

# **Отчёт по лабораторной работе №5**

**Основы работы с Midnight commander. Структура программы на языке  
NASM. Системные вызовы**

Ведьмина Александра Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>15</b>
<b>6</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

4.1	Открытие mc . . . . .	8
4.2	Переход в каталог ~/work/arch-pc . . . . .	9
4.3	Создание папки lab05 . . . . .	9
4.4	Создание файла lab5-1.asm . . . . .	10
4.5	Открытие mc . . . . .	11
4.6	Создание исполняемого файла lab5-1 . . . . .	11
4.7	Запуск lab5-1 . . . . .	12
4.8	Запуск lab5-1 . . . . .	12
4.9	Создание копии lab5-1 . . . . .	13
4.10	Изменение программы в lab5-2 . . . . .	13
4.11	Запуск lab5-2 . . . . .	14
4.12	Запуск lab5-2-1 . . . . .	14
5.1	Внесение изменений в файл lab5-1-cory.asm . . . . .	15
5.2	Запуск lab5-1-cory . . . . .	16
5.3	Внесение изменений в файл lab5-2-cory.asm . . . . .	16
5.4	Запуск lab5-2-cory . . . . .	16

## **Список таблиц**

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Изучить основы работы с тс.
2. Познакомиться со структурой программы вывода сообщений на экран и ввода со строки на языке NASM.
3. Подключить файл in\_out.asm.
4. Выполнить задания для самостоятельной работы.

### 3 Теоретическое введение

Midnight Commander — это файловый менеджер, с помощью которого можно просматривать структуру каталогов и выполнять основные операции по управлению файловой системой. Чтобы начать использовать Midnight Commander, достаточно ввести в командной строке `mc`. Программа на языке ассемблера NASM обычно состоит из трёх секций: секция кода, секция инициированных данных и секция неинициализированных данных. 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. Директивы применяются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву `DB`. Для объявления неинициированных данных в секции `.bss` используются директивы `resb`, `resw`, `resd` и другие. В качестве операндов могут выступать регистры, ячейки памяти и значения.

Простейший способ вывести строку на экран — использовать системный вызов `write`. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции `int` необходимо поместить значение 4 в регистр `eax`. Первым аргументом `write`, помещаемым в регистр `ebx`, задаётся дескриптор файла. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов `read`. Его аргументы — такие же, как у вызова `write`, только для «чтения» с клавиатуры используется файловый дескриптор 0.

## 4 Выполнение лабораторной работы

Открываю Midnight Commander.

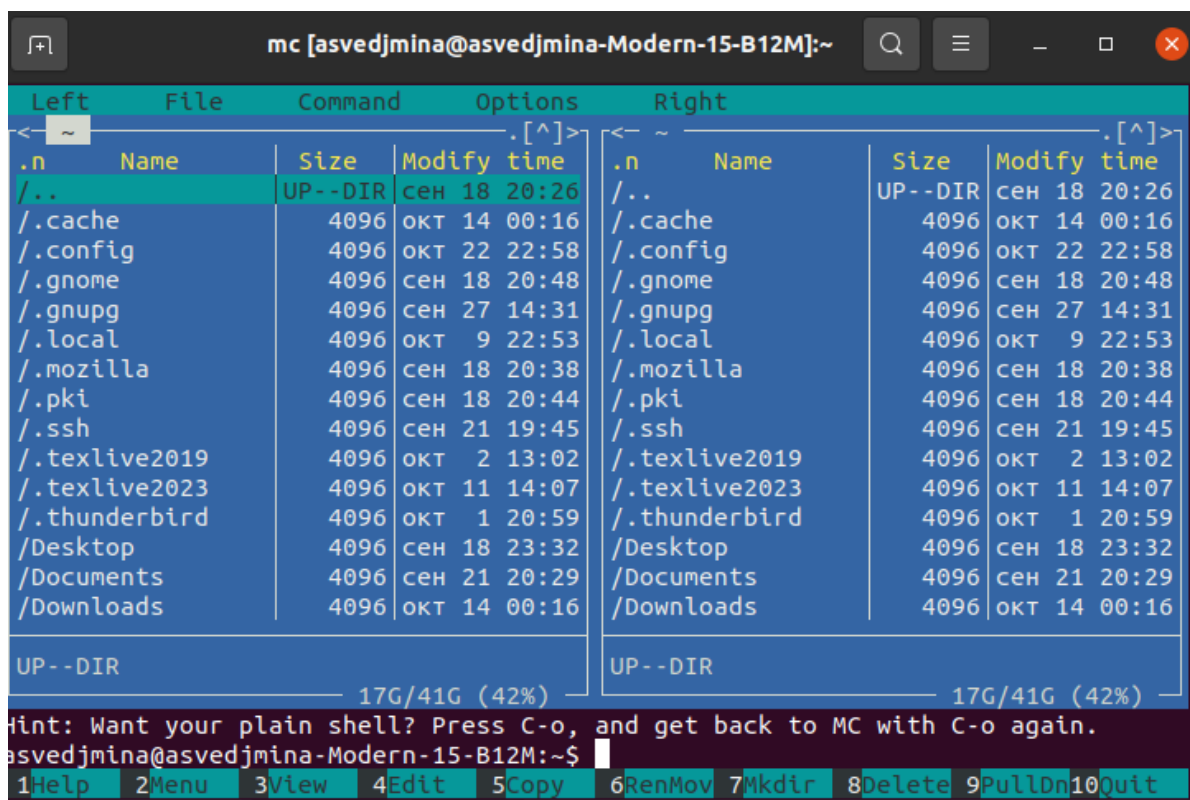


Рис. 4.1: Открытие mc

Перехожу в каталог, созданный при выполнении лабораторной №4.





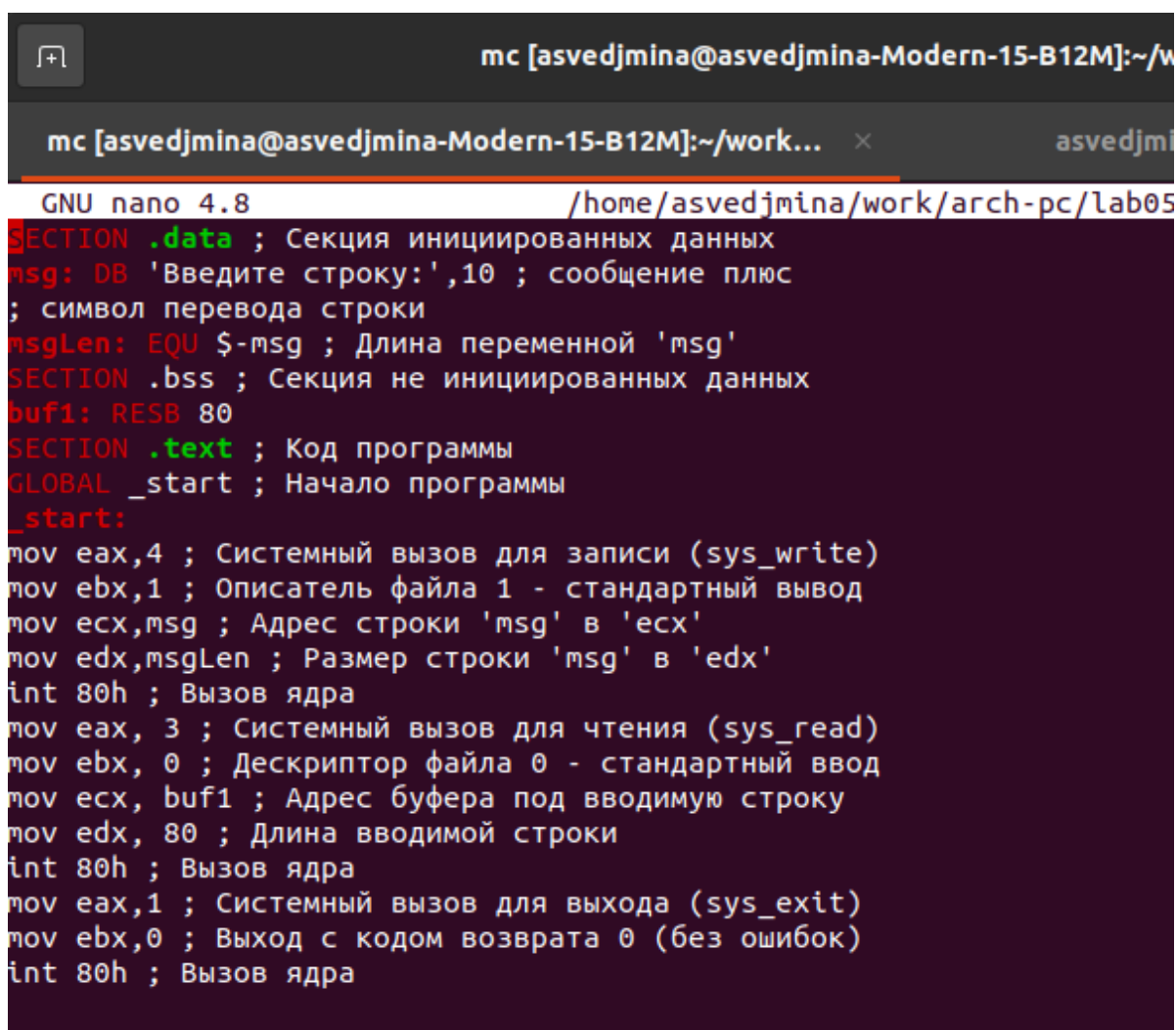
```
< ~ /work/arch-pc/lab05 .[^]>
.n      Name      Size      Modify time
/..      UP--DIR  OKT 25 13:53
lab5-1.asm      0      OKT 25 13:54

UP--DIR

17G/41G (42%)
Hint: You can disable all requests for confirmation in Options/Confirmation.
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$
```

Рис. 4.4: Создание файла lab5-1.asm

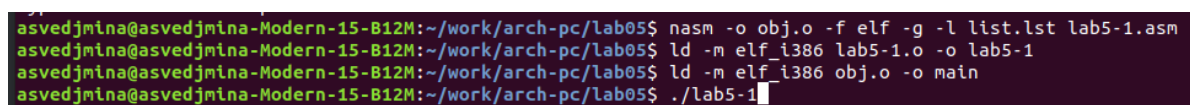
Открываю этот файл в редакторе nano и ввожу предлагаемый текст программы, после чего закрываю файл.



```
mc [asvedjmina@asvedjmina-Modern-15-B12M]:~/w
mc [asvedjmina@asvedjmina-Modern-15-B12M]:~/work... x asvedjmi
GNU nano 4.8 /home/asvedjmina/work/arch-pc/lab05
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start:
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки '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. 4.5: Открытие mc

Транслирую lab5-1.asm в объектный файл, выполняю его компоновку и запускаю исполняемый файл. На запрос ввожу своё ФИО (Ведьмина Александра Сергеевна).



```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ nasm -o obj.o -f elf -g -l list.lst lab5-1.asm
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1.o -o lab5-1
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 obj.o -o main
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-1
```

Рис. 4.6: Создание исполняемого файла lab5-1

```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Ведьмина Александра Сергеевна
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$
```

Рис. 4.7: Запуск lab5-1

Далее скачиваю файл in\_out.asm с ТУИСа и помещаю его в ~/work/arch-pc/lab05.

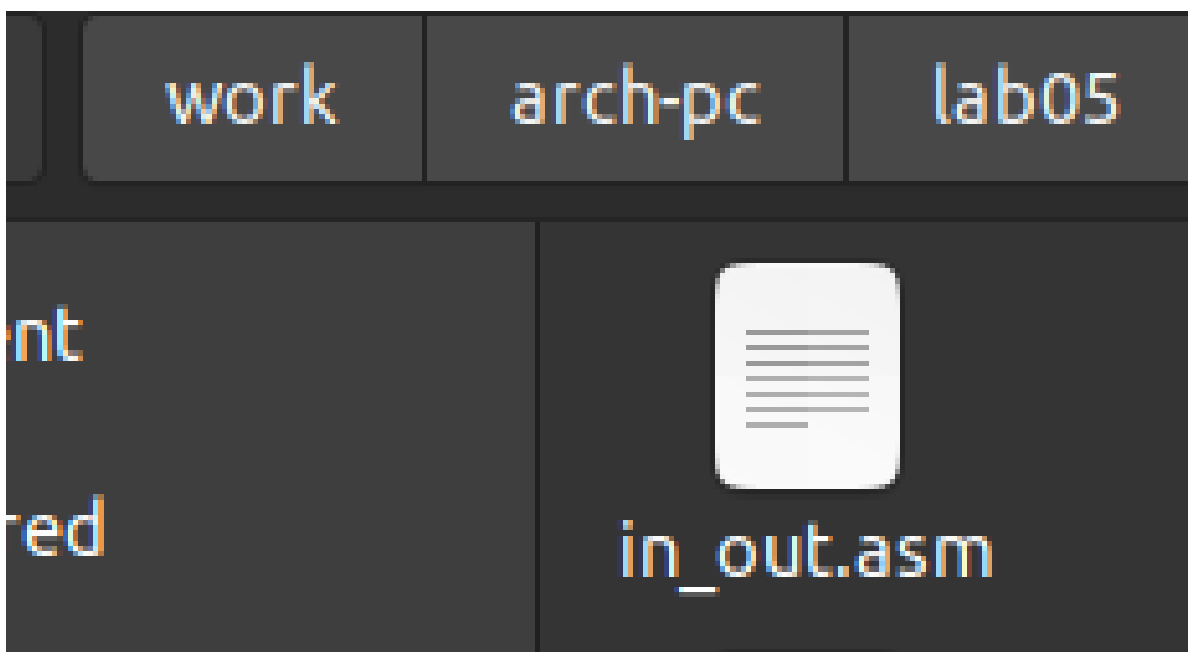


Рис. 4.8: Запуск lab5-1

Делаю копию файла lab5-1.asm и называю её lab5-2.asm.

Left	File	Command	Options	Right
<-	~/work/arch-pc/lab05			<- ~/work/arch-pc/lab05
.n	Name	Size	Modify time	.n Name Size Modify time
/..		UP--DIR	окт 25 13:57	/.. UP--DIR окт 25 13:57
in_out.asm		3942	окт 25 14:26	in_out.asm 3942 окт 25 14:26
*lab5-1		8792	окт 25 14:22	*lab5-1 8792 окт 25 14:22
lab5-1.asm		1227	окт 25 14:13	lab5-1.asm 1227 окт 25 14:13
lab5-1.o		752	окт 25 14:20	lab5-1.o 752 окт 25 14:20
lab5-2.asm		1227	окт 25 14:13	lab5-2.asm 1227 окт 25 14:13
list.lst		2255	окт 25 14:21	list.lst 2255 окт 25 14:21
*main		9124	окт 25 14:23	*main 9124 окт 25 14:23
obj.o		1216	окт 25 14:21	obj.o 1216 окт 25 14:21
lab5-2.asm				UP--DIR
17G/41G (42%)				17G/41G (42%)

Рис. 4.9: Создание копии lab5-1

Используя подпрограммы из in\_out.asm, вношу изменения в lab5-2.asm.

```

mc [asvedjmina@asvedjmina-Modern-15-B12M]:~/work/arch-...
GNU nano 4.8 /home/asvedjmina/work/arch-pc/lab05/lab5-2.asm
%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 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. 4.10: Изменение программы в lab5-2

Создаю исполняемый файл и запускаю его.

```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Ведьмина Александра
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$
```

Рис. 4.11: Запуск lab5-2

В файле lab5-2.asm заменяю `sprintLF` на `sprint`, создаю исполняемый файл lab5-2-1 и запускаю его, после чего переименовываю его в lab5-2.

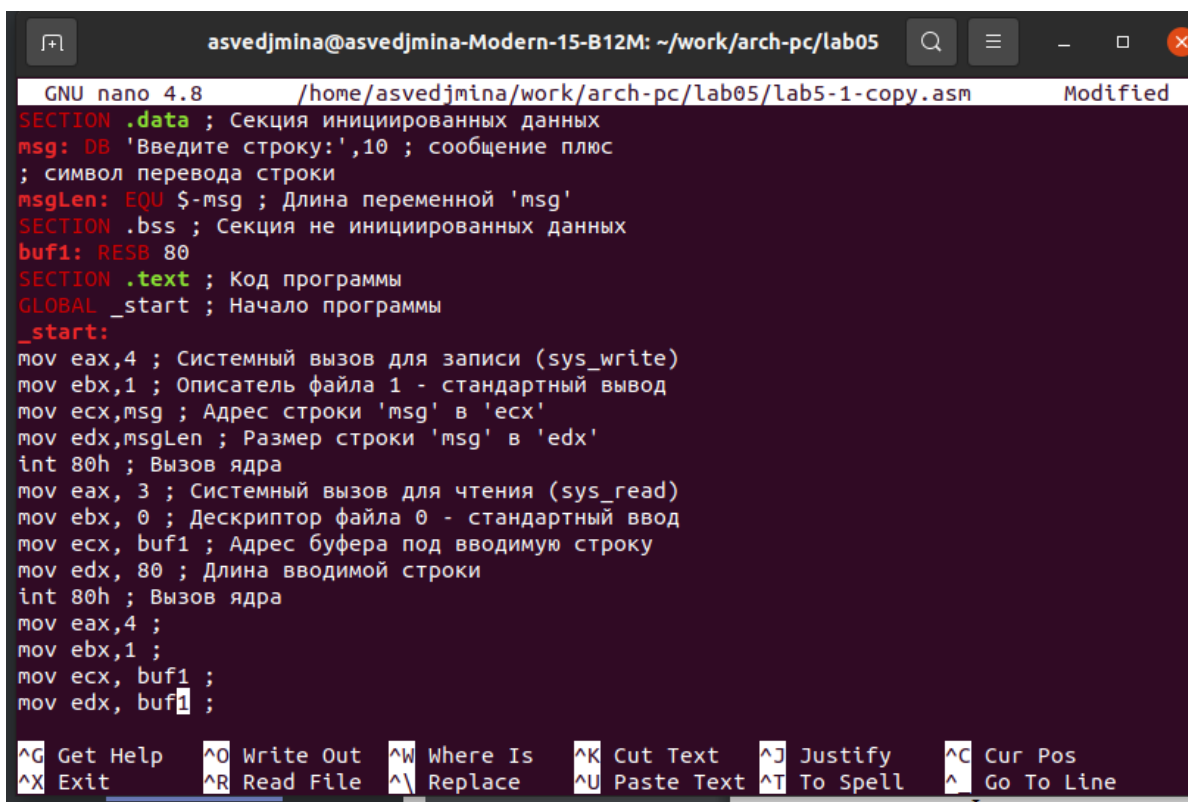
```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2.o
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку:
Ведьмина Александра Сергеевна
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$
```

Рис. 4.12: Запуск lab5-2-1

Разница между программами со `sprintLF` и `sprint` состоит в том, что просит ввести данные на той же строке, на которой выводится их запрос, в то время как вторая предлагает сделать это на новой строке.

## 5 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-copy.asm и вношу изменения в программу так, чтобы она выводила введенную строку на экран.



```
GNU nano 4.8 /home/asvedjmina/work/arch-pc/lab05/lab5-1-copy.asm Modified
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start:
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки '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,4 ;
mov ebx,1 ;
mov ecx, buf1 ;
mov edx, buf1 ;

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line
```

Рис. 5.1: Внесение изменений в файл lab5-1-copy.asm

Получаю исполняемый файл и запускаю его, вводя на запрос своё ФИО.

```

asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ nasm -f elf lab5-1-copy.asm
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-copy
lab5-1-copy.o
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-1-copy
Введите строку:
Ведьмина Александра Сергеевна
Ведьмина Александра Сергеевна
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$

```

Рис. 5.2: Запуск lab5-1-copy

Создаю копию файла lab-2.asm с именем lab5-2-copy.asm, изменяю программу так, чтобы она тоже выводила на экран введённую строку.

```

GNU nano 4.8 /home/asvedjmina/work/arch-pc/lab05/lab5-2-copy.asm
%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 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
mov eax,4 ;
mov ebx,1 ;
mov ecx,buf1 ;
int 80h ;
call quit ; вызов подпрограммы завершения

```

Рис. 5.3: Внесение изменений в файл lab5-2-copy.asm

После этого создаю исполняемый файл и запускаю его.

```

asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ nasm -f elf lab5-2-copy.asm
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-copy
lab5-2-copy.o
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$ ./lab5-2-copy
Введите строку: Ведьмина Александра Сергеевна
Ведьмина Александра Сергеевна
asvedjmina@asvedjmina-Modern-15-B12M:~/work/arch-pc/lab05$

```

Рис. 5.4: Запуск lab5-2-copy



## 6 Выводы

В ходе этой лабораторной работы я научилась в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.