

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

Абакумов Тимофей Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Основы работы с тс . . . . .	7
3.2	Структура программы на языке ассемблера NASM . . . . .	9
3.3	Подключение внешнего файла . . . . .	10
3.4	Задания для самостоятельной работы . . . . .	13
<b>4</b>	<b>Выводы</b>	<b>17</b>

# Список иллюстраций

3.1	Midnight Commander . . . . .	7
3.2	Перемещение между директориями . . . . .	8
3.3	Создание файла . . . . .	8
3.4	Проверка создания файла . . . . .	8
3.5	Открытие файла . . . . .	9
3.6	Ввод текста . . . . .	9
3.7	Компановка файла . . . . .	10
3.8	Запуск файла . . . . .	10
3.9	Вывод результата . . . . .	10
3.10	Скачивание файла . . . . .	10
3.11	Перемещение файла . . . . .	11
3.12	Копирование файла . . . . .	11
3.13	Проверка успешности копирования . . . . .	12
3.14	Исправление текста . . . . .	12
3.15	Создание и проверка исполняемого файла . . . . .	12
3.16	Создание копии файла . . . . .	13
3.17	Изменения программы . . . . .	13
3.18	Проверка работы файла . . . . .	14
3.19	Копирование файла . . . . .	15
3.20	Проверка работы программы . . . . .	15

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

### **Порядок выполнения лабораторной работы**

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

# 3 Выполнение лабораторной работы

## 3.1 Основы работы с mc

1. Для начала откроем Midnight Commander с помощью команды mc (рис. 3.1).

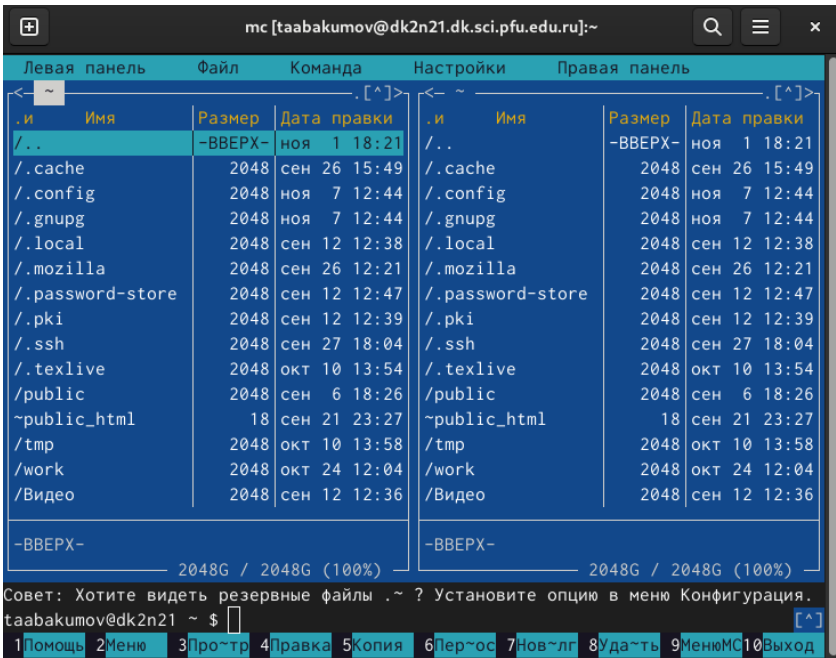


Рис. 3.1: Midnight Commander

2. Пользуясь клавишами ↑ , ↓ и Enter перейдём в каталог ~/work/study/2024-2025/arch-pc/labs/lab05 (рис. 3.2).

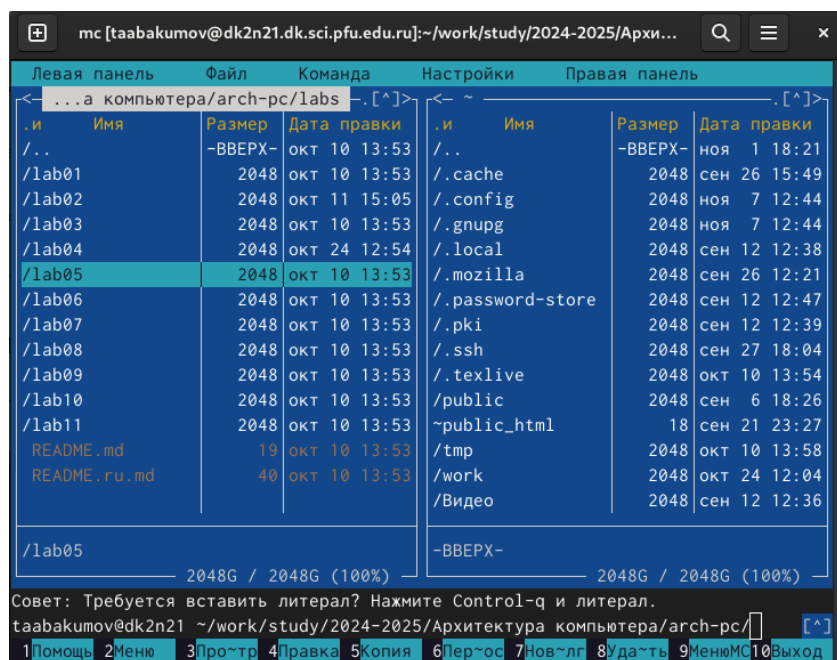


Рис. 3.2: Перемещение между директориями

3. Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm (рис. 3.3).



Рис. 3.3: Создание файла

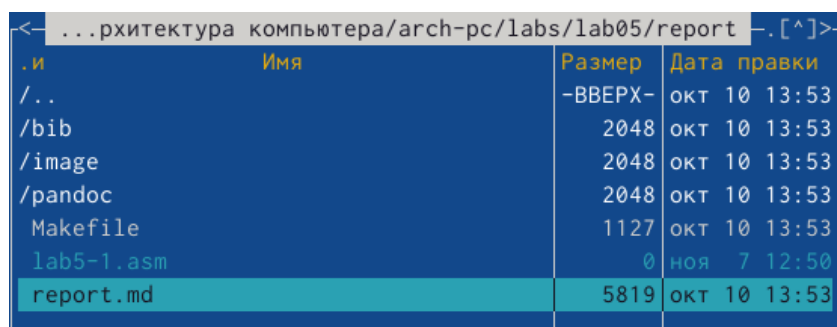


Рис. 3.4: Проверка создания файла



## 3.2 Структура программы на языке ассемблера NASM

4. С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе (рис. 3.5).

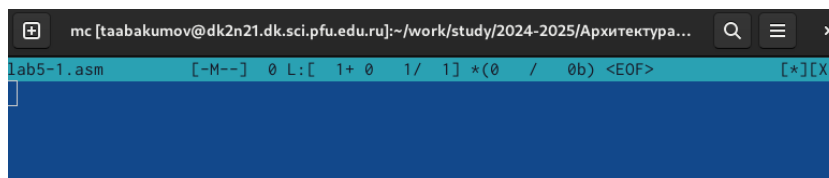


Рис. 3.5: Открытие файла

5. Введём текст программы из листинга 5.1, затем сохраним изменения и закроем файл (рис. 3.6).

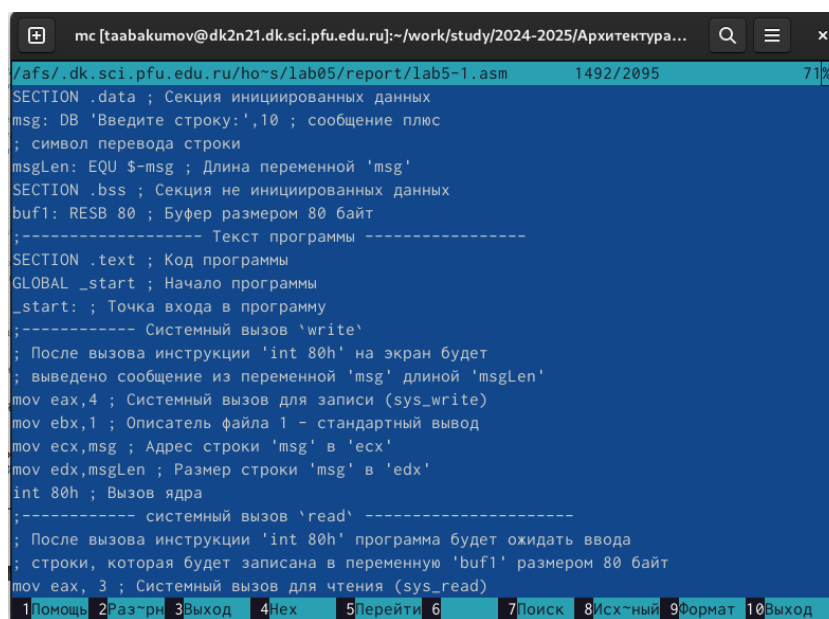


Рис. 3.6: Ввод текста

6. Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введём свою ФИО (рис. 3.7).

```
/report $ ld -m elf_i386 -o lab5-1 lab5-1.o [^]  
вКтлг 8Удалить 9МенюМС 10Выход
```

Рис. 3.7: Компиляция файла

```
/report $ ./lab5-1  
вКтлг 8Удалить 9М
```

Рис. 3.8: Запуск файла

```
taabakumov@dk2n21 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05/report $ ./lab5-1  
Введите строку:  
Абакумов Тимофей Александрович
```

Рис. 3.9: Вывод результата

### 3.3 Подключение внешнего файла

9. Скачаем файл `in_out.asm` со страницы курса в ТУИС (рис. 3.10).

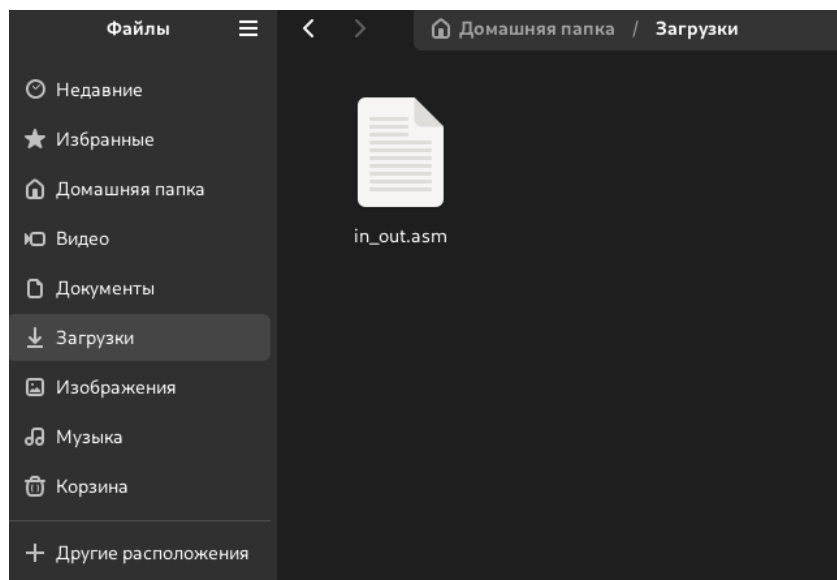


Рис. 3.10: Скачивание файла

10. Подключаемый файл `in_out.asm` должен лежать в том же каталоге, что и файл с программой, в которой он используется, поэтому перемещаем его из папки Загрузки (рис. 3.11).

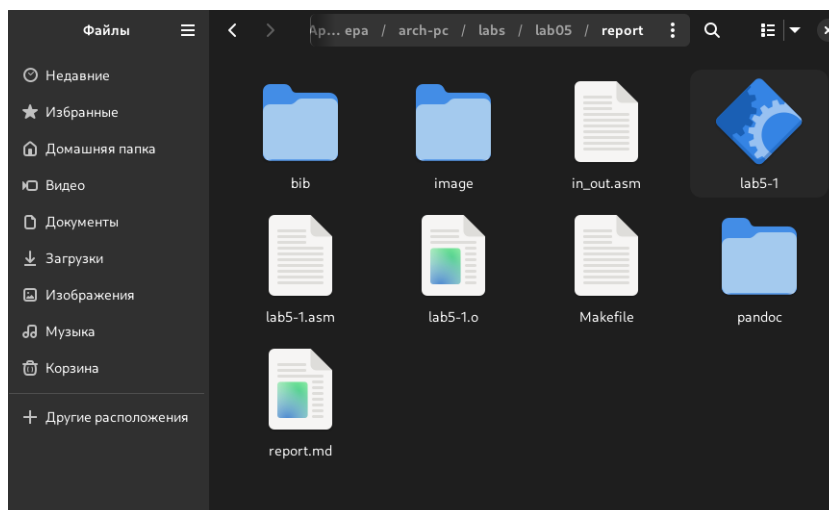


Рис. 3.11: Перемещение файла

11. 1. С помощью функциональной клавиши F5 создадим копию файла `lab5-1.asm` с именем `lab5-2.asm`. Выделим файл `lab5-1.asm`, нажмём клавишу F5, введём имя файла `lab5-2.asm` и нажмём клавишу Enter (рис. 3.12).

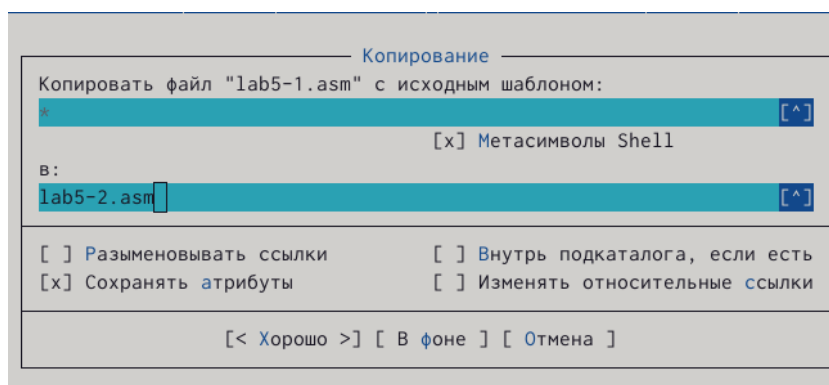


Рис. 3.12: Копирование файла

(рис. 3.13).

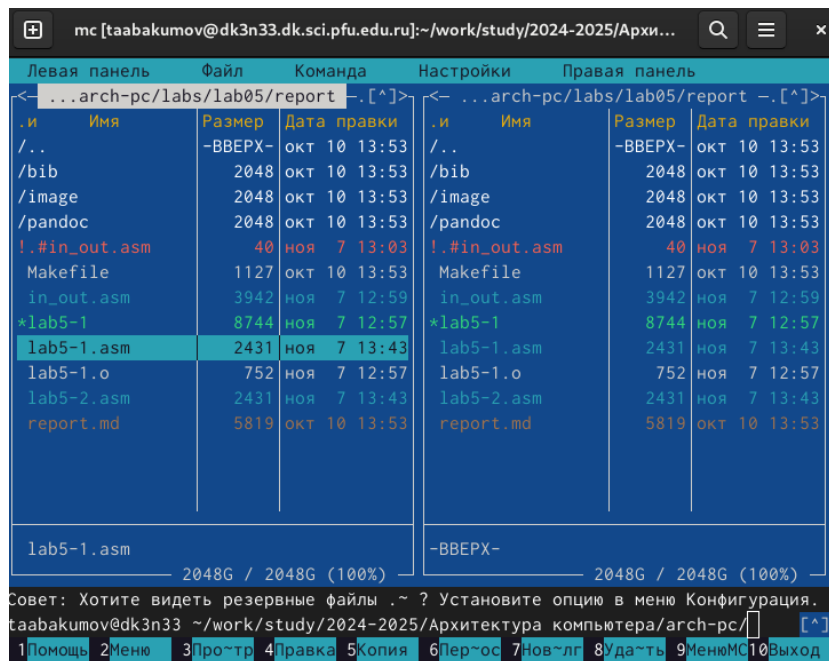


Рис. 3.13: Проверка успешности копирования

12. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используем подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2. Создадим исполняемый файл и проверим его работу (рис. 3.14).

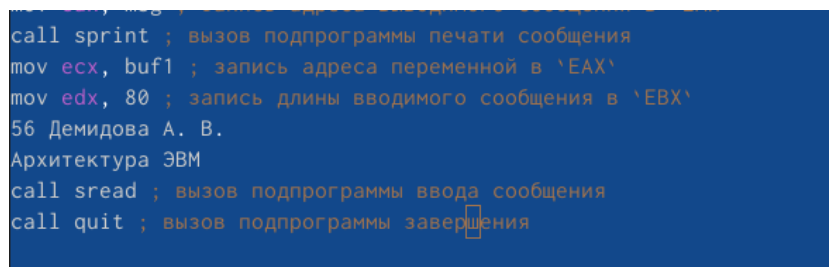


Рис. 3.14: Исправление текста



Рис. 3.15: Создание и проверка исполняемого файла

## 3.4 Задания для самостоятельной работы

1. Создадим копию файла lab5-1.asm. Внесём 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. Проверим работу исполняемого файла. На приглашение ввести строку введём свою ФИО.(рис. 3.16).

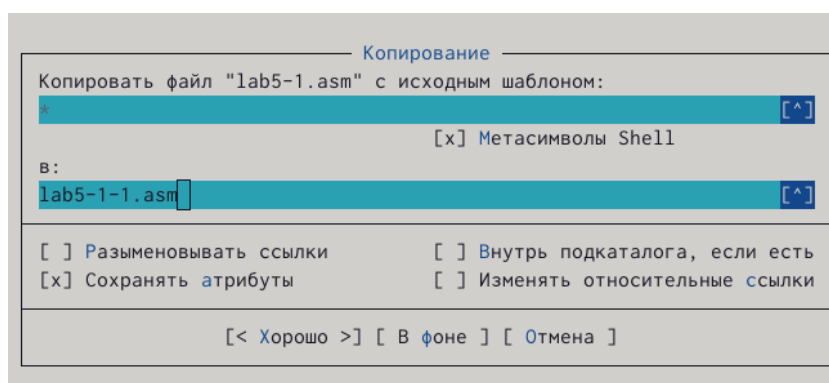


Рис. 3.16: Создание копии файла

```
mc [taabakumov@dk3n33.dk.sci.pfu.edu.ru]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
lab5-1-1.asm [----] 20 L:[ 1+ 0 1/ 26] *(24 /1521b) 1080 0x438
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10
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 '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 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
```

Рис. 3.17: Изменения программы

```

taabakumov@dk3n33 ~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab05/report $ nasm -f elf lab5-1-1.asm
taabakumov@dk3n33 ~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab05/report $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
taabakumov@dk3n33 ~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab05/report $ ./lab5-1-1
Введите строку:
Абакунов Тимофей Александрович
Абакунов Тимофей Александрович

```

Рис. 3.18: Проверка работы файла

Код программы из пункта 1:

SECTION .data ; Секция инициированных данных

msg: DB 'Введите строку:',10

msgLen: EQU \$-msg ; Длина переменной 'msg'

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,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 ; Системный вызов для записи (sys\_write)

mov ebx,1 ; Описатель файла '1' - стандартный вывод

mov ecx,buf1 ; Адрес строки buf1 в ecx

mov edx,buf1 ; Размер строки buf1

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

2. Создадим копию файла lab5-2.asm. Исправим текст программы с использование 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 (рис. 3.19).

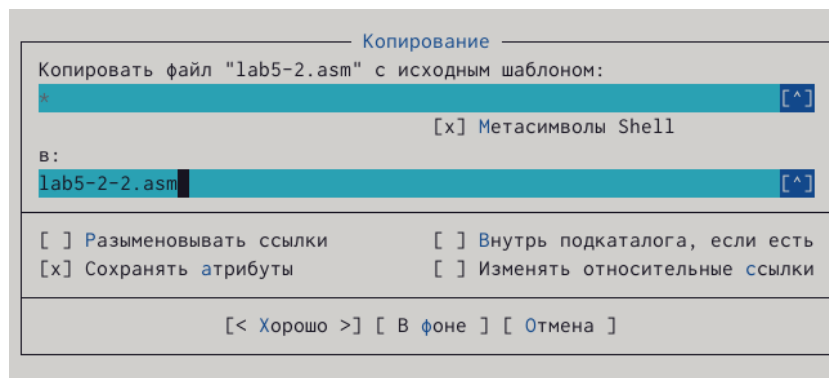


Рис. 3.19: Копирование файла

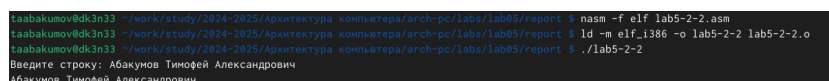


Рис. 3.20: Проверка работы программы

Код программы из пункта 2:

%include 'in\_out.asm'

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 ; Системный вызов для записи (sys\_write)  
mov ebx, 1 ; Описатель файла '1' - стандартный вывод  
mov ecx, buf1 ; Адрес строки buf1 в ecx  
int 80h ; Вызов ядра  
call quit ; вызов подпрограммы завершения



## 4 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.