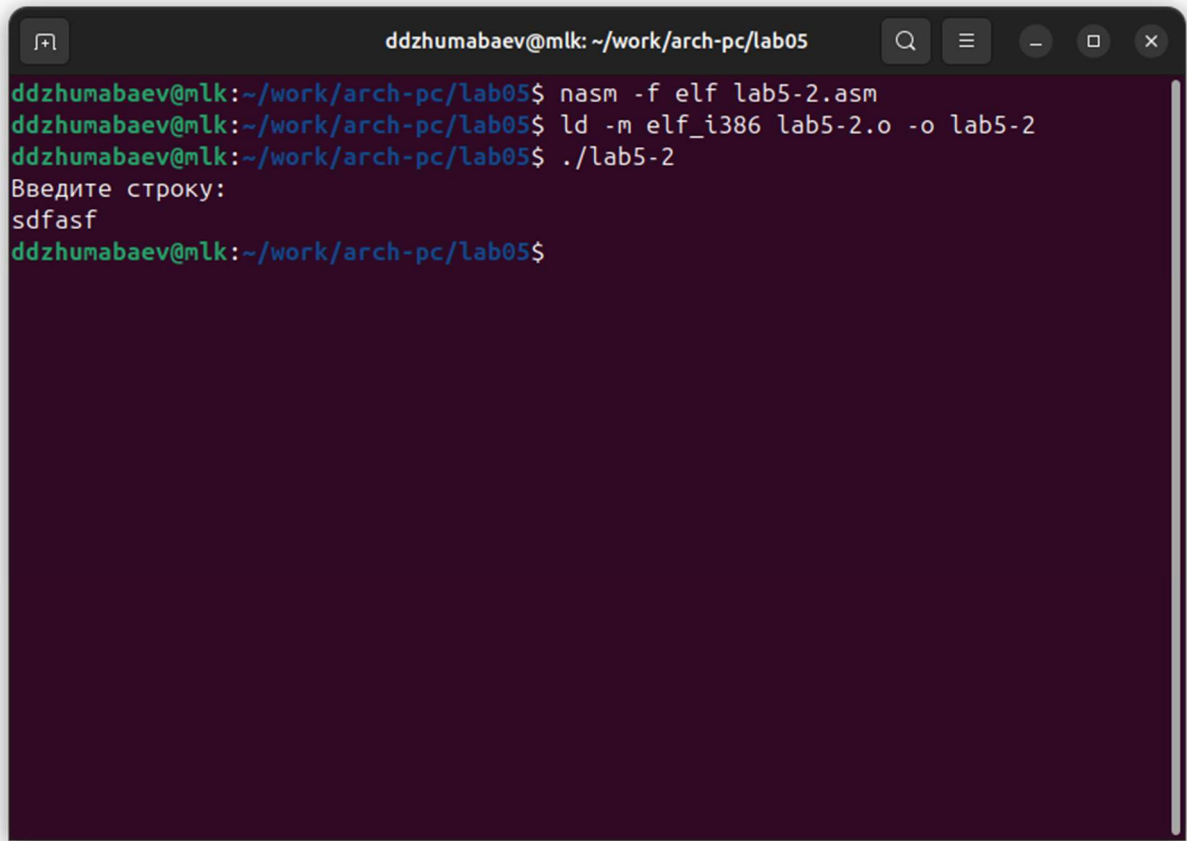
mov ecx, buf1 ;.
mov edx, 80 ;.

call sread.
call quit.
```

At the bottom of the terminal window, there is a menu bar with the following items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9PullDn, 10Quit.

Рис. 4.11: Изменение программы

Транслирую, компоную и запускаю программу с подключенным файлом (рис. 4.12).



```
ddzhumabaev@mlk: ~/work/arch-pc/lab05
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
sdfasf
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

Рис. 4.12: Запуск измененной программы

Редактирую файл и заменяю в нем под программу sprintLF на sprint. Разница под программ в том, что вторая вызывает ввод на той же строке(рис.4.13).

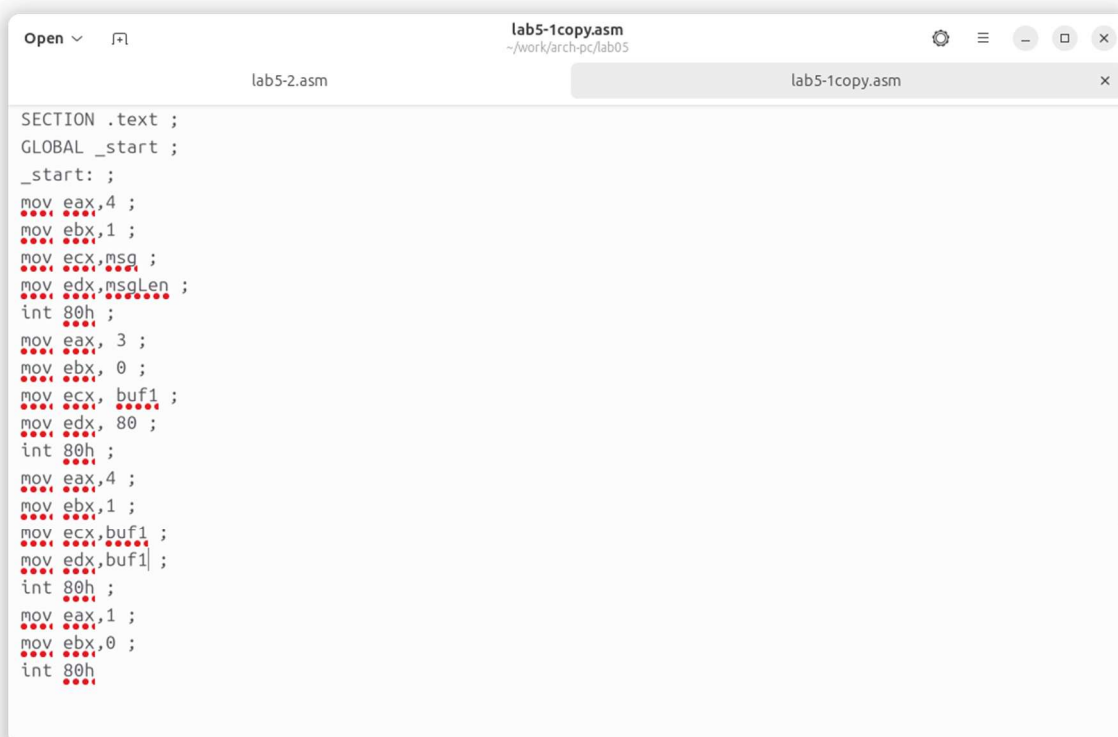
```
ddzhumabaev@mlk: ~/work/arch-pc/lab05
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
sdfasf
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

```
ddzhumabaev@mlk: ~/work/arch-pc/lab05
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1copy.o -o lab5-1copy
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
DamiR
DamiR
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

Рис. 4.13: Запуск изменной программы с другой под программой

## 4.4 Задание для самостоятельной работы

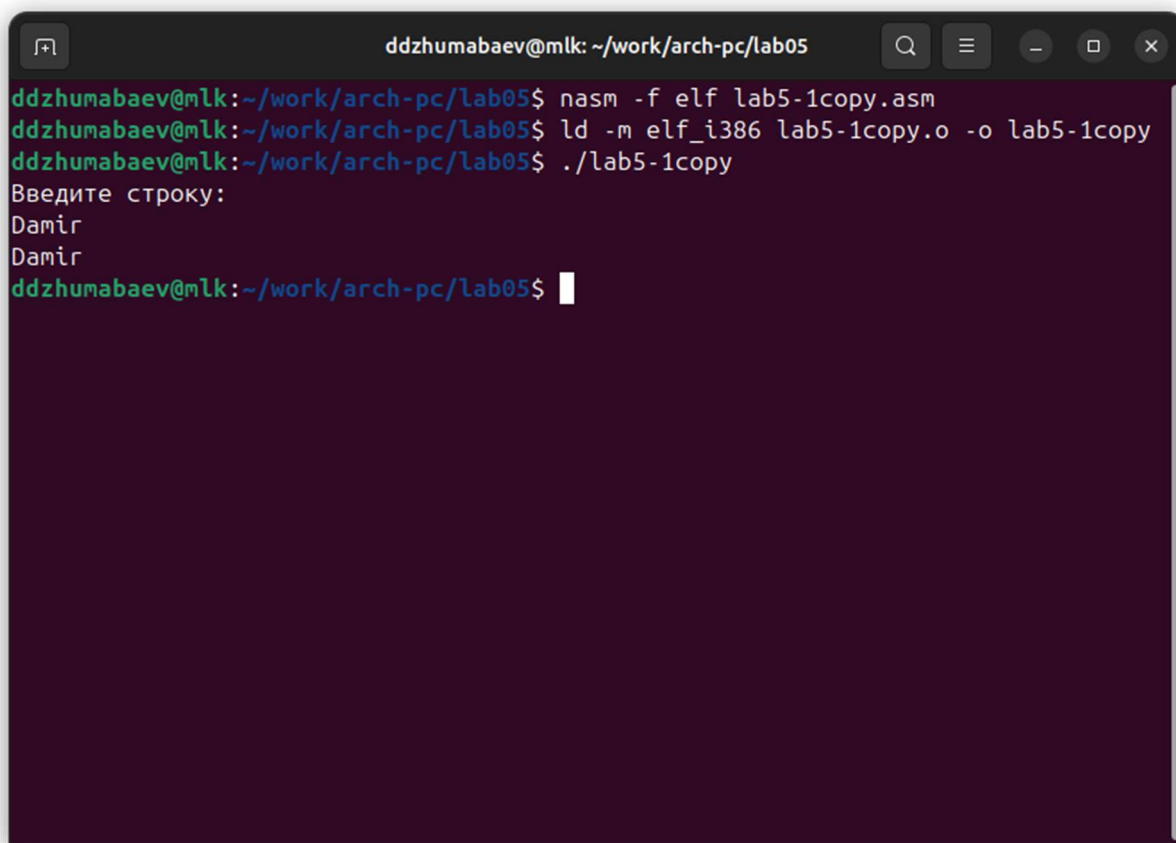
Создаю копию lab5-1.asm, редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры(рис.4.14).



```
SECTION .text ;
GLOBAL _start ;
_start: ;
mov eax,4 ;
mov ebx,1 ;
mov ecx,msg ;
mov edx,msgLen ;
int 80h ;
mov eax, 3 ;
mov ebx, 0 ;
mov ecx, buf1 ;
mov edx, 80 ;
int 80h ;
mov eax,4 ;
mov ebx,1 ;
mov ecx,buf1 ;
mov edx,buf1 ;
int 80h ;
mov eax,1 ;
mov ebx,0 ;
int 80h ;
```

Рис 4.6 Передача файла компоновщику

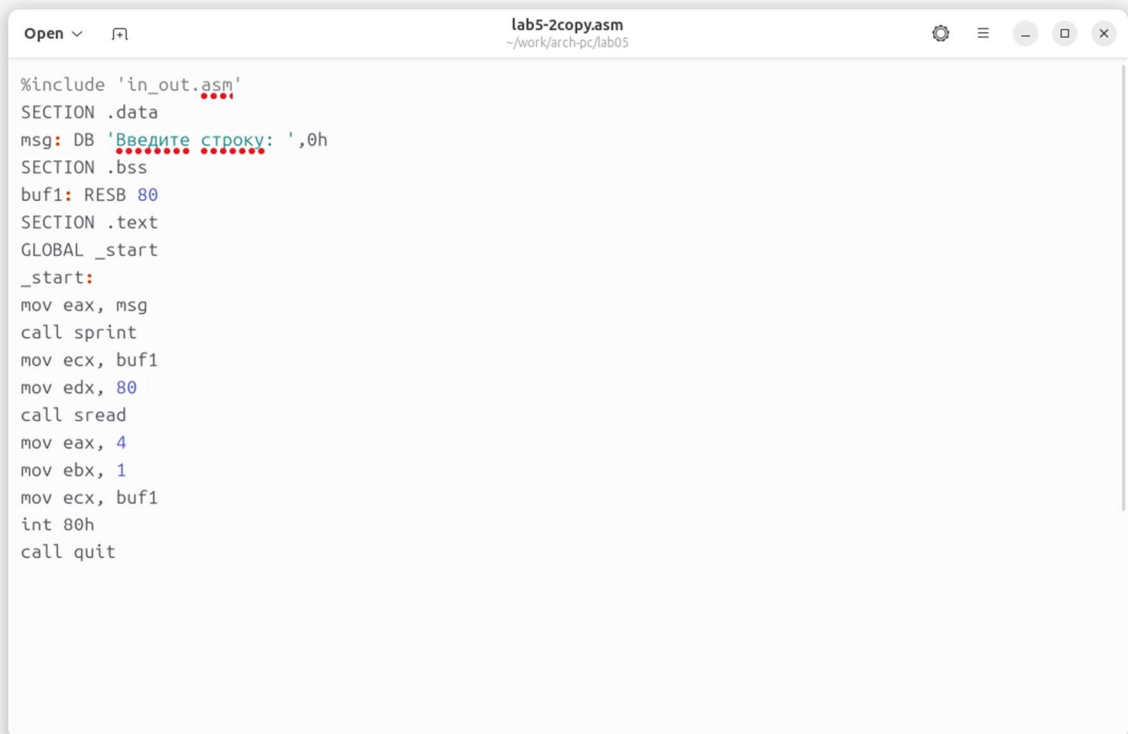
Рис. 4.14: Редактирование копии Транслирую, компоную и запускаю свою программу(рис.4.15).



```
ddzhumabaev@mlk: ~/work/arch-pc/lab05
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1copy.o -o lab5-1copy
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
Damiг
Damiг
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

Рис. 4.15: Запуск своей программы

Создаю копию lab5-2.asm, редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры(рис.4.16).

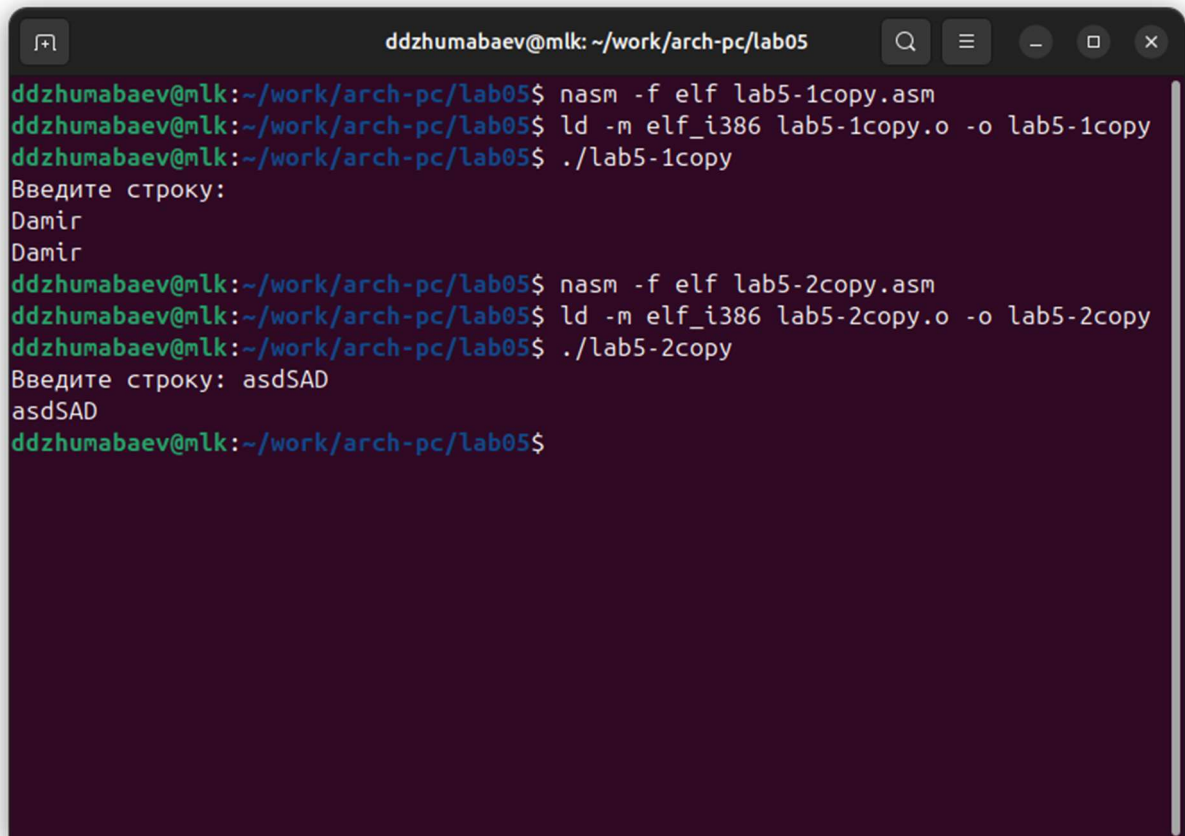


```
Open  lab5-2copy.asm
~/work/arch-pc/lab05

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit
```

Рис. 4.16: Редактирование копии

Транслирую, компоную и запускаю свою программу (рис.4.17).

A terminal window with a dark purple background and white text. The window title is 'ddzhumabaev@mlk: ~/work/arch-pc/lab05'. The terminal shows the following commands and output:

```
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1copy.o -o lab5-1copy
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
Damiг
Damiг
ddzhumabaev@mlk:~/work/arch-pc/lab05$ nasm -f elf lab5-2copy.asm
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2copy.o -o lab5-2copy
ddzhumabaev@mlk:~/work/arch-pc/lab05$ ./lab5-2copy
Введите строку: asdSAD
asdSAD
ddzhumabaev@mlk:~/work/arch-pc/lab05$
```

Рис. 4.17: Запуск своей программы

## 5 Выводы

При выполнении данной лабораторной работы я приобрёл практические навыки работы в Midnight Commander, а так же освоил инструкции языка ассемблера `mov` и `int`.