

**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: Ванюшкина Т.В.

Группа: НКАбд-01-24

Студ.билет: 1132246713

**МОСКВА**

2024г

## **СОДЕРЖАНИЕ**

1. Цель работы.....	3
2. Теоретическое введение.....	4
3. Выполнение лабораторной работы.....	5-14
3.1 Основы работы в тс.....	5-8
3.2 Подключение внешнего файла in_out.asm.....	8-12
3.3 Задание для самостоятельной работы.....	12-14
4. Выводы.....	15
5. Список литературы.....	16

## **1. Цель работы**

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## **2.Теоретическое введение**

Midnight Commander (или просто mc)—это программа,которая позволяетпросматривать 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.

## 3.Выполнение лабораторной работы

### 3.1 Основы работы в mc

Открываю Midnight Commander:

```
tatyana@vbox:~$ mc
```

Рис 1

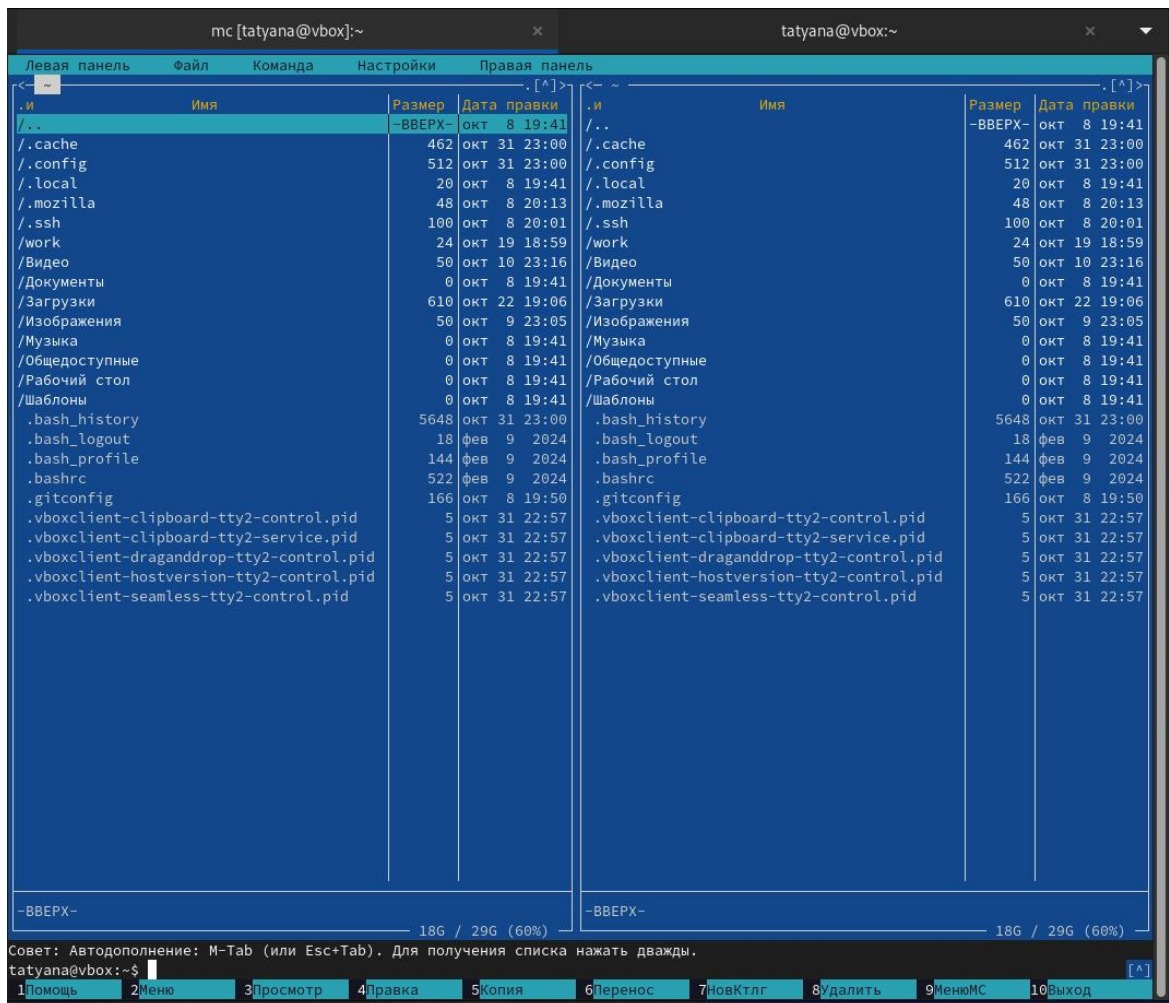


Рис 2

Пользуясь клавишами ↑, ↓ и Enter перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4:

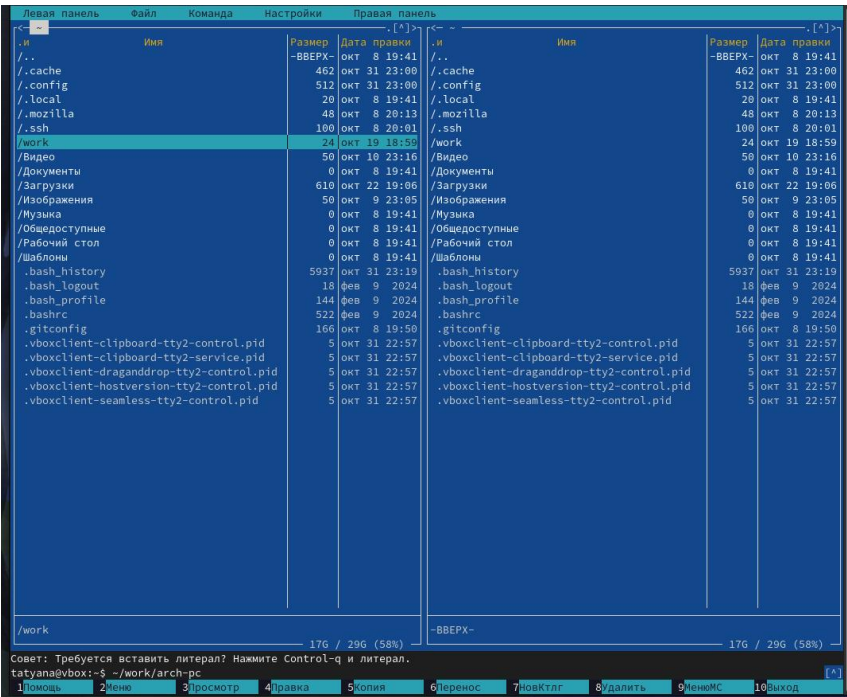


Рис 3

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm:

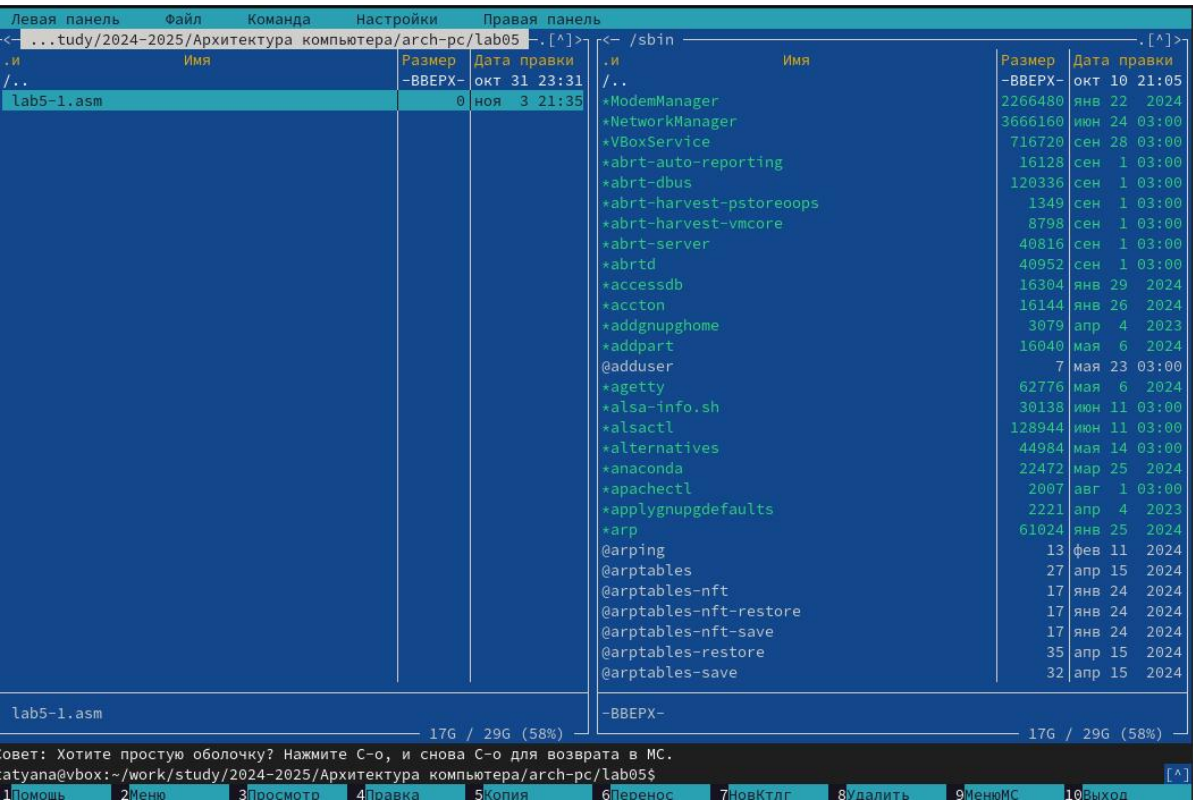


Рис 4

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встречном редакторе:

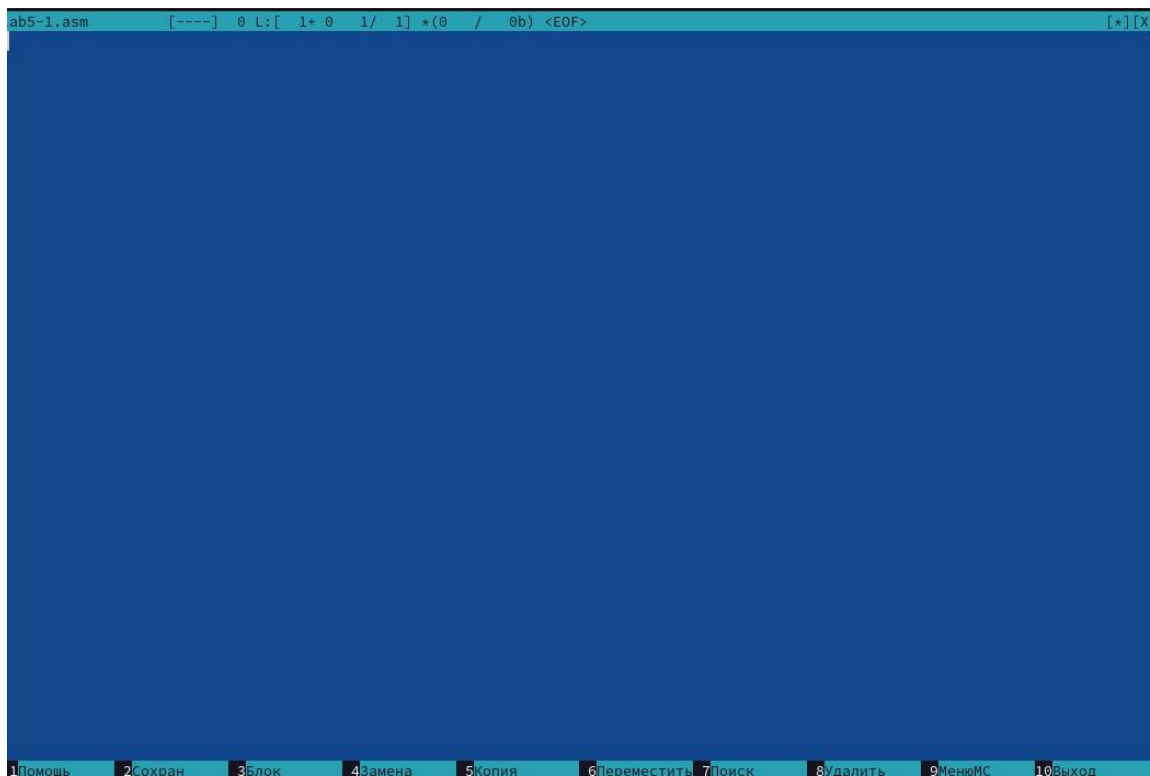


Рис 5

Ввожу текст программы из листинга 5.1:

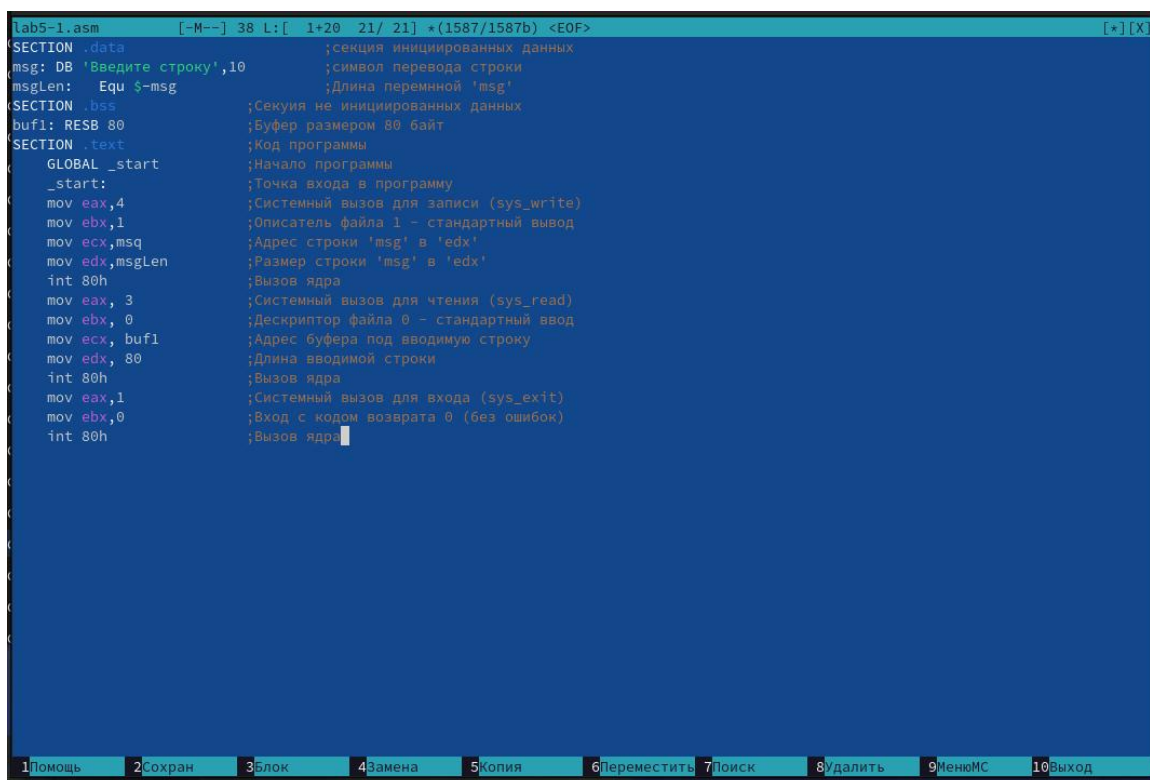
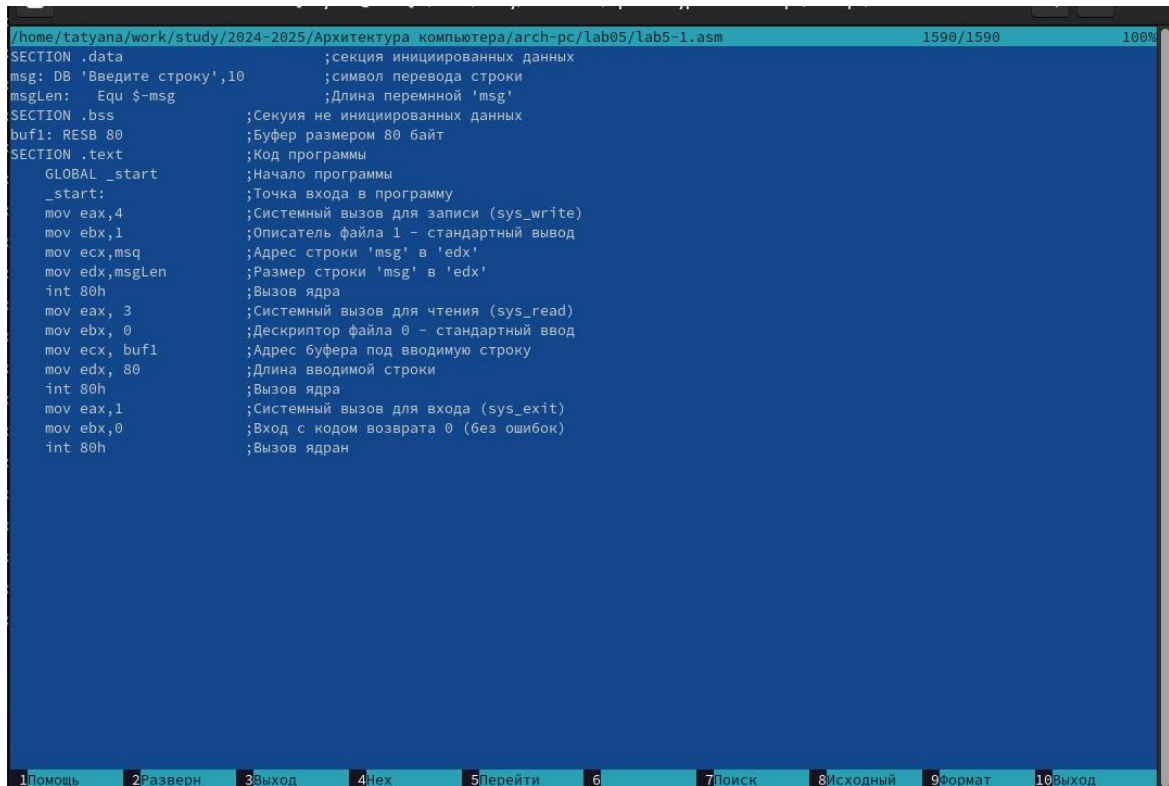


Рис 6

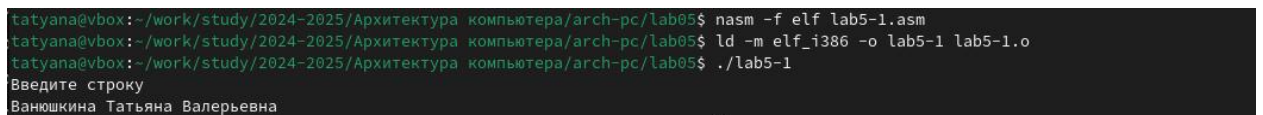
С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра:



```
SECTION .data ;секция иницированных данных
msg: DB 'Введите строку',10 ;символ перевода строки
msgLen: Equ $-msg ;Длина переменной 'msg'
SECTION .bss ;Секция не иницированных данных
buf1: RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,4 ;Системный вызов для записи (sys_write)
mov ebx,1 ;Описатель файла 1 - стандартный вывод
mov ecx,msg ;Адрес строки 'msg' в 'edx'
mov edx,msgLen ;Размер строки 'msg' в 'edx'
int 80h ;Вызов ядра
mov eax,3 ;Системный вызов для чтения (sys_read)
mov ebx,0 ;Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ;Адрес буфера под вводимую строку
mov edx,80 ;Длина вводимой строки
int 80h ;Вызов ядра
mov eax,1 ;Системный вызов для выхода (sys_exit)
mov ebx,0 ;Выход с кодом возврата 0 (без ошибок)
int 80h ;Вызов ядра
```

Рис 7

Транслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл:

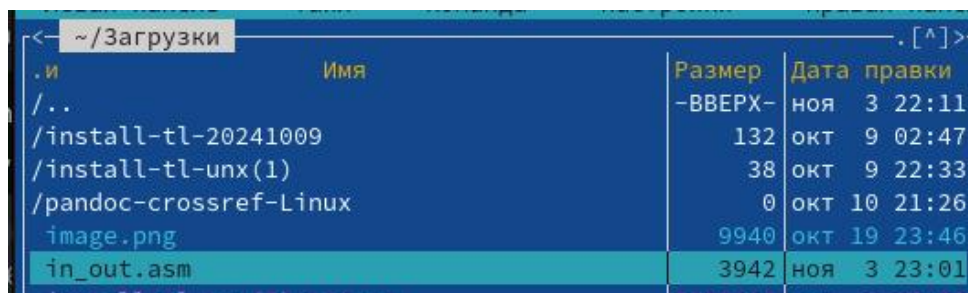


```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-1
Введите строку
Ванюшкина Татьяна Валерьевна
```

Рис 8

## 3.2 Подключение внешнего файла in\_out.asm

Скачиваю файл in\_out.asm со страницы курса в ТУИС:



.и	Имя	Размер	Дата правки
/..		-ВВЕРХ-	ноя 3 22:11
/install-tl-20241009		132	окт 9 02:47
/install-tl-unx(1)		38	окт 9 22:33
/pandoc-crossref-Linux		0	окт 10 21:26
image.png		9940	окт 19 23:46
in_out.asm		3942	ноя 3 23:01
install-tl-unx(1).tar.gz		5756185	ноя 0 22:33



Рис 9

В одной из панелей mc открываю каталогс файлом lab5-1.asm. В другой панели каталогсо скаченным файлом in\_out.asm. Копирую файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5:

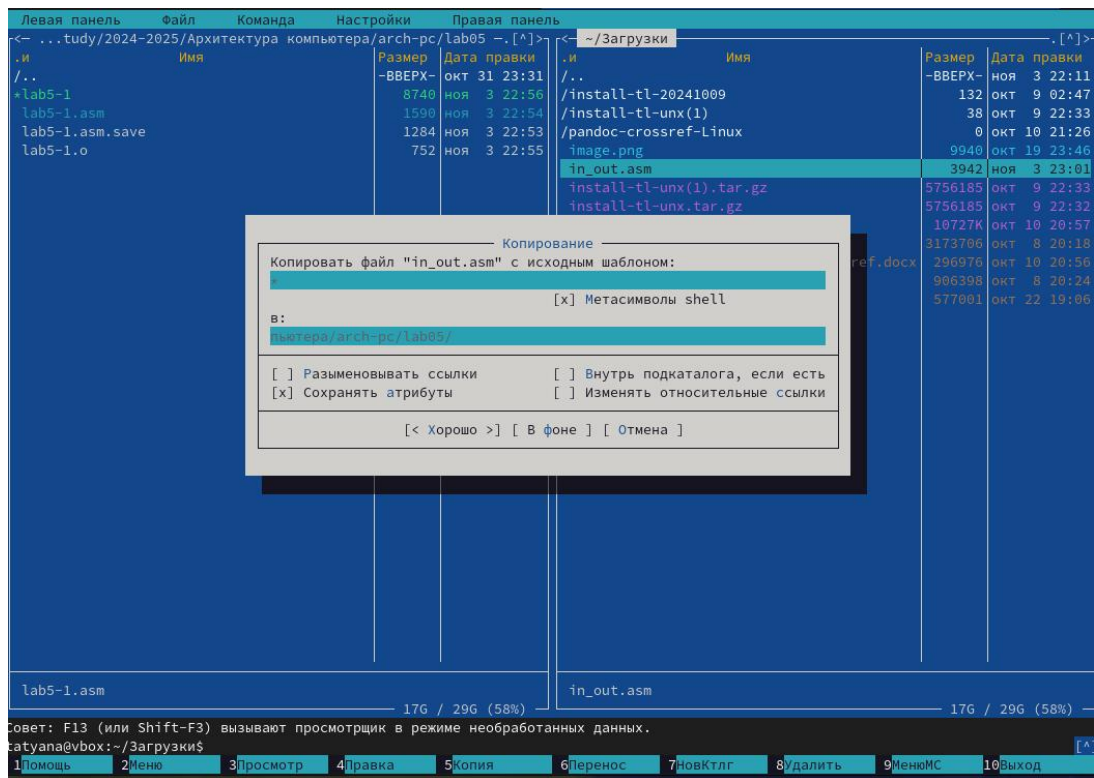


Рис 10

Левая панель	Файл	Команда	Настройки	Правая панель
<pre>&lt;- ...tudy/2024-2025/Архитектура компьютера/arch-pc/lab05 -.[^]&gt;</pre>				
.и	Имя	Размер	Дата правки	
/..		-ВВЕРХ-	окт 31 23:31	
in_out.asm		3942	ноя 3 23:01	
*lab5-1		8740	ноя 3 22:56	
lab5-1.asm		1590	ноя 3 22:54	
lab5-1.asm.save		1284	ноя 3 22:53	
lab5-1.o		752	ноя 3 22:55	

Рис 11

С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm именем lab5-2.asm:

Копирование

Копировать файл "lab5-1" с исходным шаблоном:

[^]

[x] Метасимволы shell

в:

пьютера/arch-pc/lab05/

[^]

[ ] Разыменовывать ссылки

[x] Сохранять атрибуты

[ ] Внутрь подкаталога, если есть

[ ] Изменять относительные ссылки

[< Хорошо >]

[ В фоне ]

[ Отмена ]

Рис 12

Левая панель	Файл	Команда	Настройки	Правая панель
< ...tudy/2024-2025/Архитектура компьютера/arch-pc/lab05 -.[^]>				
.и	Имя	Размер	Дата правки	
/..		-ВВЕРХ-	окт 31 23:31	
/lab5-2.asm		0	ноя 4 12:32	
in_out.asm		3942	ноя 3 23:01	
*lab5-1		8740	ноя 3 22:56	
lab5-1.asm		1590	ноя 3 22:54	
lab5-1.asm.save		1284	ноя 3 22:53	
lab5-1.o		752	ноя 3 22:55	

Рис 13

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2:

```
lab05-2.asm [-M--] 62 L: [ 1+13 14/ 14] *(1207/1207b) <EOF>
#include 'in_out.asm' ;подключение внешнего файла
SECTION .data ;секция иницированных данных
msg: DB 'Введите строку:' ,0h ;сообщение
SECTION .bss ;секция не иницированных данных
buf1: RESB 80 ;буфер размером 80 байт
SECTION .text ;код программы
GLOBAL _start ;начало программы
_start ;точка входа в программу
mov eax, msg ;запись адреса выводимого сообщения в 'EAX'
call sprintf ;вызов подпрограммы печати сообщения
mov ecx, buf1 ;запись адреса переменной в 'EAX'
mov edx, 80 ;запись длины выводимого сообщения в 'EBX'
call sread ;вызов подпрограммы ввода сообщения
call quit ;вызов подпрограммы завершения
```

Рис 14

Компеляция и исполнение файла:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-2.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-2 lab05-2.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-2
Введите строку :
Ванюшкина Татьяна Валерьевна
```

Рис 15

В файле lab5-2.asm заменяю подпрограмму:

```
lab05-2.asm [-M--] 62 L: [ 1+13 14/ 14] *(1136/1136b) <EOF>
#include 'in_out.asm' ;подключение внешнего файла
SECTION .data ;секция иницированных данных
msg: DB 'Введите строку : ' ,0h ;сообщение
SECTION .bss ;секция не иницированных данных
buf1: RESB 80 ;буфер размером 80 байт
SECTION .text ;код программы
GLOBAL _start ;начало программы
_start ;точка входа в программу
mov eax, msg ;запись адреса выводимого сообщения в 'EAX'
call sprintf ;вызов подпрограммы печати сообщения
mov ecx, buf1 ;запись адреса переменной в 'EAX'
mov edx, 80 ;запись длины выводимого сообщения в 'EBX'
call sread ;вызов подпрограммы ввода сообщения
call quit ;вызов подпрограммы завершения
```

Рис 16

Компеляция и исполнение файла:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-2.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-2 lab05-2.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-2
Введите строку : Ванюшкина Татьяна Валерьевна
```

Рис 17

### 3.3 Задание для самостоятельной работы

Создаю копию файла lab5-1.asm:

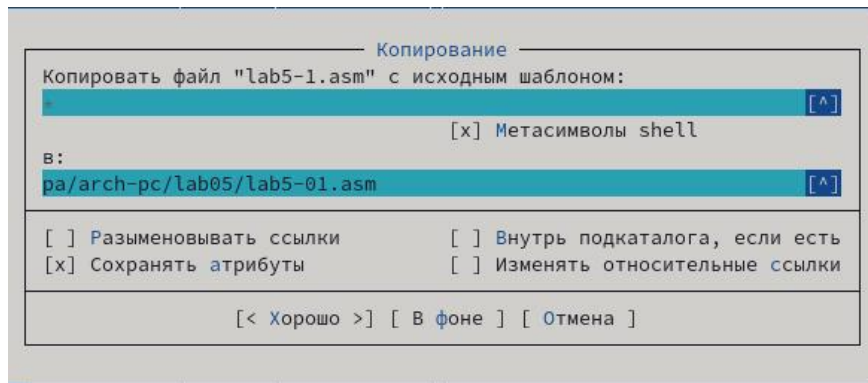


Рис 18

Вношу изменения в программу ,так чтобы она работала по следующему алгоритму:

- вывести приглашениетипа“Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран

```
lab5-01.asm [----] 32 L:[ 1+20 21/ 21] *(1464/1464b) <EOF>
SECTION .data ;секция иницированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
msgLen: EQU $-msg ;длина переменной 'msg'
SECTION .bss ;секция не иницированных данных
buf1: RESB ;буфер размеров 80 байт
SECTION .text ;код программы
GLOBAL _start ;начало программы
_start: ;точка входа в программу
mov eax,4 ;системный вызов для записи (sys_write)
mov ebx,1 ;описатель файла 1 - стандартный вывод
mov ecx,msg ;адрес строки в 'msg' в 'ecx'
mov edx,msgLen ;размер строки 'msg' в 'edx'
int 80h ;вызов ядра
mov eax,3 ;системный вызов для чтения (sys_read)
mov ebx,0 ;дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ;адрес буфера под вводимую строку
mov edx,80 ;длина вводимой строки
int 80h ;вызов ядра
mov eax,1 ;системный вызов для выхода (sys_exit)
mov ebx,0 ;выход с кодом возврата 0 (без ошибок)
int 80h ;вызов ядра
```

Рис 19

Осуществляю проверку с помощью клавиши F3:

```
/home/tatyana/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05/lab5-01.asm
SECTION .data                ;секция инициированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
msgLen: EQU $-msg            ;длина переменной 'msg'
SECTION .bss                 ;секция не инициированных данных
buf1: RESB                   ;буфер размеров 80 байт
SECTION .text                ;код программы
GLOBAL _start               ;начало программы
_start:                     ;точка входа в программу
mov eax,4                   ;системный вызов для записи (sys_write)
mov ebx,1                   ;описатель файла 1 - стандартный вывод
mov ecx,msg                 ;адрес строки в 'msg' в 'ecx'
mov edx,msgLen              ;размер строки 'msg' в 'edx'
int 80h                    ;вызов ядра
mov eax,3                   ;системный вызов для чтения (sys_read)
mov ebx,0                   ;дискриптор файла 0 - стандартный ввод
mov ecx,buf1                ;адрес буфера под вводимую строку
mov edx,80                  ;длина вводимой строки
int 80h                    ;вызов ядра
mov eax,1                   ;системный вызов для выхода (sys_exit)
mov ebx,0                   ;выход с кодом возврата 0 (без ошибок)
int 80h                    ;вызов ядра
```

Рис 20

Получаю исполняемый файл. Проверяю его работу. На приглашение ввести строку ввожу свою фамилию:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-01.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-01 lab5-01.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-01
Введите строку:
Ванюшкина Татьяна
```

Рис 21

Создаю копию файла lab5-2.asm:

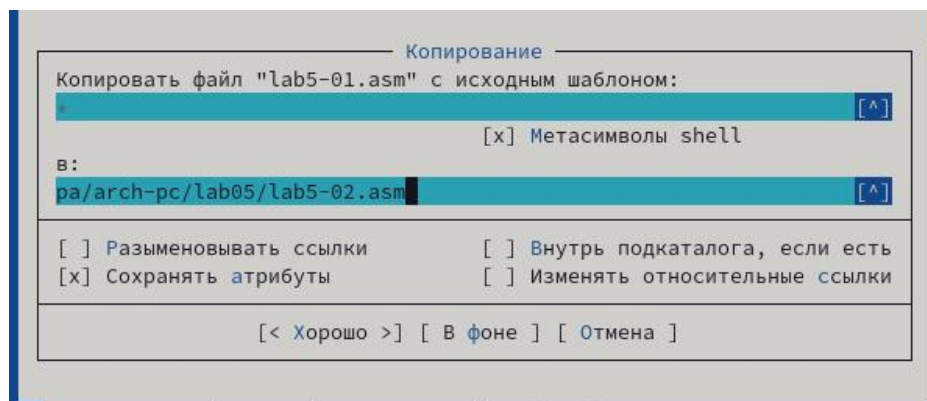


Рис 22



Исправляю текст программы с использованием подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран

```
lab5-02.asm      [----] 60 L: [ 1+13 14/ 14] *(1167/1167b) <EOF>
#include 'in_out.asm'      ;подключение внешнего файла
SECTION .data              ;секция инициированных данных
msg: DB 'Введите строку: ',0h ;сообщение
SECTION .bss               ;секция не инициированных данных
buf1: RESB 80              ;буфер размером 80 байт
SECTION .text              ;код программы
GLOBAL _start              ;начало программы
_start:                   ;точка входа в программу
mov eax,msg                ;запись адреса выводимого сообщения в 'EAX'
call sprint                ;вызов подпрограммы печати сообщения
mov ecx,buf1               ;запись адреса переменной в 'EAX'
mov edx,80                 ;запись длины вводимого сообщения в 'EBX'
call sread                 ;вызов подпрограммы ввода сообщения
call quit                  ;вызов подпрограммы завершения
```

Рис 23

Осуществляю проверку с помощью клавиши F3:

```
/home/tatyana/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05/lab5-02.asm 1167/1167
#include 'in_out.asm'      ;подключение внешнего файла
SECTION .data              ;секция инициированных данных
msg: DB 'Введите строку: ',0h ;сообщение
SECTION .bss               ;секция не инициированных данных
buf1: RESB 80              ;буфер размером 80 байт
SECTION .text              ;код программы
GLOBAL _start              ;начало программы
_start:                   ;точка входа в программу
mov eax,msg                ;запись адреса выводимого сообщения в 'EAX'
call sprint                ;вызов подпрограммы печати сообщения
mov ecx,buf1               ;запись адреса переменной в 'EAX'
mov edx,80                 ;запись длины вводимого сообщения в 'EBX'
call sread                 ;вызов подпрограммы ввода сообщения
call quit                  ;вызов подпрограммы завершения
```

Рис 24

Создаю исполняемый файл и проверяю его работу:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-02.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-02 lab5-02.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab5-02
Введите строку: Ванюшкина Татьяна Валерьевна
```

Рис 25

#### **4. Выводы**

В ходе выполнения лабораторной работы я приобрела практические навыки работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.



## **5. Список литературы**

Курс: Архитектура компьютеров и операционные системы.

Раздел "Архитектура компьютеров" (02.03.00, УГСН) ([rudn.ru](http://rudn.ru))