

# Шаблон отчёта по лабораторной работе по 5

## Дисциплина: компьютера архитектур

Дзаки Рафли Зайдан

### Содержание

1	Цель работы .....	1
2	Выполнение лабораторной работы.....	1
3	Задание для самостоятельной работы.....	9
4	Выводы.....	10

### #Список иллюстраций



1	Цель работы	4	2	Выполнение лабораторной работы	5	2.1	Задание 1: .....	1
.....	5	2.2	Задание 2: .....	6	2.3	Задание 3: .....	1	
.....	6	2.4	Задание 4: .....	7	2.5	Задание 5: .....	1	
.....	8	2.6	Задание 6: .....	9	2.7	Задание 7: .....	1	
.....	11	2.8	Задание 8: .....	12	2.9	Задание 9: .....	1	
.....	14	3	Задание для самостоятельной работы	16	3.1	Задание	10: .....	
10: .....	16	3.2	Задание 11: .....	18	4			
Вывод	20							

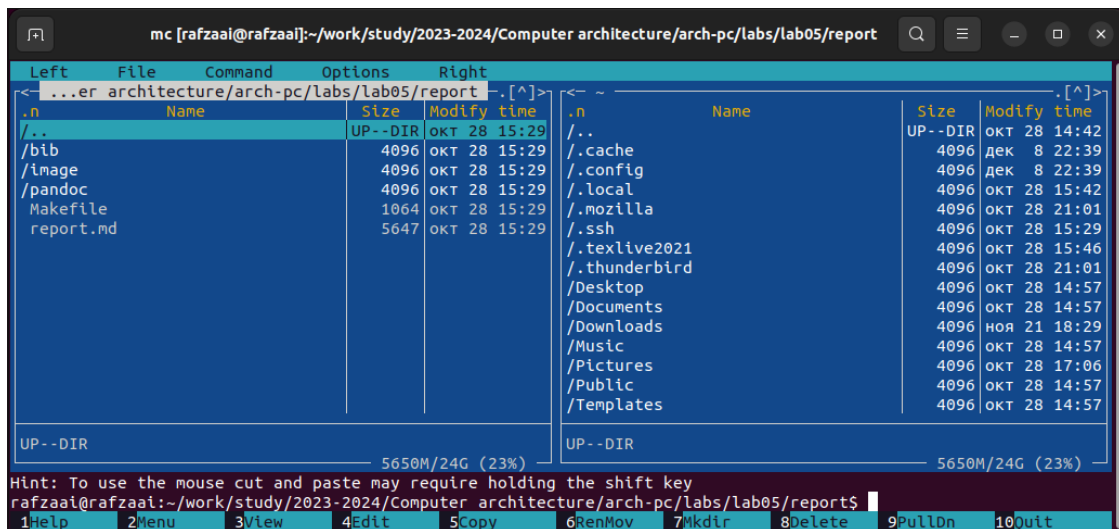
## 1 Цель работы

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

## 2 Выполнение лабораторной работы

### 2.1 Задание 1.

Откройте Midnight Commander. Пользуясь клавишами ,  и Enter перейдите в каталог ~/work/arch-рс. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm.



*Переход в каталог курса и введение команды на создание файла*

Midnight Commander не был предустановлен на используемом мною дистрибутиве, но после установки работал исправно. В столбце справа можно видеть содержание директории lab05/report, в столбце слева - домашней директории. В строчке снизу введена команда для создания файла.

## 2.2 Задание 2.

С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе.

```
rafzaal@rafzaal:~$ mc
rafzaal@rafzaal:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report$ touch lab5-1.asm

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/mcedit
 3. /usr/bin/vim.tiny
 4. /bin/ed

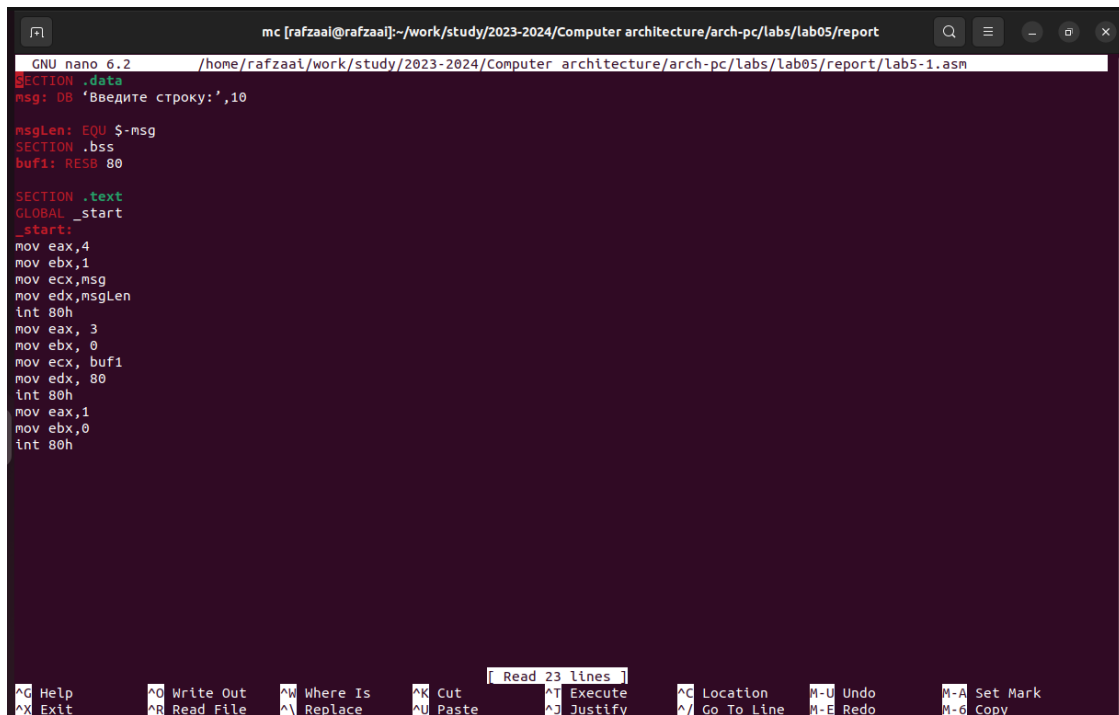
Choose 1-4 [1]:
```

{#fig:001 width=100%

Из предложенных на выбор редакторов я выбрала nano, для чего ввела в строку цифру 1.

## 2.3 Задание 3.

Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл.



```
GNU nano 6.2 /home/rafzaal/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-1.asm
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

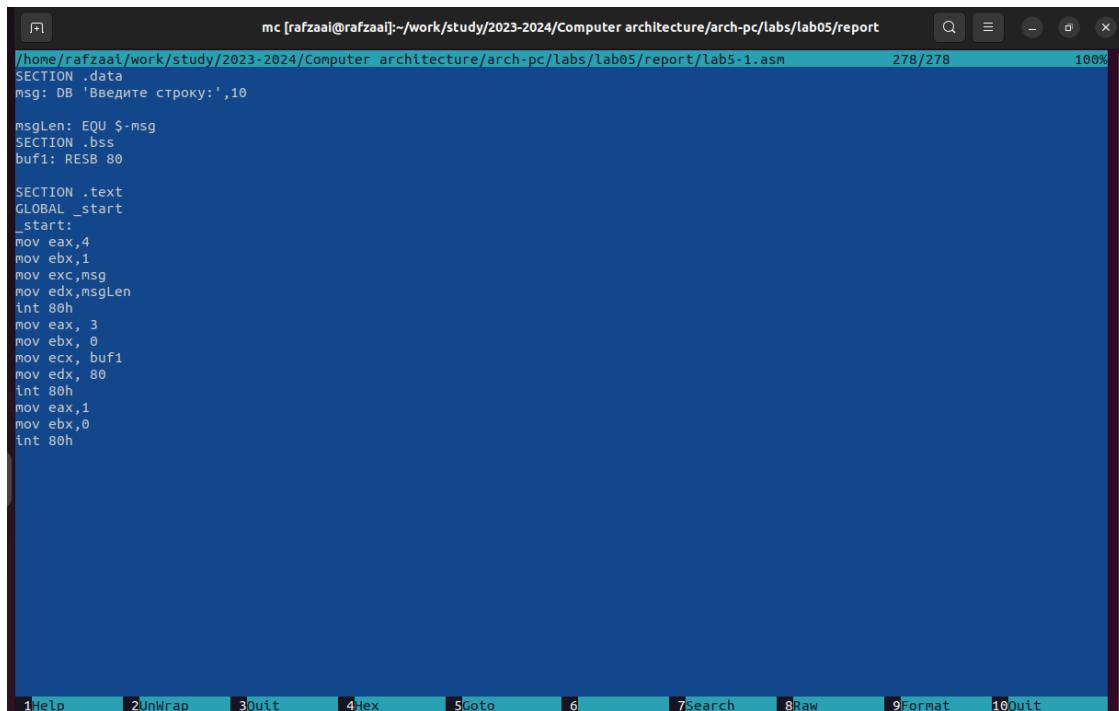
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

*Введение текста программы из листинга*

Текст программы скопирован, комментарии удалены. Изменения сохранены. Работает.

## 2.4 Задание 4.

С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.



```
mc [rafzaai@rafzaai]:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report
/home/rafzaai/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-1.asm 278/278 100%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

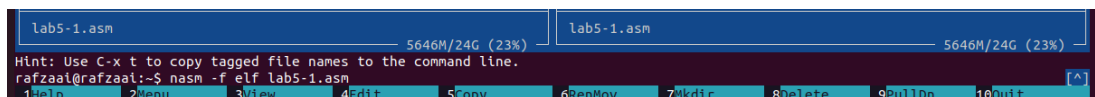
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

*Тот же самый текст, но теперь открытый в Midnight Commander*

Символ-в-символ тот же самый текст, что был введен и до этого. Сохранение успешно.

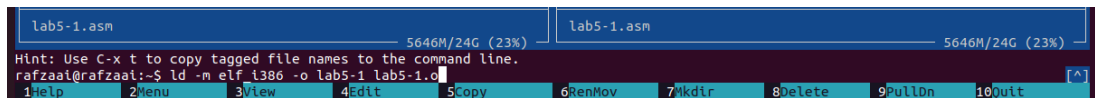
## 2.5 Задание 5.

Оттранслируйте текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.



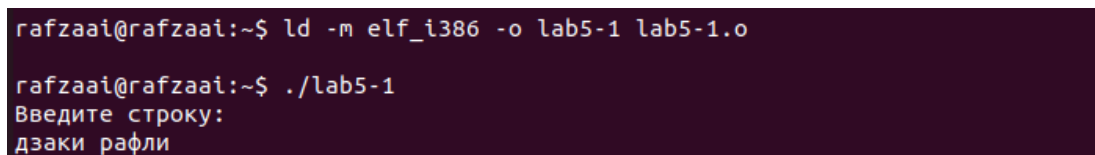
```
lab5-1.asm 5646M/24G (23%) lab5-1.asm 5646M/24G (23%)
Hint: Use C-x t to copy tagged file names to the command line.
rafzaai@rafzaai:~$ nasm -f elf lab5-1.asm
```

*Превращение текста программы в объектный код*



```
lab5-1.asm 5646M/24G (23%) lab5-1.asm 5646M/24G (23%)
Hint: Use C-x t to copy tagged file names to the command line.
rafzaai@rafzaai:~$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

*Создание нового объектного файла и файла листинга программы*



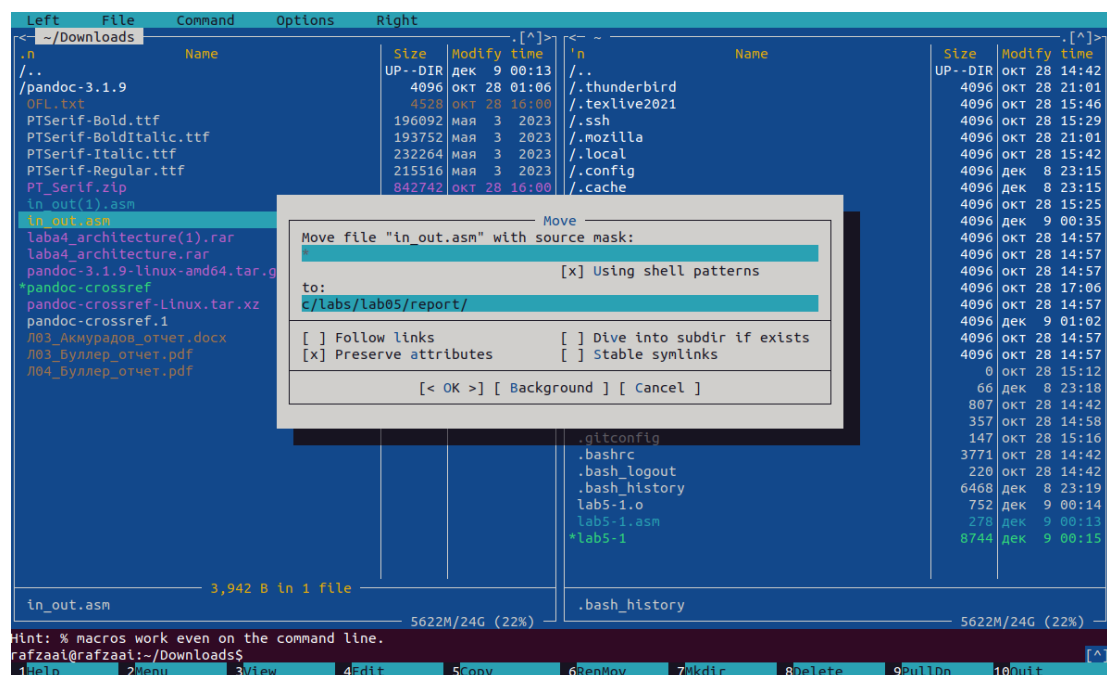
```
rafzaai@rafzaai:~$ ld -m elf_i386 -o lab5-1 lab5-1.o
rafzaai@rafzaai:~$ ./lab5-1
Введите строку:
дзак рафли
```

*Выполнение программы*

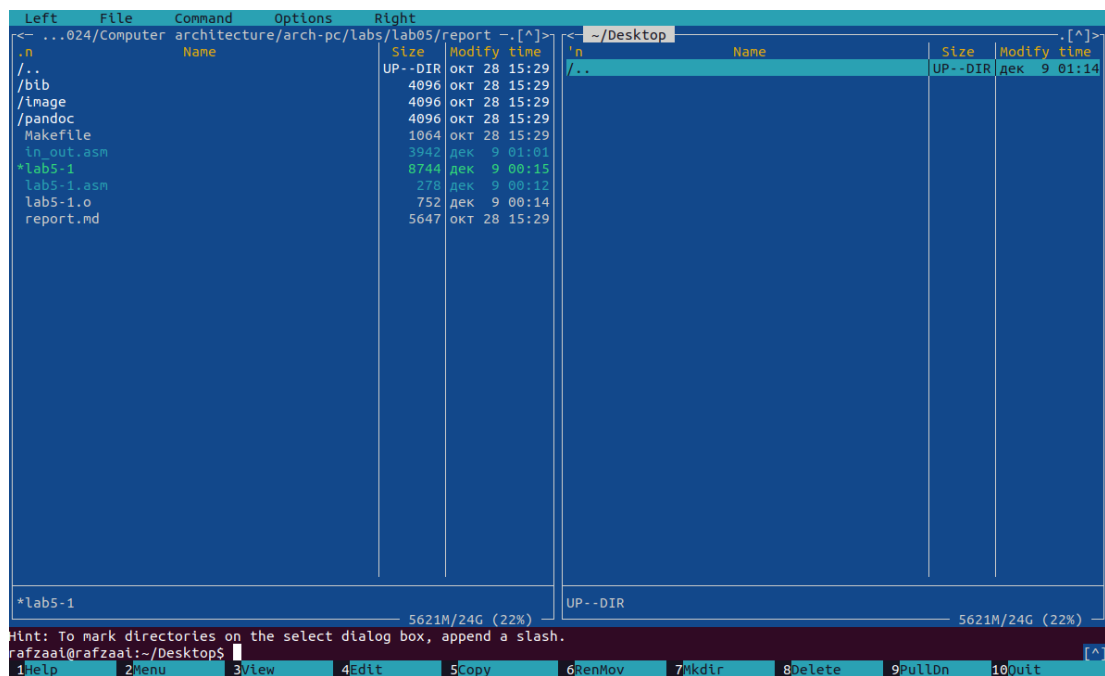
Программа скомпилировалась и выполнялась успешно: попросила меня ввести строку и закрылась сразу же после ввода.

## 2.6 Задание 6.

В одной из панелей mc откройте каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями используйте Tab). Скопируйте файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.



*Копирование файла*

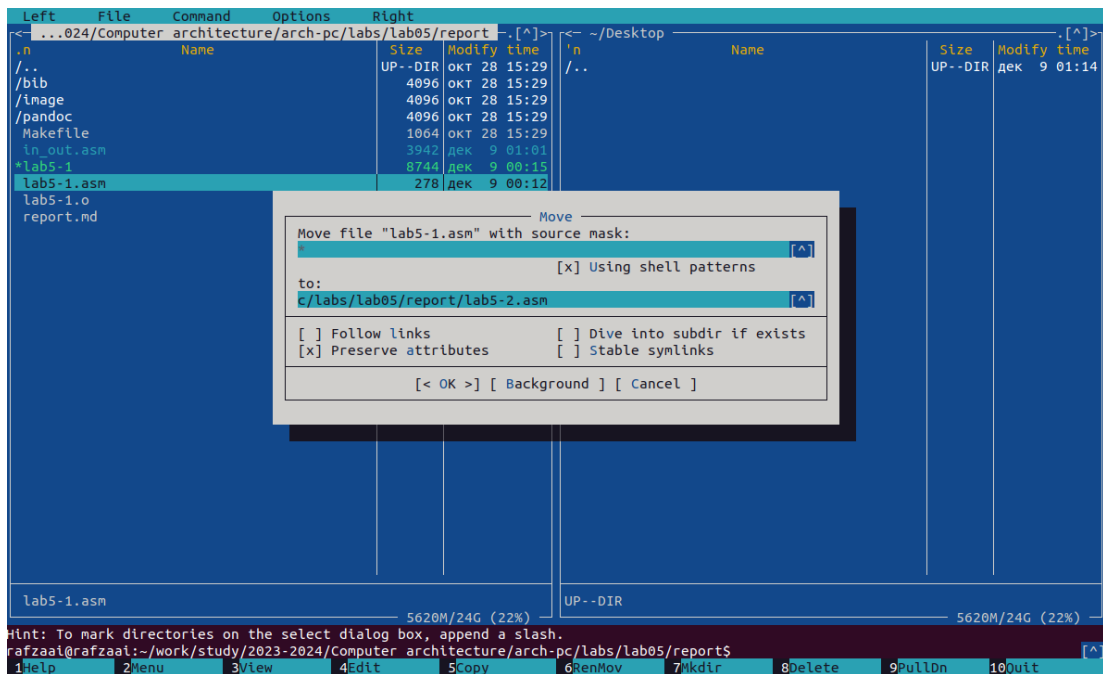


### Файл скопирован

Файл, лежавший до этого на рабочем столе, исчез оттуда после копирования и появился в директории лабораторной работы: копирование выполнено успешно.

## 2.7 Задание 7.

С помощью функциональной клавиши F6 создайте копию файла lab5-1.asm с именем lab5-2.asm. Выделите файл lab5-1.asm, нажмите клавишу F6, введите имя файла lab5-2.asm и нажмите клавишу Enter



Создание копии файла

## 2.8 Задание 8.

Исправьте текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используйте подпрограммы sprintf, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу

```

/home/rafzaal/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-2.asm 228/228 100%
#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 sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit

```

Исправленный код программы

В начало файла указана команда на подключение внешнего файла in\_out.asm, в котором описаны подпрограммы sprintLF, sread и quit. Они вызываются далее в тексте программы, заменяя часть команд mov.

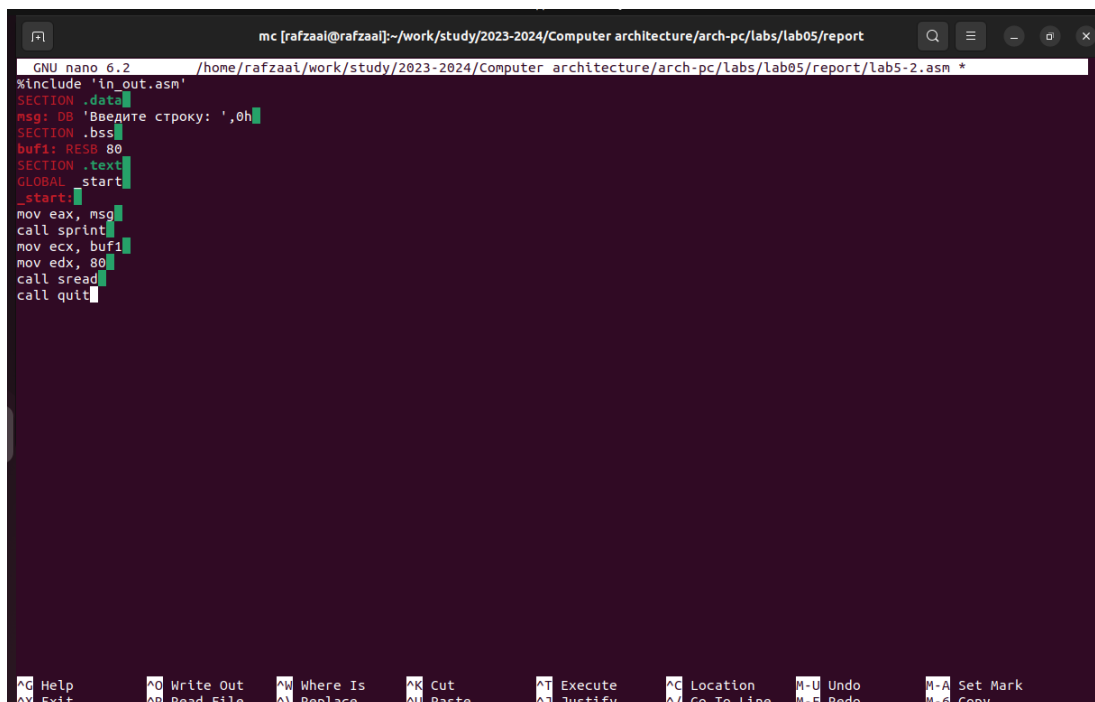
```
rafzaai@rafzaai:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report$ mc
$ nasm -f elf lab5-2.asm
$ ld -m elf_i386 -o lab5-2 lab5-2.o
$ ./lab5-2
Введите строку:
дзак рафли
```

### *Компиляция программы*

Программа скомпилировалась успешно и снова попросила меня ввести строку, после чего завершилась.

## 2.9 Задание 9.

В файле lab5-2.asm замените подпрограмму sprintLF на sprint. Создайте исполняемый файл и проверьте его работу. В чем разница?



```
GNU nano 6.2 /home/rafzaai/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-2.asm *
%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
call quit
```

### *Исправленный код программы*

```
$ ./lab5-2
Введите строку:
дзак рафли
```

### *Компиляция обновленной программы*



Программа скомпилировалась без значительных изменений, но замена `sprintLF` на `sprint` привела к тому, что исчез символ переноса строки при выводе сообщения на экран.

### 3 Задание для самостоятельной работы

#### 3.1 Задание 1.

Создайте копию файла `lab5-1.asm`. Внесите изменения в программу (без использования 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. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

```
GNU nano 6.2 /home/rafzaai/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-11.asm
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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, 80
    int 80h

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

*Исправленный код программы*

```
rafzaai@rafzaai:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report$ ./lab5-1-1
Введите строку:
дзаки рафли
дзаки рафли
```

*Компиляция обновленной программы*

Для того, чтобы после введения строки полученный ввод был показан еще раз, скопируем блок вывода текста за одним исправлением: вместо переменной `msg` теперь нужно вывести полученный ввод (он сохраняется в переменную `buf1`). при компиляции программы можно видеть, что это работает именно так, как и было задумано.

#### 3.2 Задание 2.

Создайте копию файла lab5-2.asm. Исправьте 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. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

```
GNU nano 6.2 /home/rafzaai/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report/lab5-2-2.asm *
#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 sprintf

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprintf
call quit
```

*Исправленный код программы*

```
rafzaai@rafzaai:~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab05/report$ ./lab5-2-2
Введите строку:
дзэки рафли
дзэки рафли
```

*Компиляция обновленной программы*

Для того, чтобы после введения строки полученный ввод был показан еще раз, в этот раз просто еще один раз вызовем команду sprintf. При компиляции программы можно видеть, что это работает именно так, как и было задумано.

## 4 Выводы

При выполнении лабораторной работы были приобретены практические навыки работы в Midnight Commander и освоены инструкции языка ассемблера mov и int.