

Отчёт по лабораторной работе №6

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

Кочина Дарья Сергеевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	16

Список иллюстраций

4.1	Окно Midnight Commander. Смена текущего каталога	7
4.2	Окно Midnight Commander. Создание каталога	8
4.3	Окно Midnight Commander. Создание файла	8
4.4	Окно Midnight Commander. Редактор mcedit	9
4.5	Файл lab6-1.asm	10
4.6	Ввод команд и вывод ФИО	10
4.7	Копирование файла lab6-1.asm	11
4.8	Создании копии файла lab6-1.asm	11
4.9	Копия файла с названием lab6-2.asm	12
4.10	Текст файла из листинга	12
4.11	Проверка файла	12
4.12	Изменённый текст файла из листинга	13
4.13	Проверка изменённого файла	13
4.14	Изменённый файл lab6-3.asm	14
4.15	Проверка изменённого файла lab6-3.asm	14
4.16	Изменённый файл lab6-4.asm	15
4.17	Проверка изменённого файла lab6-4.asm	15

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

Приобрести практические навыки работы в Midhigh Commander. А также освоить инструкции языка ассемблера `mov` и `int`.

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

4 Выполнение лабораторной работы

1. Я открыла Midnight Commander. Я перешла в каталог, созданный при выполнении лабораторной работы N5. (рис. 4.1)

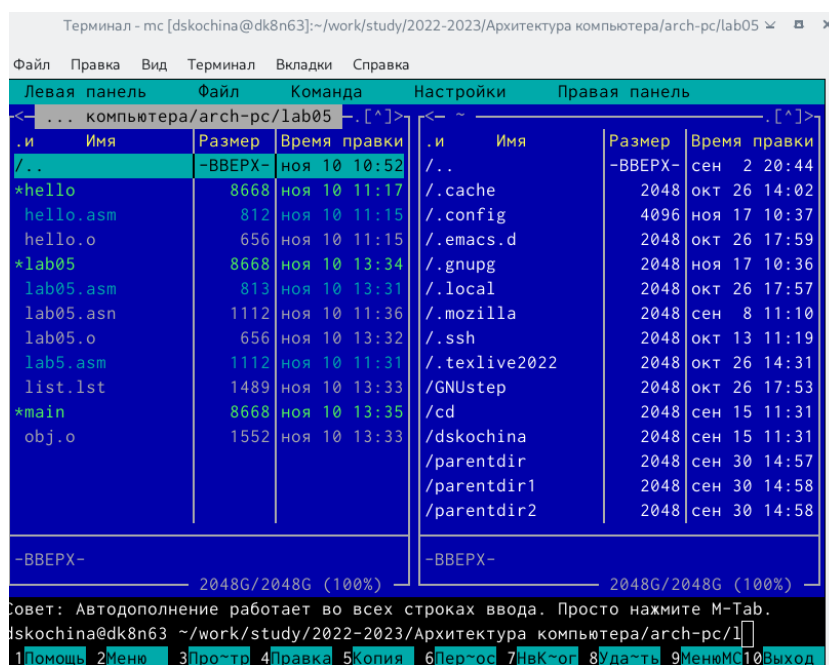


Рис. 4.1: Окно Midnight Commander. Смена текущего каталога

2. С помощью функциональной клавиши F7 я создала папку lab06 и перешла в созданный каталог. (рис. 4.2)

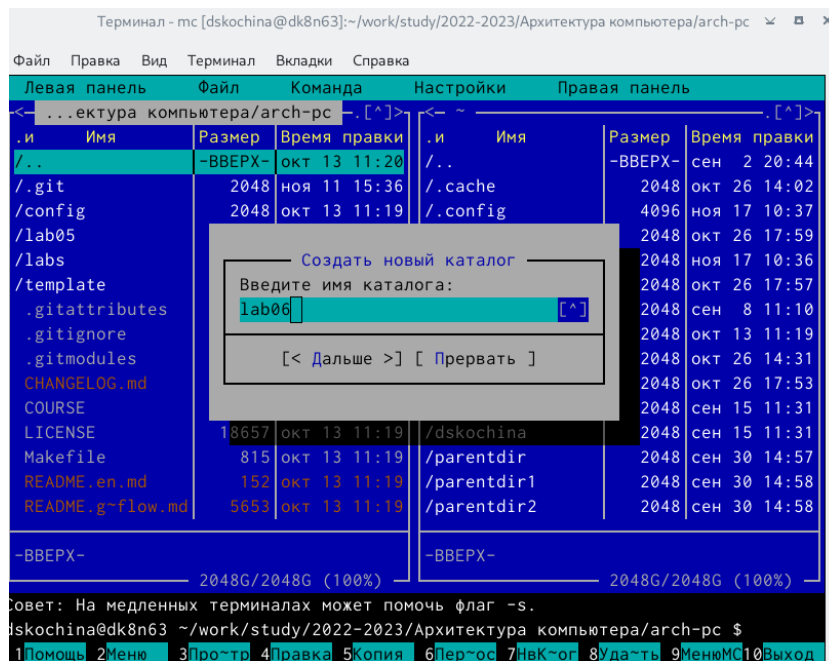


Рис. 4.2: Окно Midnight Commander. Создание каталога

3. Пользуясь строкой ввода и командой touch, я создала файл lab6-1.asm. (рис. 4.3)

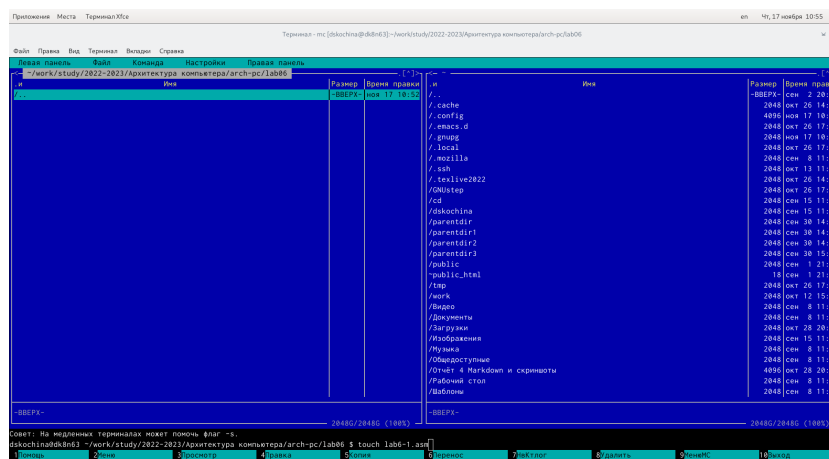


Рис. 4.3: Окно Midnight Commander. Создание файла

4. С помощью функциональной клавиши F4 я открыла файл lab6-1.asm для редактирования во встроенном редакторе. Я ввела текст программы из листинга, сохранила изменения и закрыла файл. (рис. 4.4)

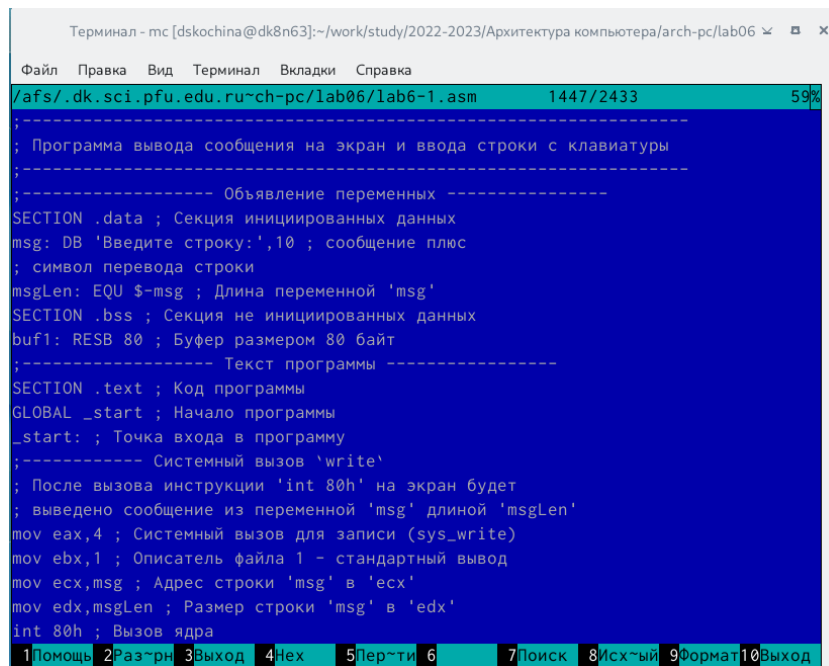

```

lab6-1.asm      [-M--] 67 L:[ 1+ 0 1/ 36] *(67 /2434b) 0010 0x
;-----
; 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 ; 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
;----- 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 ; Дескриптор файла 0 - стандартный ввод
mov ecx, 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. 4.4: Окно Midnight Commander. Редактор mcedit

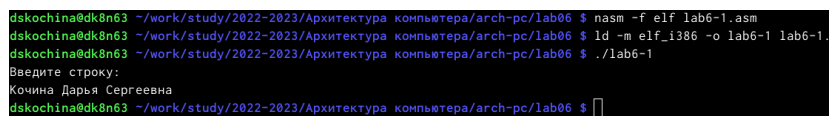
5. С помощью функциональной клавиши F3 я открыла файл lab6-1.asm для просмотра. А также убедилась, что файл содержит текст программы. (рис. 4.5)



```
Терминал - mc [dskochina@dk8n63]:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06
Файл  Правка  Вид  Терминал  Вкладки  Справка
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab06/lab6-1.asm  1447/2433  59%
;-----
; 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 0x80' 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 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 0x80 ; Вызов ядра
1Помощь 2Раз-рн 3Выход 4hex 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 4.5: Файл lab6-1.asm

6. Я оттранслировала текст программы lab6-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос я ввела ФИО. (рис. 4.6)



```
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-1.asm
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-1
Введите строку:
Кочина Дарья Сергеевна
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $
```

Рис. 4.6: Ввод команд и вывод ФИО

7. Я скачала файл in_out.asm со страницы курса в ТУИС. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется. Для этого в одной из панелей mc я открыла каталог с файлом lab6-1.asm. В другой панели - каталог со скачанным файлом in_out.asm. Я скопировала файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5. (рис. 4.7)

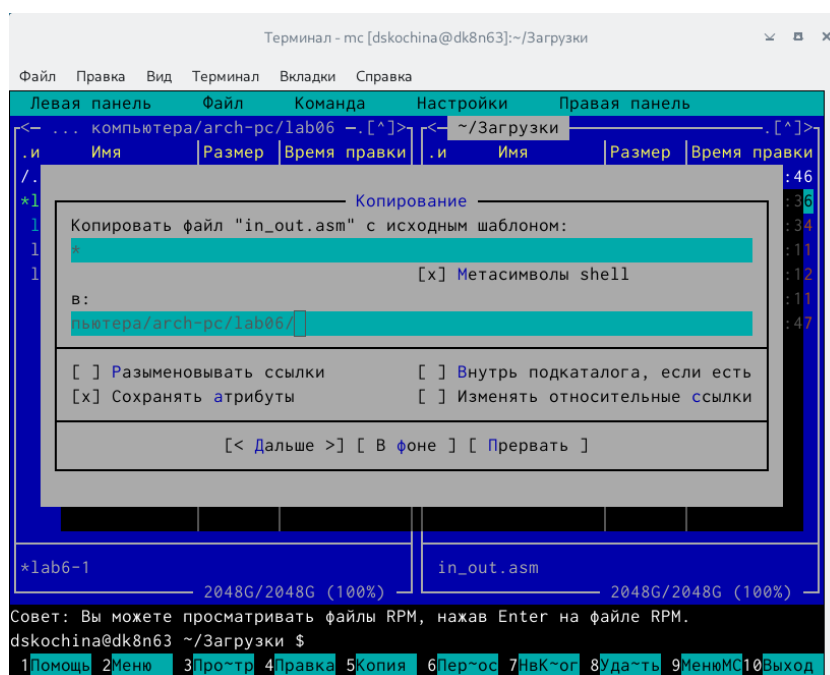


Рис. 4.7: Копирование файла lab6-1.asm

8. Я создала копию файла lab6-1.asm с именем lab6-2.asm. (рис. 4.8, 4.9)

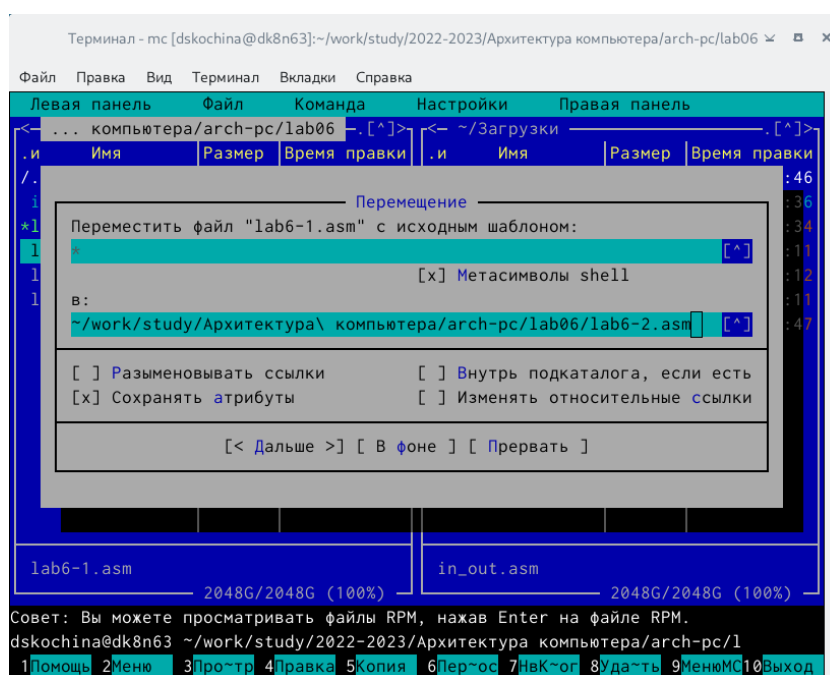


Рис. 4.8: Создании копии файла lab6-1.asm

~ /work/study/2022-2023/Архитектура компьютера/arch-pc/lab06		. [^] >	
Имя	Размер	Время правки	
./..	-ВВЕРХ-	ноя 17 11:28	
in_out.asm	3942	ноя 17 11:36	
*lab6-1	8744	ноя 17 11:33	
lab6-1.asm	2430	ноя 17 11:30	
lab6-1.asm.save	1	ноя 17 10:57	
lab6-1.o	752	ноя 17 11:32	
lab6-2.asm	2430	ноя 17 11:30	

Рис. 4.9: Копия файла с названием lab6-2.asm

9. Я исправила текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом. Я создала исполняемый файл и проверила его работу. (рис. 4.10, 4.11)

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dskochina/w
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#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 sprintLF ; 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: Текст файла из листинга

```
dskochina@dk3n35 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab06
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-4.asm
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-4
Введите строку:
Кочина Дарья Сергеевна
Кочина Дарья Сергеевна
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $
```

Рис. 4.11: Проверка файла

10. В файле lab6-2.asm я заменила подпрограмму sprintLF на sprint. Я создала исполняемый файл и проверила его работу. (рис. 4.12, 4.13)

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dskochina/w
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#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 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. 4.12: Изменённый текст файла из листинга

```

dskochina@dk8n63 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab06
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-2
Введите строку: Кочина Дарья Сергеевна
dskochina@dk8n63 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ 

```

Рис. 4.13: Проверка изменённого файла

Выполнение самостоятельной работы

11. Я создала копию файла lab6-1.asm. Внесла изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по алгоритму. (рис. 4.14)

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dskochina
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
int 80h
mov eax,1
mov ebx,0
int 80h

```

Рис. 4.14: Изменённый файл lab6-3.asm

12. Я получила исполняемый файл и проверила его работу. На приглашение ввести строку я ввела свою фамилию. (рис. 4.15)

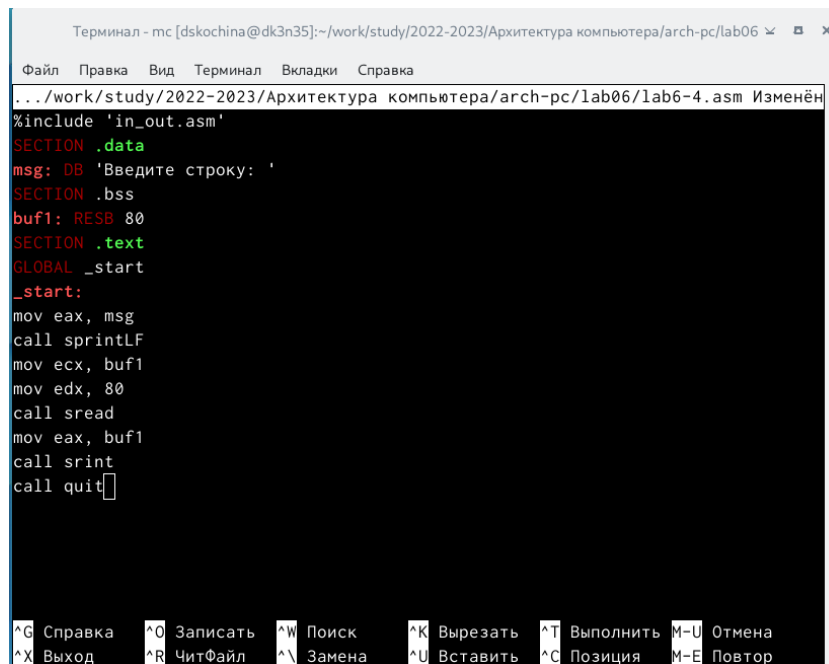
```

dskochina@dk3n35 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab06
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-3.asm
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-3
Введите строку:
Кочина Дарья Сергеевна
Кочина Дарья Сергеевна
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ 

```

Рис. 4.15: Проверка изменённого файла lab6-3.asm

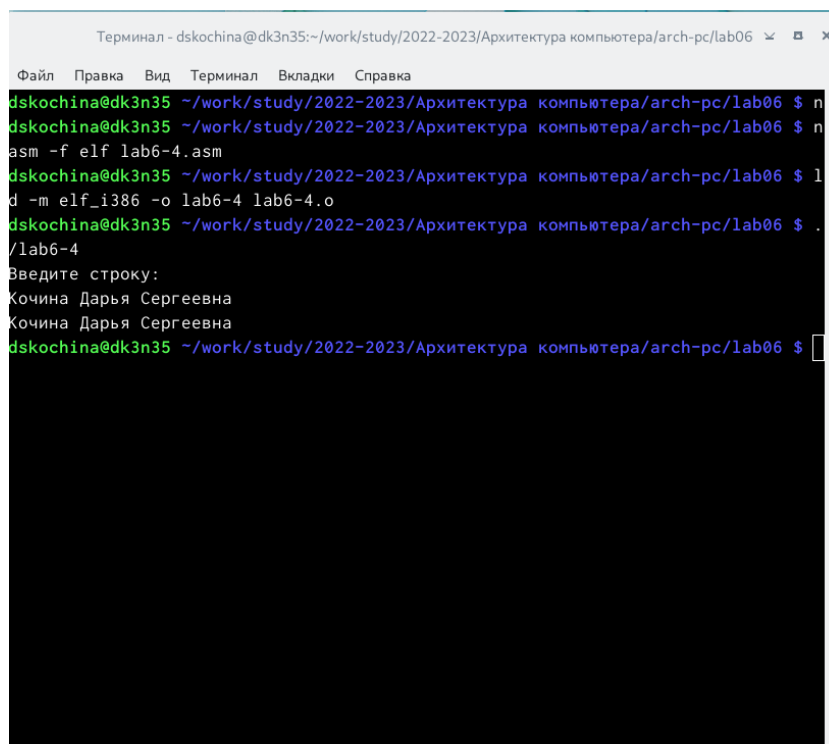
13. Я создала копию файла lab6-2.asm. Исправила текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по алгоритму. (рис. 4.16)



```
Терминал - mc [dskochina@dk3n35]:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 x
Файл Правка Вид Терминал Вкладки Справка
.../work/study/2022-2023/Архитектура компьютера/arch-pc/lab06/lab6-4.asm Изменён
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: '
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
mov eax, buf1
call sprint
call quit
```

Рис. 4.16: Изменённый файл lab6-4.asm

14. Я создала исполняемый файл и проверила его работу. (рис. 4.17)



```
Терминал - dskochina@dk3n35:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 x
Файл Правка Вид Терминал Вкладки Справка
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ n
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ n
asm -f elf lab6-4.asm
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ l
d -m elf_i386 -o lab6-4 lab6-4.o
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ .
/lab6-4
Введите строку:
Кочина Дарья Сергеевна
Кочина Дарья Сергеевна
dskochina@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $
```

Рис. 4.17: Проверка изменённого файла lab6-4.asm

5 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander. А также освоила инструкции языка ассемблера `mov` и `int`.