

Лабораторная работа №6

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

Медникова Екатерина Михайловна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	9
4	Самостоятельная работа	18
5	Выводы	20

Список иллюстраций

3.1	Переход в каталог	9
3.2	Создание папки	10
3.3	Создание файла	11
3.4	Встроенный редактор	12
3.5	Ввод текста	13
3.6	Открытие файла для просмотра	14
3.7	Компоновка файла	14
3.8	Скачивание файла	15
3.9	Копирование файла в каталог	15
3.10	Создание копии файла	16
3.11	Исправление текста программы	17
3.12	Проверка работы	17
3.13	Проверка работы	17
4.1	Копия файла и изменения в программе	18
4.2	Проверка работы	19
4.3	Копия файла и правка в программе	19
4.4	Создание файла и его проверка	19

Список таблиц

1 Цель работы

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

2 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**. В **Midnight Commander** используются функциональные клавиши **F1 — F10**, к которым привязаны часто выполняемые операции.

Следующие комбинации клавиш облегчают работу с **Midnight Commander**:

- **Tab** используется для переключения между панелями;
- Стрелки вверх и вниз используются для навигации, **Enter** для входа в каталог или открытия файла (если в файле расширений **mc.ext** заданы правила связи определённых расширений файлов с инструментами их запуска или обработки);
- **Ctrl + u** (или через меню Команда > Переставить панели) меняет местами содержимое правой и левой панелей;
- **Ctrl + o** (или через меню Команда > Отключить панели) скрывает или возвращает панели **Midnight Commander**, за которыми доступен для работы командный интерпретатор оболочки и выводимая туда информация.
- **Ctrl + x + d** (или через меню Команда > Сравнить каталоги) позволяет сравнить содержимое каталогов, отображаемых на левой и правой панелях.

Дополнительную информацию о **Midnight Commander** можно получить по команде **man mc** и на странице проекта.

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).

Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти:

- DB (define byte) — определяет переменную размером в 1 байт;
- DW (define word) — определяет переменную размером в 2 байта (слово);
- DD (define double word) — определяет переменную размером в 4 байта (двойное слово);
- DQ (define quad word) — определяет переменную размером в 8 байт (четверное слово);
- DT (define ten bytes) — определяет переменную размером в 10 байт.

Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти.

Синтаксис директив определения данных следующий:

```
DB [ , ] [ , ]
```

Для объявления неинициированных данных в секции .bss используются директивы resb, resw, resd и другие, которые сообщают ассемблеру, что необходимо зарезервировать заданное количество ячеек памяти.

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде

```
mov dst,src
```

Здесь операнд dst — приёмник, а src — источник.

Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде

```
int n
```

Здесь n — номер прерывания, принадлежащий диапазону 0–255.

3 Выполнение лабораторной работы

1. Открыла Midnight Commander. Пользуясь клавишами стрелок вверх, вниз и Enter перешла в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №5.

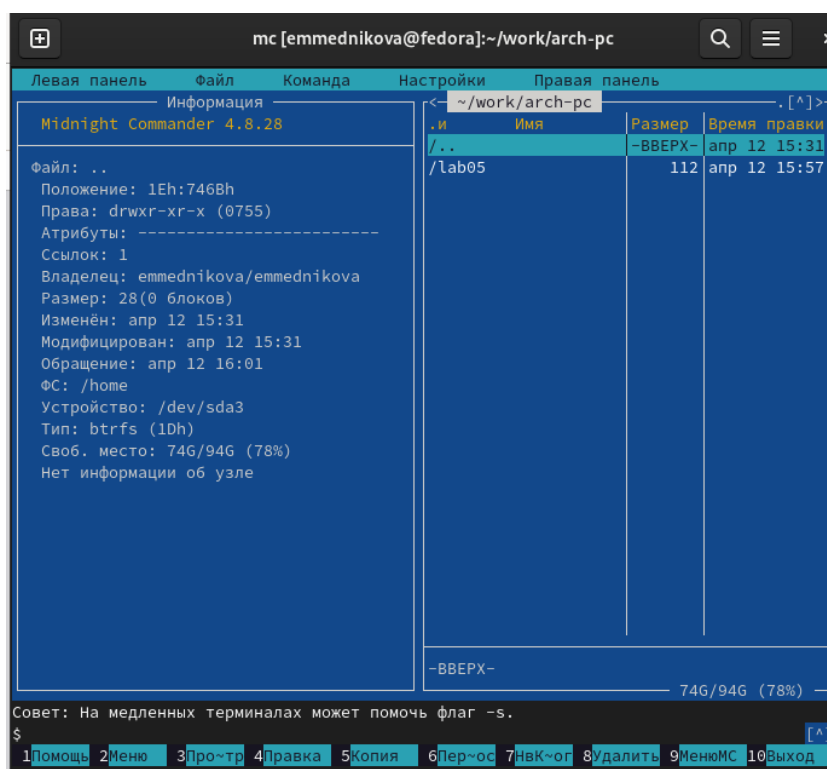


Рис. 3.1: Переход в каталог

2. С помощью функциональной клавиши F7 создала папку lab06 и перешла в созданный каталог.

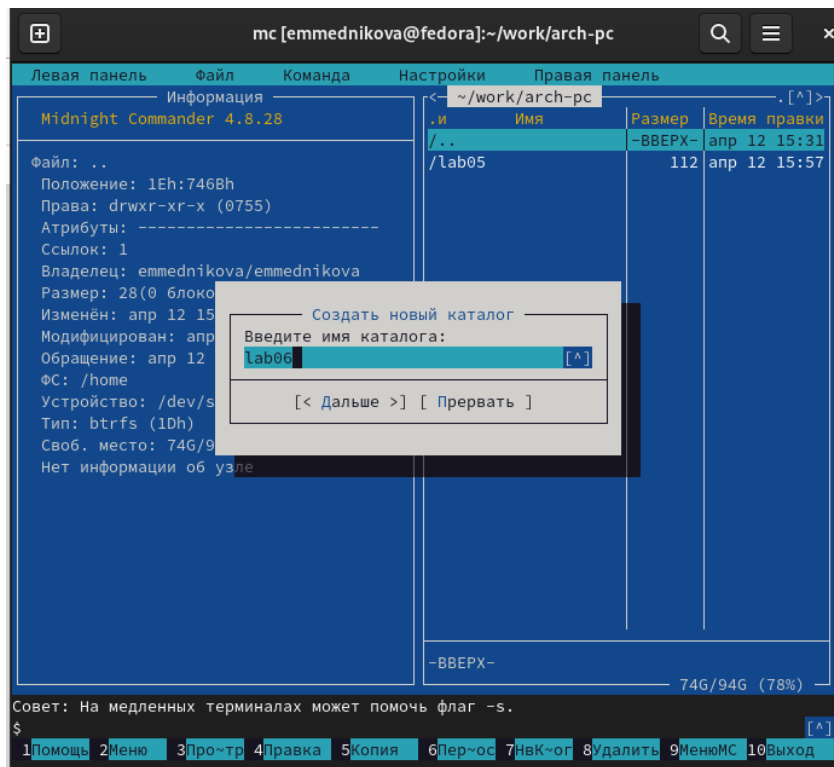


Рис. 3.2: Создание папки

3. Пользуясь строкой ввода и командой touch создала файл lab6-1.asm.

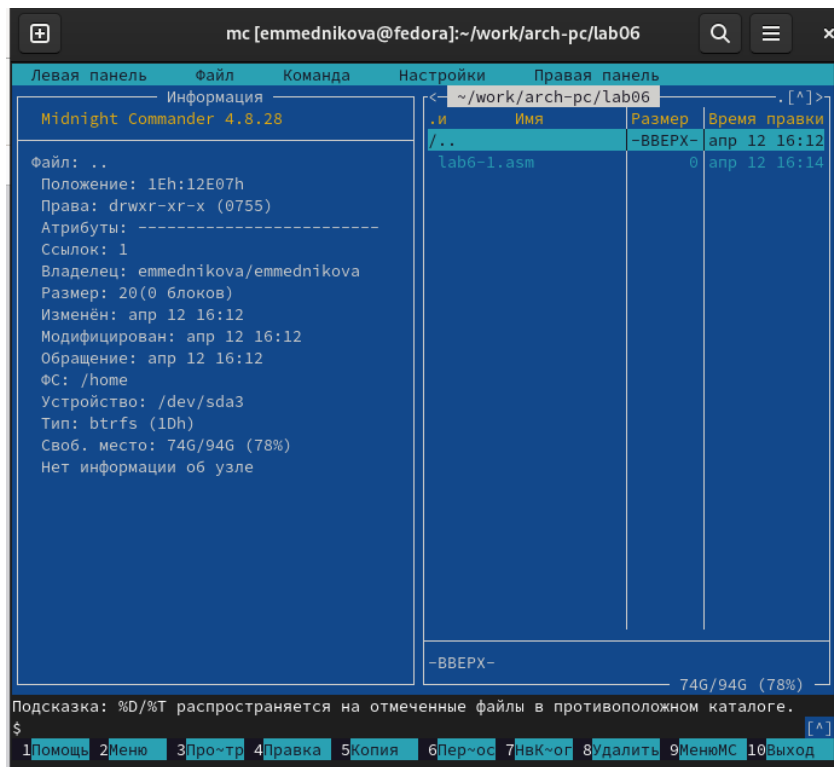


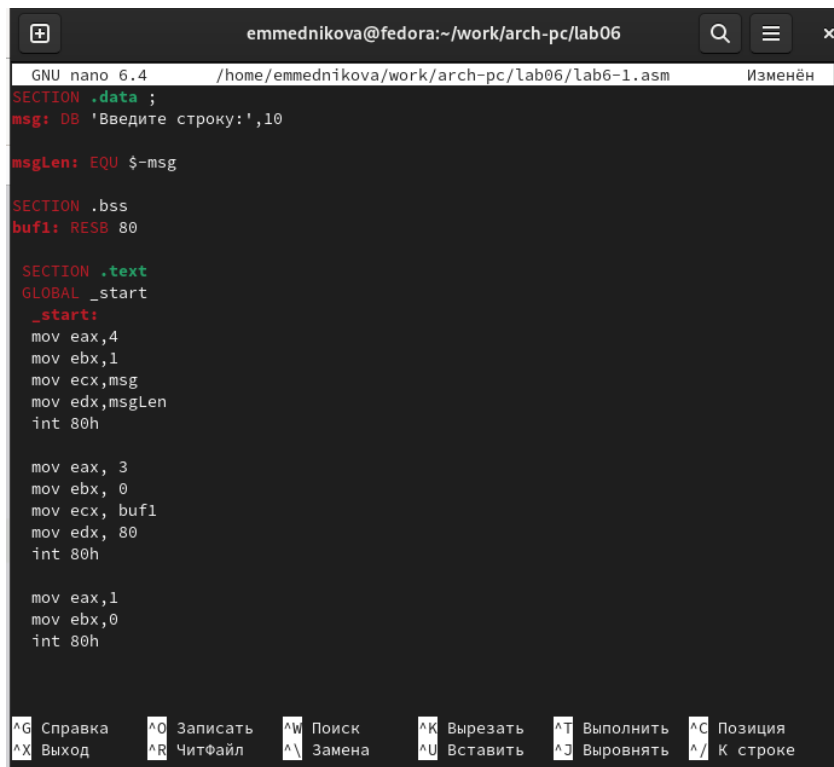
Рис. 3.3: Создание файла

4. С помощью функциональной клавиши F4 открыла файл lab6-1.asm для редактирования во встроенном редакторе.



Рис. 3.4: Встроенный редактор

5. Ввела текст программы из листинга 6.1, сохранила изменения и закрыла файл.



```
GNU nano 6.4 /home/emmednikova/work/arch-pc/lab06/lab6-1.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

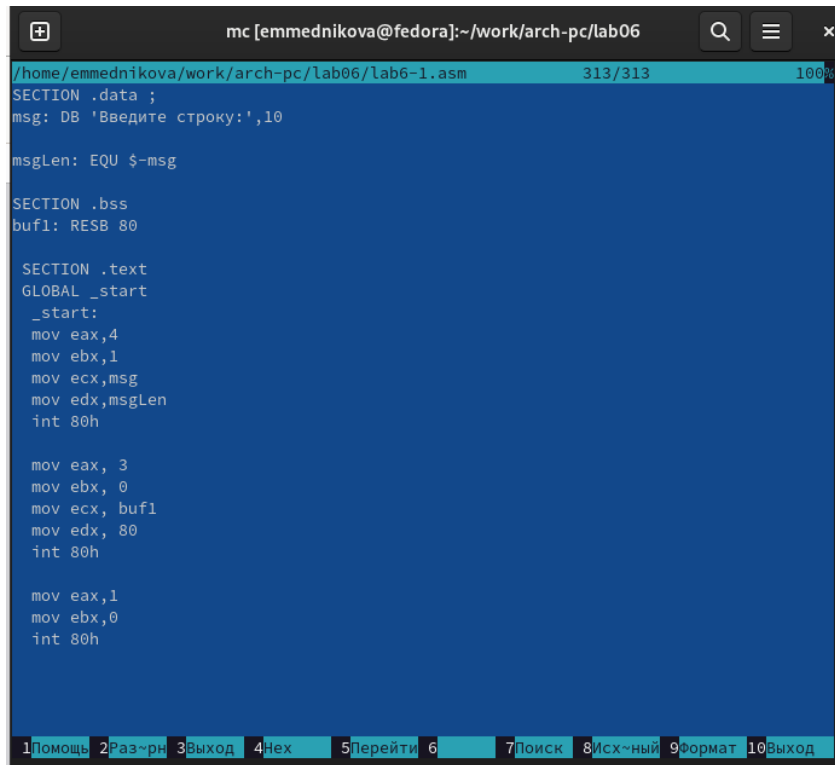
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

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

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^_ К строке

Рис. 3.5: Ввод текста

6. С помощью функциональной клавиши F3 открыла файл lab6-1.asm для просмотра. Убедилась, что файл содержит текст программы.



```
mc [emmednikova@fedora]:~/work/arch-pc/lab06
/home/emmednikova/work/arch-pc/lab06/lab6-1.asm 313/313 100%
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

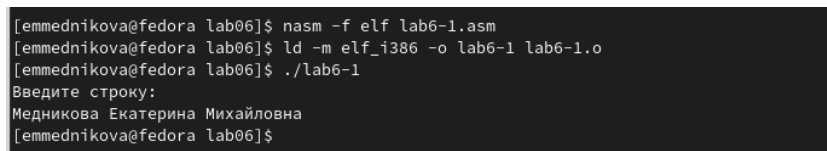
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

    mov eax,1
    mov ebx,0
    int 80h

1Помощь 2Раз-рн 3Выход 4Нех 5Перейти 6 7Поиск 8Исх-ный 9Формат 10Выход
```

Рис. 3.6: Открытие файла для просмотра

- Оттранслировала текст программы lab6-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл.



```
[emmednikova@fedora lab06]$ nasm -f elf lab6-1.asm
[emmednikova@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[emmednikova@fedora lab06]$ ./lab6-1
Введите строку:
Медникова Екатерина Михайловна
[emmednikova@fedora lab06]$
```

Рис. 3.7: Компоновка файла

- Скачала файл in_out.asm со страницы ТУИС.

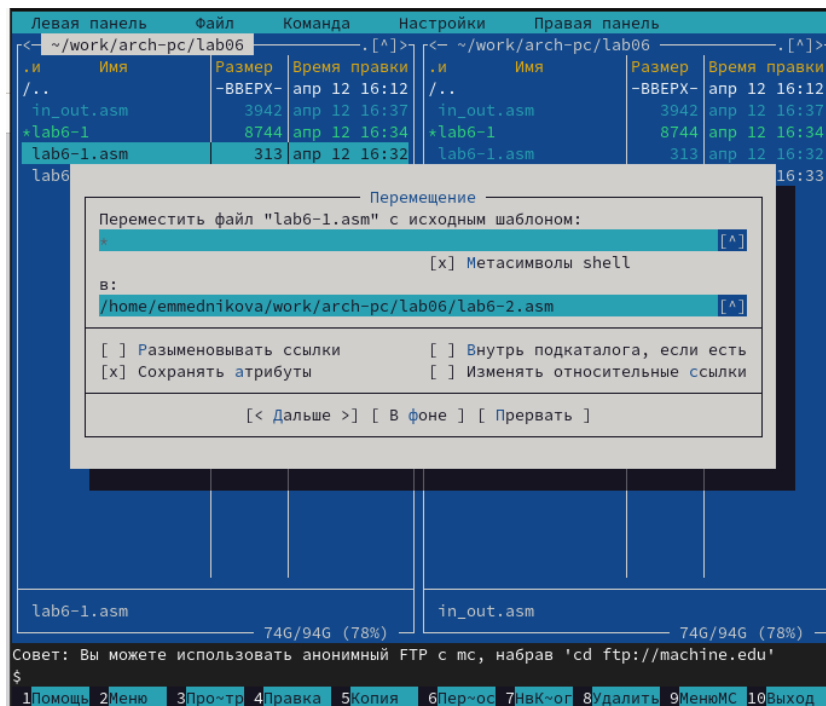


Рис. 3.10: Создание копии файла

11. Исправила текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 6.2. Создала исполняемый файл и проверила его работу.


```
GNU nano 6.4 /home/emmednikova/work/arch-pc/lab06/lab6-2.asm Изменён
#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 sprintLF

mov ecx, buf1
mov edx, 80
call sread

call quit

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Рис. 3.11: Исправление текста программы

```
[emmednikova@fedora lab06]$ nasm -f elf lab6-2.asm
[emmednikova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[emmednikova@fedora lab06]$ ./lab6-2
Введите строку:
Медникова Екатерина Михайловна
[emmednikova@fedora lab06]$
```

Рис. 3.12: Проверка работы

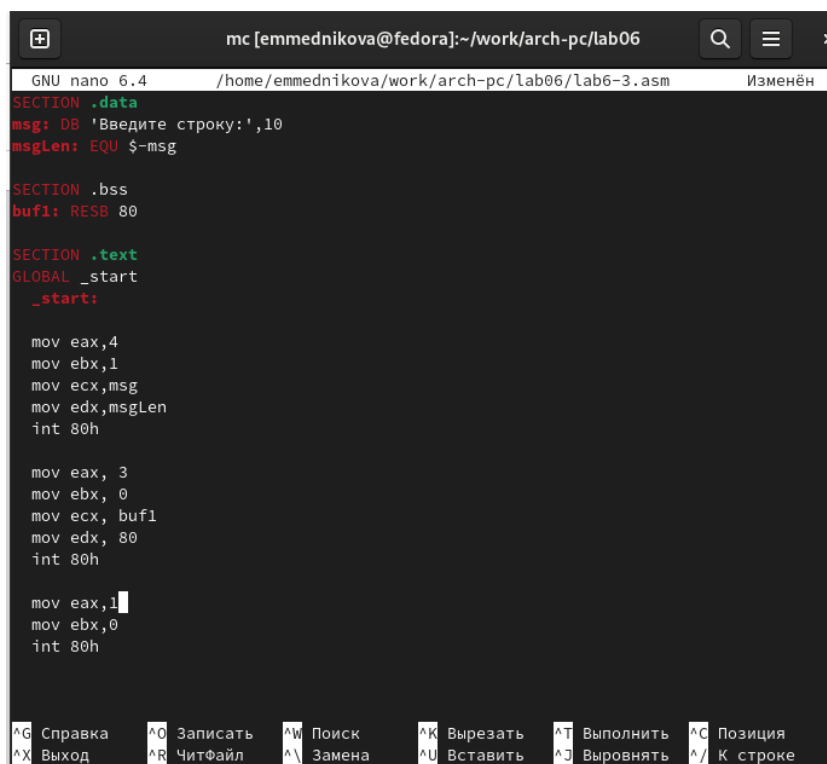
12. В файле lab6-2.asm заменила подпрограмму sprintLF на sprint. Создала исполняемый файл и проверила его работу.

```
[emmednikova@fedora lab06]$ nasm -f elf lab6-2.asm
[emmednikova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[emmednikova@fedora lab06]$ ./lab6-2
Введите строку: Медникова Екатерина Михайловна
[emmednikova@fedora lab06]$
```

Рис. 3.13: Проверка работы

4 Самостоятельная работа

1. Создала копию файла lab6-1.asm. Внесла изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по алгоритму, прописанному в задании.



```
GNU nano 6.4 /home/emmednikova/work/arch-pc/lab06/lab6-3.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

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выводить ^/_ К строке
```

Рис. 4.1: Копия файла и изменения в программе

2. Получила исполняемый файл и проверила его работу. На приглашение ввести строку ввела свои ФИО.

```
[emmednikova@fedora lab06]$ nasm -f elf lab6-3.asm
[emmednikova@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[emmednikova@fedora lab06]$ ./lab6-3
Введите строку:
Медникова Екатерина Михайловна
[emmednikova@fedora lab06]$
```

Рис. 4.2: Проверка работы

3. Создала копию файла lab6-2.asm. Исправила текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по алгоритму, прописанному в задании.

```
GNU nano 6.4 /home/emmednikova/work/arch-pc/lab06/lab6-4.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

[ Прочитано 19 строк ]
^G Справка ^O Записать ^M Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^_ Замена ^U Вставить ^J Выводить ^/_ К строке
```

Рис. 4.3: Копия файла и правка в программе

4. Создала исполняемый файл и проверила его работу.

```
[emmednikova@fedora lab06]$ nasm -f elf lab6-4.asm
[emmednikova@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[emmednikova@fedora lab06]$ ./lab6-4
Введите строку: Медникова
[emmednikova@fedora lab06]$
```

Рис. 4.4: Создание файла и его проверка

5 Выводы

Приобрела практические навыки работы в Midnight Commander. Освоила инструкции языка ассемблера `mov` и `int`.