

Отчёт по лабораторной работе№5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM.

Михайлова Полина Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14

Список иллюстраций

3.1	Открытие Midnight Commander	7
3.2	Создание папки для лабораторной работы	8
3.3	Папка lab05	8
3.4	Создание файла lab5-1.asm	8
3.5	Ввод текста	9
3.6	Проверка наличия текста	9
3.7	Транслирование текста, проверка работоспособности	10
3.8	Создание копии	10
3.9	Внесение изменений в файл	11
3.10	Проверка наличия текста	11
3.11	Транслирование текста, проверка работоспособности	12
3.12	Проверка файла3	12
3.13	транслирование текста в файл	13
3.14	Проверка работоспособности	13

Список таблиц

1 Цель работы

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

2 Задание

1. Создайте копию файла lab6-1.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.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте 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.

3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM.

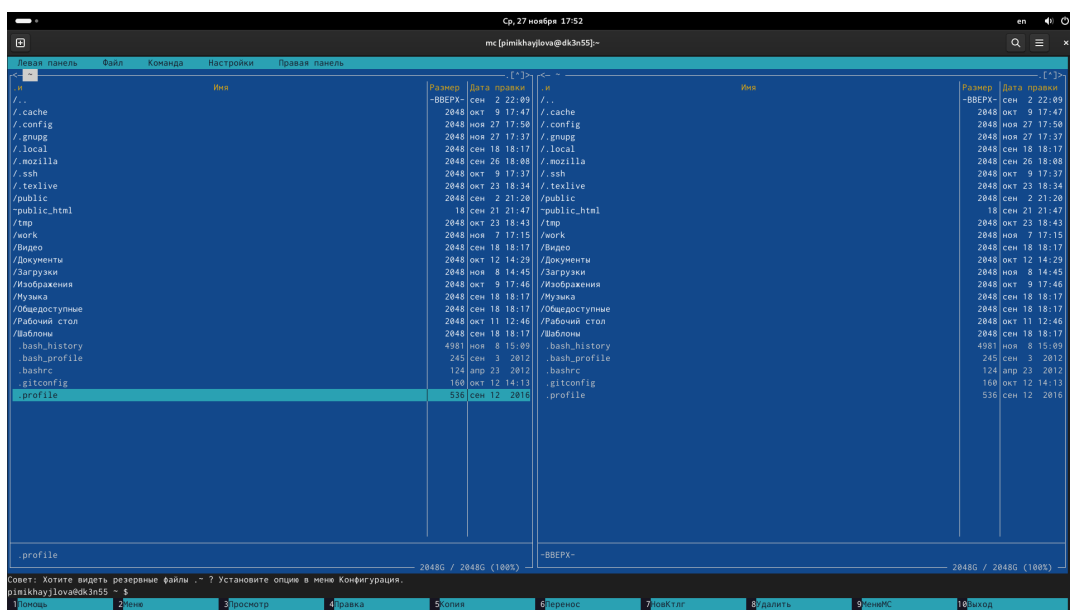


Рис. 3.1: Открытие Midnight Commander

2. С помощью функциональной клавиши F7 создаём папку lab05.

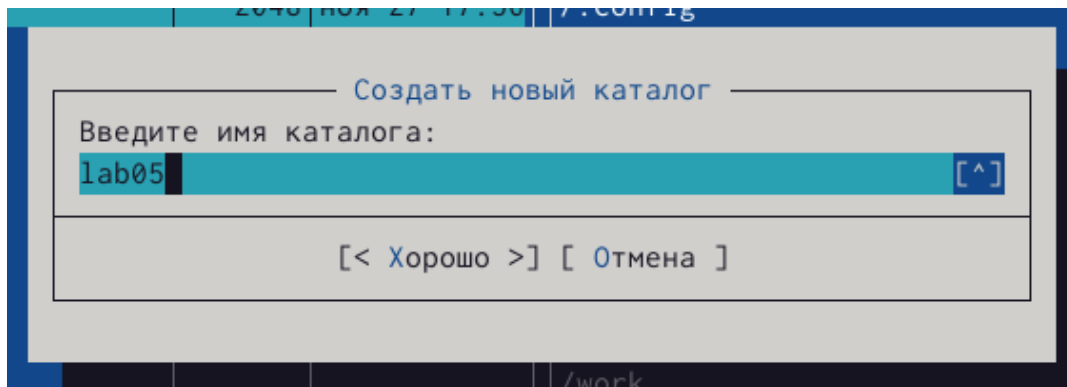


Рис. 3.2: Создание папки для лабораторной работы

3. Убедимся в правильном создании папки.

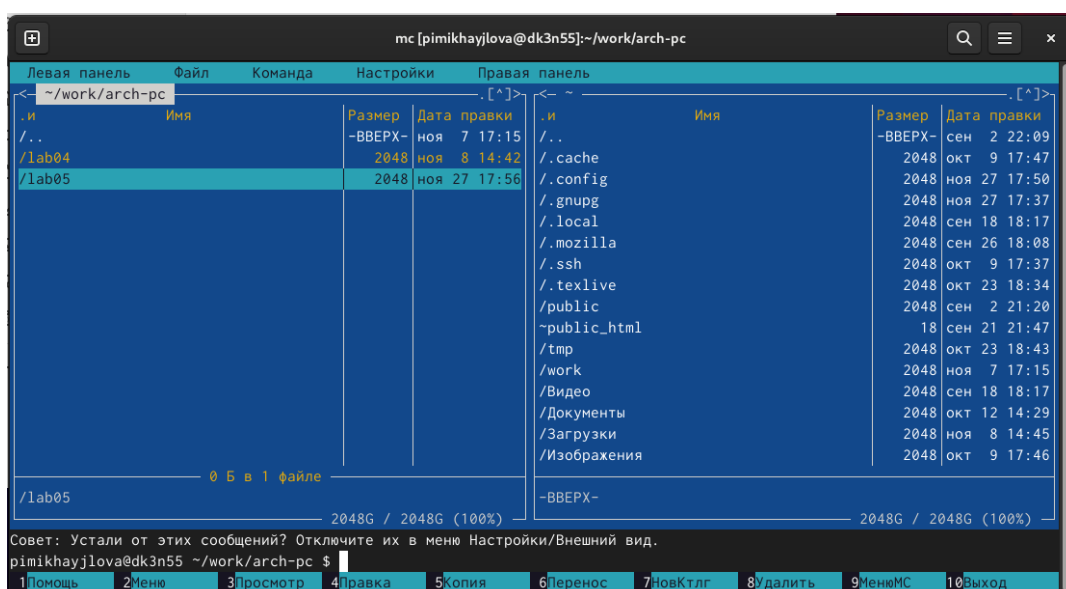


Рис. 3.3: Папка lab05

4. Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm.

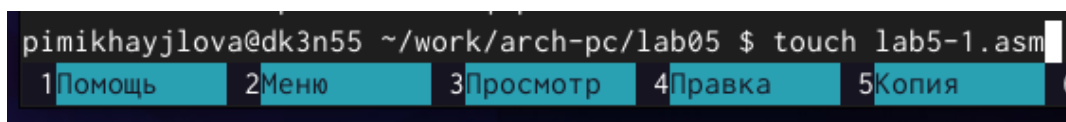
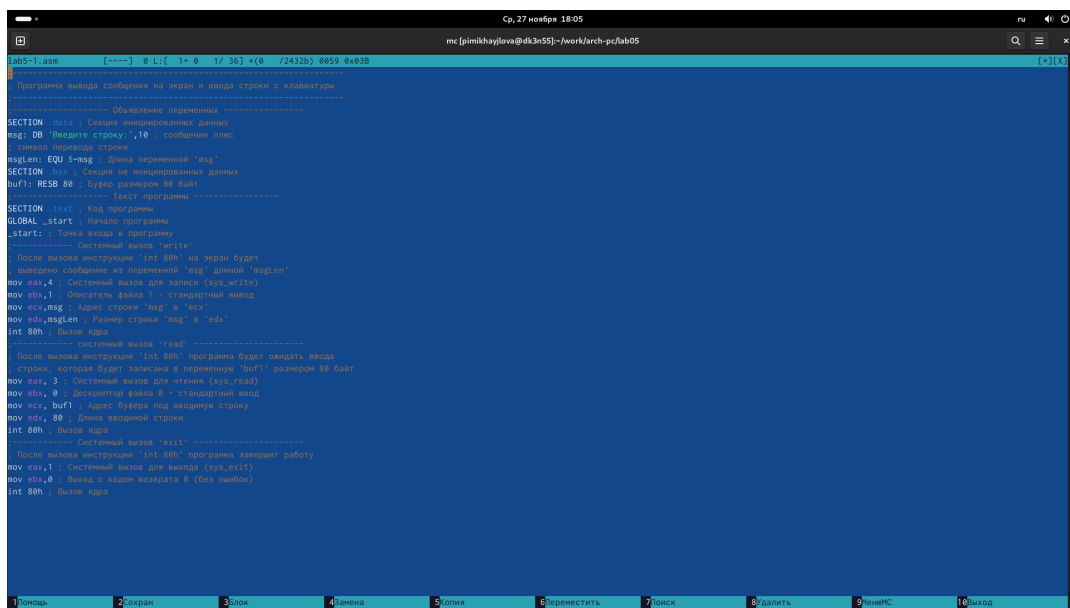


Рис. 3.4: Создание файла lab5-1.asm

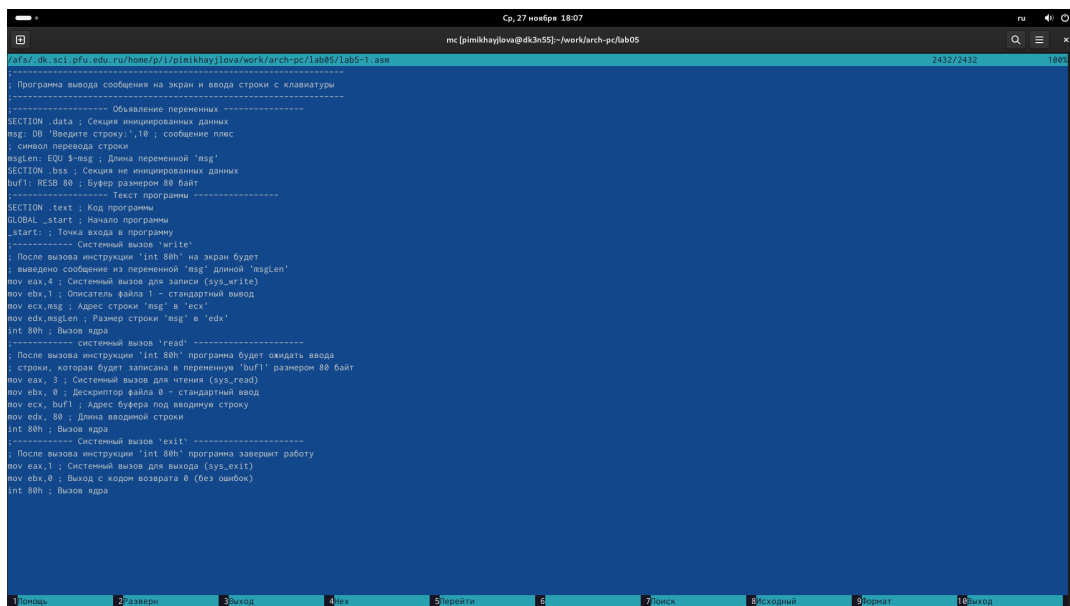
5. С помощью функциональной клавиши F4 откроем файл lab5-1.asm и введём текст из листинга 6.1.



```
lab5-1.asm [-----] 0 L[ 1+ 0 1/ 36] +(0 /24326) 0859 8x838
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
; Объявление переменных
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",10 ; сообщение плюс
; символ перевода строки
msglen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; буфер размером 80 байт
;-----
; Текст программы
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;-----
; Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msglen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описание файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msglen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;-----
; Системный вызов 'read'
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov ebx,3 ; Системный вызов для чтения (sys_read)
mov ecx,0 ; Дескриптор файла 0 - стандартный ввод
mov edx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;-----
; Системный вызов 'exit'
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 3.5: Ввод текста

6. С помощью функциональной клавиши F3 откроем файл lab5-1.asm для проверки наличия текста.



```
./fs/.../lab5-1.asm 2432/2432 100%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
; Объявление переменных
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",10 ; сообщение плюс
; символ перевода строки
msglen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; буфер размером 80 байт
;-----
; Текст программы
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;-----
; Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msglen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описание файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msglen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;-----
; Системный вызов 'read'
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov ebx,3 ; Системный вызов для чтения (sys_read)
mov ecx,0 ; Дескриптор файла 0 - стандартный ввод
mov edx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;-----
; Системный вызов 'exit'
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 3.6: Проверка наличия текста

7. Оттранслируем текст программы lab5-1.asm в объектный файл.

```
pimikhaylova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
pimikhaylova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
pimikhaylova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Polina Mihailova
pimikhaylova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 3.7: Транслирование текста, проверка работоспособности

8. С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm.

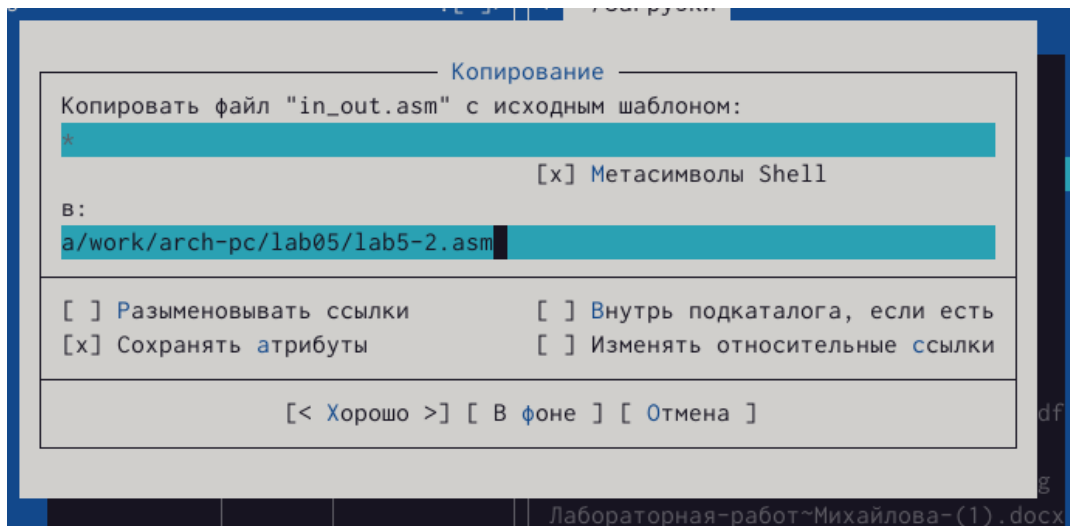


Рис. 3.8: Создание копии

9. Исправим текст программы в соответствии с листингом 6.2

```
lab5-2.asm [-M--] 69 L: [ 1+ 0 1/ 18] *(69 /1225b) 0010 0x00A [*][X]
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%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 вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 3.9: Внесение изменений в файл

10. Проверим, что текст был изменен

```
mc [pimikhaylova@dk3n55]:~/work/arch-pc/lab05
1225/1225 100%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%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 вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

1Помощь 2Разверн 3Выход 4Тех 5Перейти 6 7Поиск 8/сходный 9Формат 10Выход
```

Рис. 3.10: Проверка наличия текста

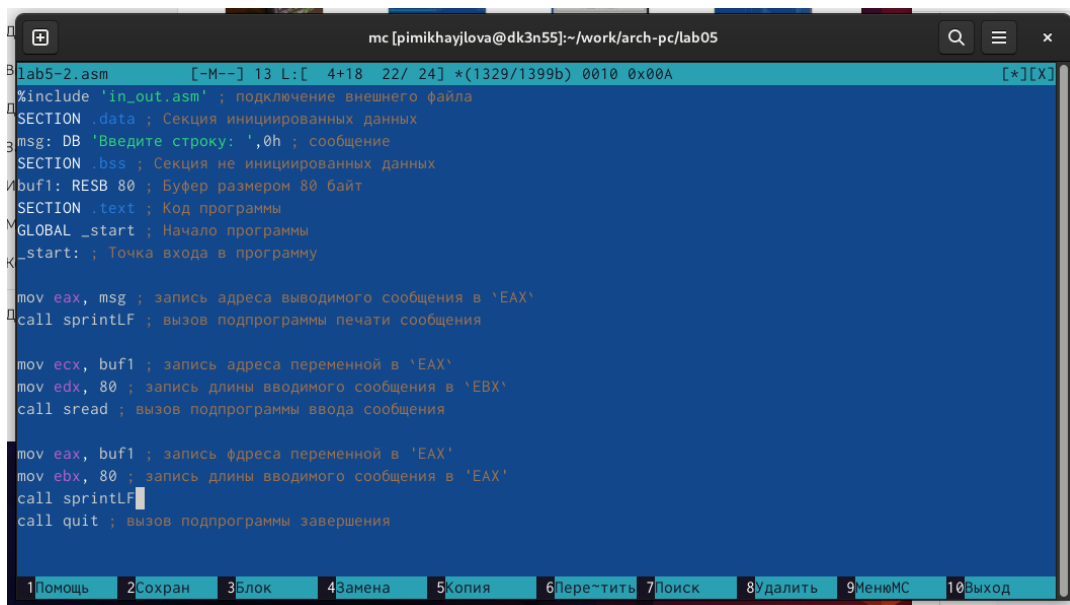
11. Оттранслируем текст программы lab5-2.asm в объектный файл и проверим его работоспособность

```
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab-2.o
ld: невозможно найти lab-2.o: Нет такого файла или каталога
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
123
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 3.11: Транслирование текста, проверка работоспособности

12. Исправьте текст программы, так чтобы она работала по следующему алгоритму:

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



```
lab5-2.asm [-M--] 13 L:[ 4+18 22/ 24] *(1329/1399b) 0010 0x00A [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 вводимого сообщения в 'EBX'
call read ; вызов подпрограммы ввода сообщения

mov eax, buf1 ; запись адреса переменной в 'EAX'
mov ebx, 80 ; запись длины вводимого сообщения в 'EBX'
call sprintf
call quit ; вызов подпрограммы завершения
```

Рис. 3.12: Проверка файла3

13. Оттранслируем текст программы в объектный файл

```
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
```

Рис. 3.13: транслирование текста в файл

14. Внесём изменения в текст программы в файле lab5.asm

```
pimikhayjlova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Михайлова Полина
Михайлова Полина
```

Рис. 3.14: Проверка работоспособности

4 Выводы

В ходе лабораторной работы мною были приобретены практические навыки работы в Midnight Commander, а также освоены инструкции языка ассемблера mov и int. Я научился работать с МС, и с его помощью работать с файлами (Создание, переименовывание, копирование, перемещение, удаление, и тд.)