

# **Отчёта по лабораторной работе № 5**

**Основы работы с Midnight Commander (mc). Структура  
программы на языке ассемблера NASM. Системные вызовы в ОС  
GNU Linux**

Гущина Екатерина Антоновна

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение лабораторной работы</b>	<b>6</b>
<b>3 Задание для самостоятельной работы</b>	<b>13</b>
<b>4 Выводы</b>	<b>17</b>

# **Список иллюстраций**

2.1 Каталог ~/work/arch-pc . . . . .	6
2.2 Создание каталога lab05 . . . . .	7
2.3 Создание файла lab5-1.asm . . . . .	7
2.4 Редактирование файла lab5-1.asm . . . . .	8
2.5 Сборка и запуск программы lab5-1 . . . . .	9
2.6 Добавляем файл in_out.asm . . . . .	9
2.7 Добавляем файл lab5-2.asm . . . . .	10
2.8 Редактирование файла lab5-2.asm . . . . .	11
2.9 Сборка и запуск программы lab5-2 . . . . .	11
2.10 Замена подпрограммы sprintLF на sprint . . . . .	12
2.11 Сборка и запуск измененной программы lab5-2 . . . . .	12
3.1 Редактирование файла lab5-3.asm . . . . .	14
3.2 Сборка и запуск программы lab5-3 . . . . .	14
3.3 Редактирование файла lab5-4.asm . . . . .	15
3.4 Сборка и запуск программы lab5-4 . . . . .	16

# **Список таблиц**

# **1 Цель работы**

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

## 2 Выполнение лабораторной работы

Откроем Midnight Commander командой `mc` и перейдем в каталог `~/work/arch-pc`, созданный при выполнении лабораторной работы №4:

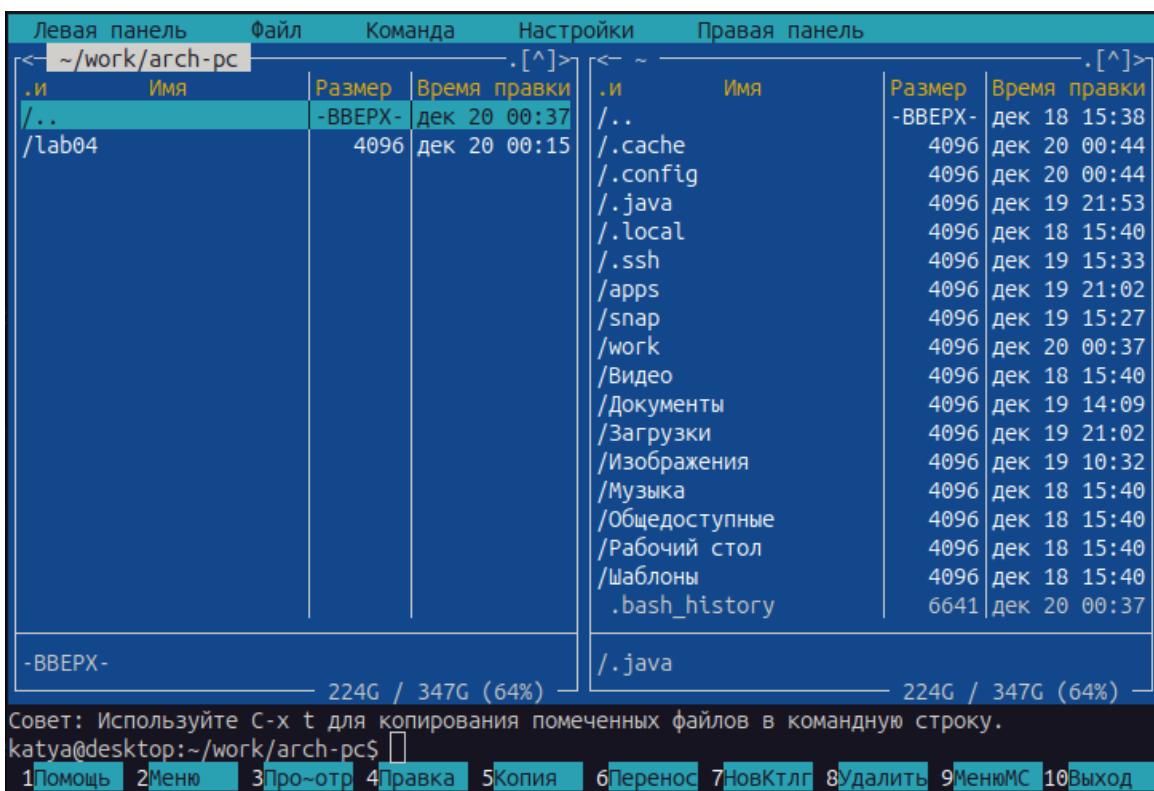


Рисунок 2.1: Каталог ~ /work/arch-рс

С помощью функциональной клавиши F7 создадим папку lab05:

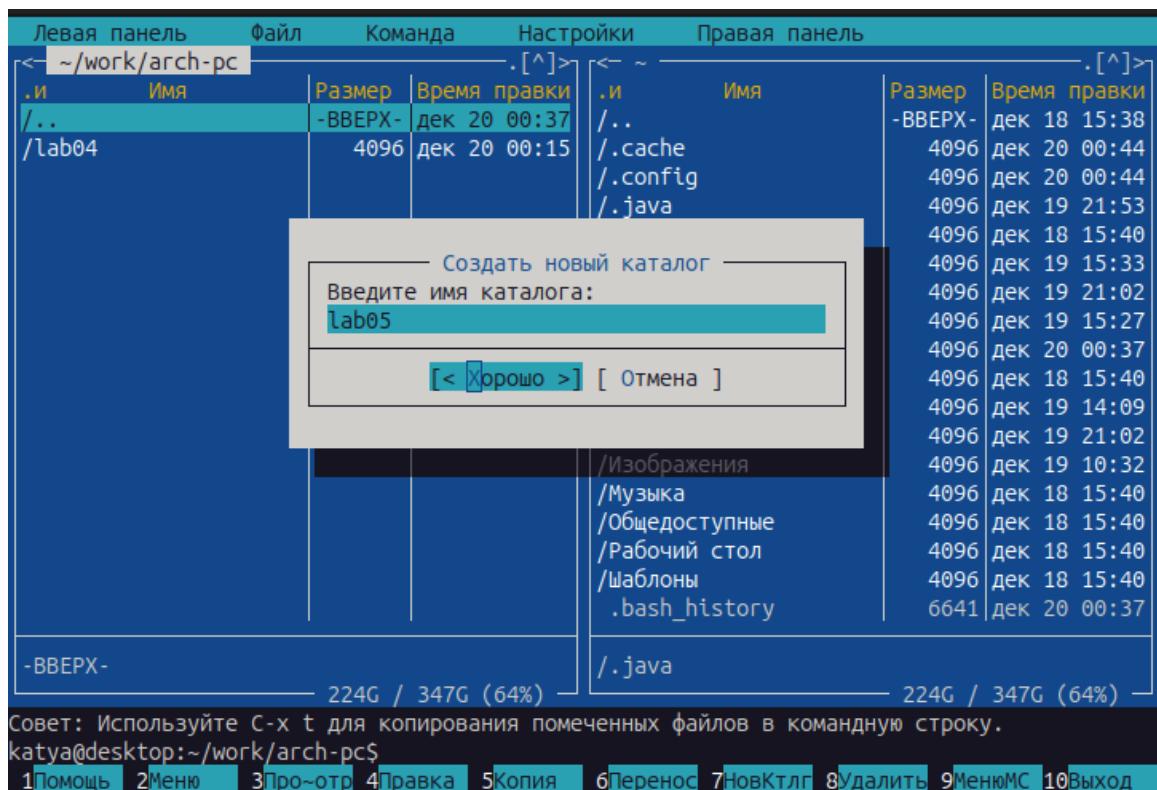


Рисунок 2.2: Создание каталога lab05

Перейдем в созданный каталог и в строке ввода командой touch создадим файл lab5-1.asm:

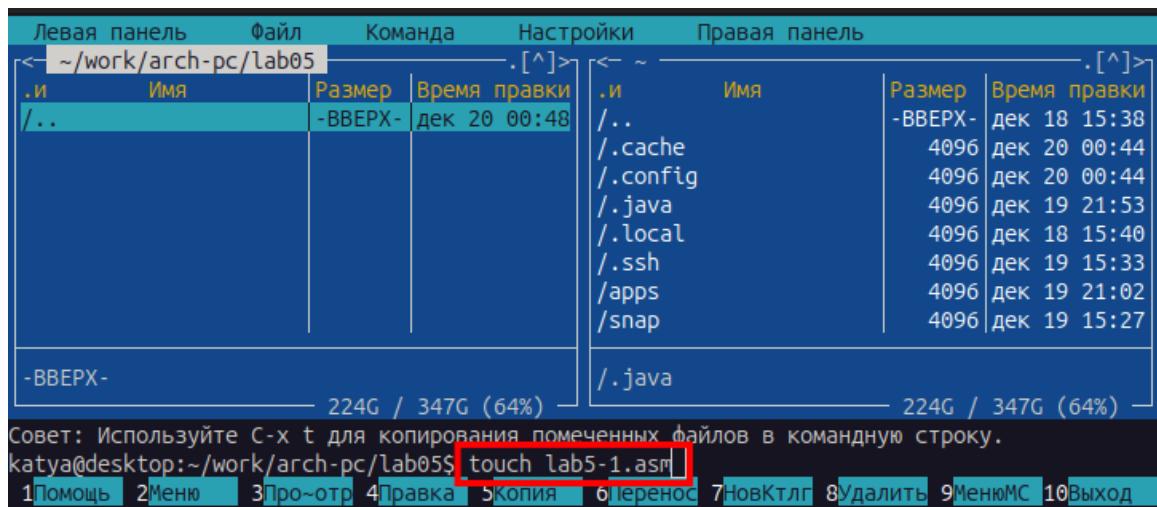


Рисунок 2.3: Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенным редакторе, введем текст программы, сохраним изменения (F2) и выйдем из редактора (F10):

The screenshot shows a text editor window with assembly code. The code is organized into sections: .text, \_start, write system call, read system call, and exit system call. Each section contains comments explaining its purpose. The assembly instructions use registers eax, ebx, ecx, edx, and the instruction int 80h. The code is written in Russian.

```
/home/katya/w-05/lab5-1.asm [----] 0 L:[ 15+31 46/ 46] *(2718/2718b) <EOF> [*][X]

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      ; Описатель файла '1' - стандартный вывод
    mov ecx, msg    ; Адрес строки hello в ecx
    mov edx, msgLen ; Размер строки hello
    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 eax, 3      ; Системный вызов для чтения (sys_read)
    mov ebx, 0      ; Описатель файла '0' - стандартный ввод
    mov ecx, buf    ; Адрес буфера под вводимую строку
    mov edx, 80     ; Длина буфера
    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
```

1Помощь 2Сох~ить 3Блок 4Замена 5Копия 6Пер~ить 7Поиск 8Удалить 9МенюМС 10Выход

Рисунок 2.4: Редактирование файла lab5-1.asm

С помощью клавиши F3 откроем файл lab5-1.asm для просмотра и убедимся, что файл содержит текст программы.

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку „Ведите строку:“ и ожидает ввода с клавиатуры. На запрос введем ФИО:

```

katya@desktop:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
katya@desktop:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
katya@desktop:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Гущина Екатерина Антоновна
katya@desktop:~/work/arch-pc/lab05$ 

```

Рисунок 2.5: Сборка и запуск программы lab5-1

Скачаем файл in\_out.asm со страницы курса в ТУИС. Скопируем файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5:

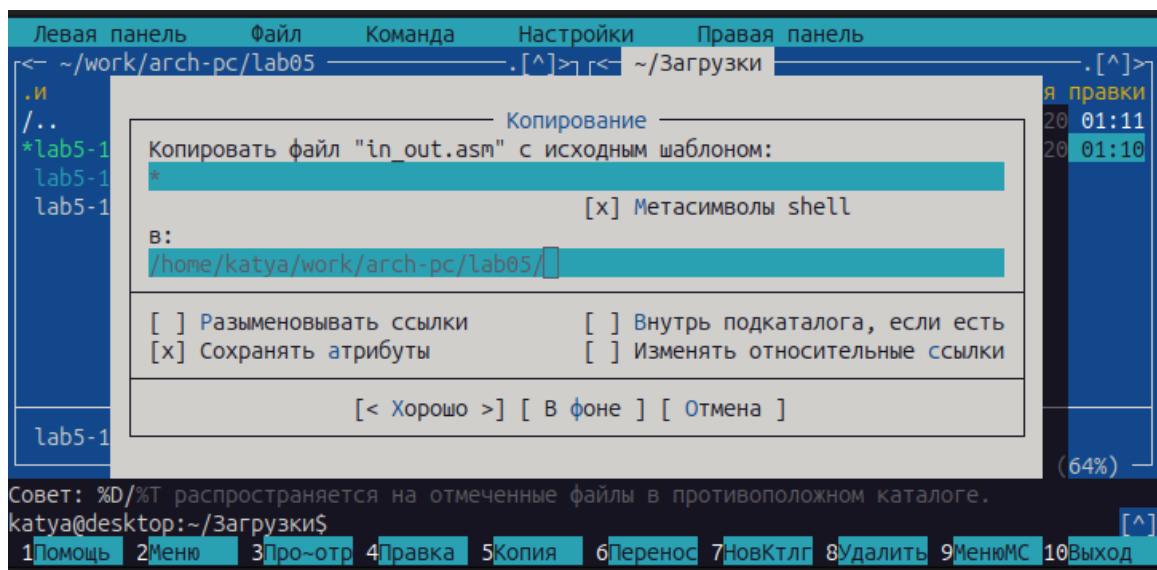


Рисунок 2.6: Добавляем файл in\_out.asm

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm. Выделим файл lab5-1.asm, нажмем клавишу F6, введем имя файла lab5-2.asm и нажмем клавишу Enter:

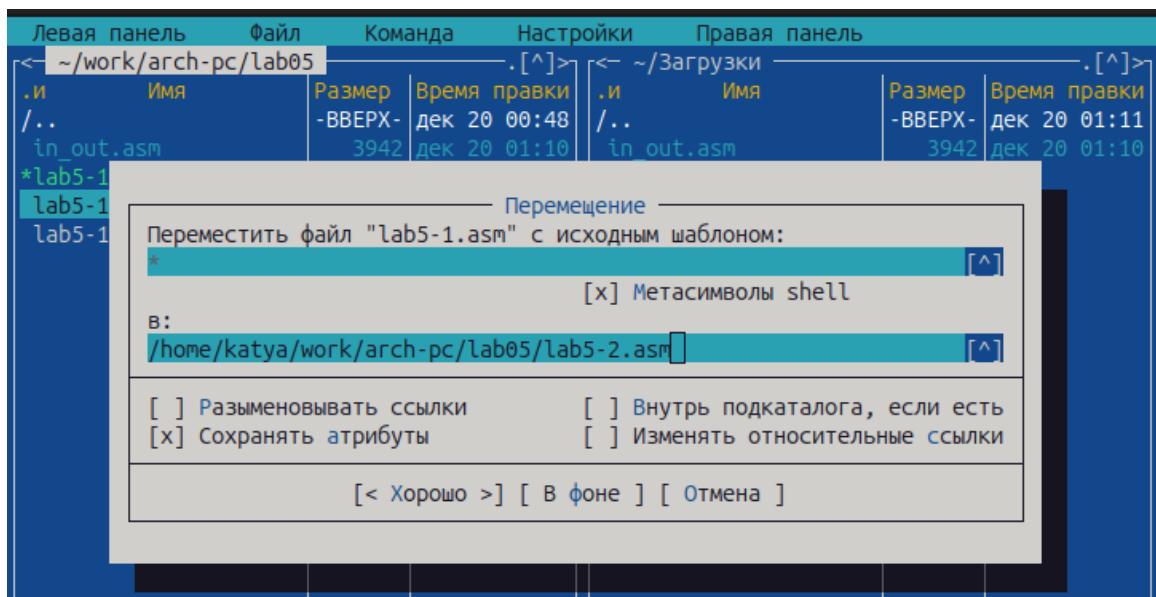


Рисунок 2.7: Добавляем файл lab5-2.asm

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 из задания:

```
/home/katya/w~05/lab5-2.asm [---] 0 L:[ 1+ 2 3/ 26] *(188 /1407b) 0059 0x03B [*][X]
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;

%include 'in_out.asm'<-> ; подключение внешнего файла

----- Объявление переменных -----
section .data ; Секция инициализированных данных
    msg: db 'Введите строку: ',0h ; сообщение

section .bss ; Секция неинициализированных данных
    buf resb 80 ; Буфер размером 80 байт

section .text ; Код программы
    global _start ; Начало программы
_start:
    mov eax, msg ; Запись адреса выводимого сообщения в 'eax'
    call sprintLF ; Вызов подпрограммы печати сообщения

    mov ecx, buf ; Запись адреса буфера в 'ecx'
    mov edx, 80 ; Запись длины буфера в 'edx'
    call sread ; Вызов подпрограммы ввода сообщения

    call quit ; Вызов подпрограммы завершения

1 Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пер~ить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рисунок 2.8: Редактирование файла lab5-2.asm

Создадим исполняемый файл и провим его работу:

```
katya@desktop:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
katya@desktop:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
katya@desktop:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Гущина Екатерина Антоновна
katya@desktop:~/work/arch-pc/lab05$ 
```

Рисунок 2.9: Сборка и запуск программы lab5-2

Программа с использованием подпрограмм работает так же.

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint.

```

/home/katya/w~05/lab5-2.asm [---] 0 L:[ 1+ 2 3/ 26] *(188 /1407b) 0059 0x03B [*][X]
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----

%include 'in_out.asm'<-> ; подключение внешнего файла

----- Объявление переменных -----
section .data ; Секция инициализированных данных
    msg: db 'Введите строку: ',0h ; сообщение

section .bss ; Секция неинициализированных данных
    buf resb 80 ; Буфер размером 80 байт

section .text ; Код программы
    global _start ; Начало программы
_start:
    mov eax, msg ; Запись адреса выводимого сообщения в 'eax'
    call sprint ; Вызов подпрограммы печати сообщения

    mov ecx, buf ; Запись адреса буфера в 'ecx'
    mov edx, 80 ; Запись длины буфера в 'edx'
    call sread ; Вызов подпрограммы ввода сообщения

    call quit ; Вызов подпрограммы завершения

----- Меню -----
1 Помощь 2 Сох~ить 3 Блок 4 Замена 5 Копия 6 Пер~ить 7 Поиск 8 Удалить 9 Меню MS 10 Выход

```

Рисунок 2.10: Замена подпрограммы `sprintLF` на `sprint`

Создадим исполняемый файл и проверим его работу:

```

katya@desktop:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
katya@desktop:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
katya@desktop:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Гущина Екатерина Антоновна
katya@desktop:~/work/arch-pc/lab05$ 

```

Рисунок 2.11: Сборка и запуск измененной программы `lab5-2`

Разница в том, что после строки приглашения больше нет перевода строки, ввод продолжается на той же строке.

## **3 Задание для самостоятельной работы**

1. Создадим копию файла lab5-1.asm - файл lab5-3.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму:
  - вывести приглашение типа «Ведите строку:»;
  - ввести строку с клавиатуры;
  - вывести введённую строку на экран.

```
/home/katya/w~05/lab5-3.asm [----] 0 L:[ 29+31 60/ 60] *(3630/3630b) <EOF> [*][X]
    mov edx, msgLen ; Размер строки hello
    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 eax, 3      ; Системный вызов для чтения (sys_read)
    mov ebx, 0      ; Описатель файла '0' - стандартный ввод
    mov ecx, buf    ; Адрес буфера под вводимую строку
    mov edx, 80      ; Длина буфера
    int 80h          ; Вызов ядра
    mov [count], eax; Количество введенных символов возвращается в eax
                  ; запоминаем в переменной count

;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'count'

    mov eax, 4      ; Системный вызов для записи (sys_write)
    mov ebx, 1      ; Описатель файла '1' - стандартный вывод
    mov ecx, buf    ; Адрес буфера buf в есх
    mov edx, [count]; Размер введенной 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
```

Рисунок 3.1: Редактирование файла lab5-3.asm

Программа выводит введенную строку.

2. Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем свою фамилию.

```
katya@desktop:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
katya@desktop:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
katya@desktop:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Гущина
Гущина
katya@desktop:~/work/arch-pc/lab05$ 
```

Рисунок 3.2: Сборка и запуск программы lab5-3

3. Создадим копию файла lab5-2.asm - файл lab5-4.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.

```
/home/katya/w~05/lab5-4.asm [---] 0 L:[ 1+ 4 5/ 32 ] *(304 /1848b) 0010 0x00A [*][X]
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
; с последующим ее выводом
;-----

%include 'in_out.asm'<-> ; подключение внешнего файла

;----- Объявление переменных -----
section .data ; Секция инициализированных данных
    msg: db 'Введите строку: ',0h ; сообщение

section .bss ; Секция неинициализированных данных
    buf 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, buf ; Запись адреса буфера в 'ecx'
    mov edx, 80 ; Запись длины буфера в 'edx'
    call sread ; Вызов подпрограммы ввода сообщения
<-----><-----> ; Количество введенных символов возвращается в eax,
    mov [buf + eax], byte 0 ; вставим за последним символом нулевой символ ('\0')

    mov eax, buf ; Запись адреса буфера в 'eax'
    call sprint ; Вызов подпрограммы печати введенной строки

    call quit ; Вызов подпрограммы завершения

1 Помощь 2 Сох~ить 3 Блок 4 Замена 5 Копия 6 Пер~ить 7 Поиск 8 Удалить 9 МенюМС 10 Выход
```

Рисунок 3.3: Редактирование файла lab5-4.asm

4. Создадим исполняемый файл и проверим его работу:

```
katya@desktop:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
katya@desktop:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
katya@desktop:~/work/arch-pc/lab05$ ./lab5-4
Введите строку: Гущина
Гущина
katya@desktop:~/work/arch-pc/lab05$
```

Рисунок 3.4: Сборка и запуск программы lab5-4

Программа выводит введенную строку.

## **4 Выводы**

Приобрели практические навыки работы в Midnight Commander. Освоили инструкции языка ассемблера mov и int.