

Отчет по лабораторной работе №5

Основы работы с Midnight Commander

Новикова Анастасия Андреевна

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 3.1 | Основы работы с mc | 7 |
| 3.2 | Структура программы на языке ассемблера NASM | 9 |
| 3.3 | Подключение внешнего файла | 12 |
| 3.4 | Выполнение заданий для самостоятельной работы | 14 |
| 4 | Выводы | 19 |

Список иллюстраций

| | | |
|------|---|----|
| 3.1 | Открытый mc | 7 |
| 3.2 | Перемещение между директориями | 8 |
| 3.3 | Создание каталога | 8 |
| 3.4 | Перемещение между директориями | 9 |
| 3.5 | Создание файла | 9 |
| 3.6 | Открытие файла для редактирования | 10 |
| 3.7 | Редактирование файла | 10 |
| 3.8 | Открытие файла для просмотра | 11 |
| 3.9 | Трансляция в объектный файл | 11 |
| 3.10 | Проверка создания объектного файла | 11 |
| 3.11 | Компиляция файла и передача на обработку компоновщику | 11 |
| 3.12 | Исполняемый файл | 12 |
| 3.13 | Исполнение файла | 12 |
| 3.14 | Копирование файла | 12 |
| 3.15 | Копирование файла | 13 |
| 3.16 | Редактирование файла | 13 |
| 3.17 | Исполнение файла | 13 |
| 3.18 | Исполнение файла | 14 |
| 3.19 | Копирование файла | 14 |
| 3.20 | Редактирование файла | 15 |
| 3.21 | Исполнение файла | 15 |
| 3.22 | Копирование файла | 17 |
| 3.23 | Редактирование файла | 17 |
| 3.24 | Исполнение файла | 18 |

Список таблиц

1 Цель работы

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

2 Задание

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

Открываю Midnight Commander, введя в терминал mc (рис. 3.1).

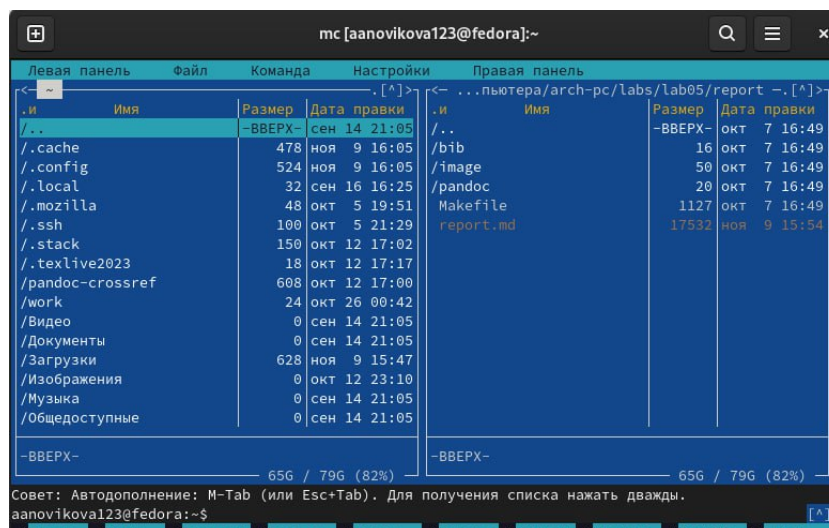


Рис. 3.1: Открытый mc

Перехожу в каталог ~/work/arch-рс, используя файловый менеджер mc (рис. 3.2)

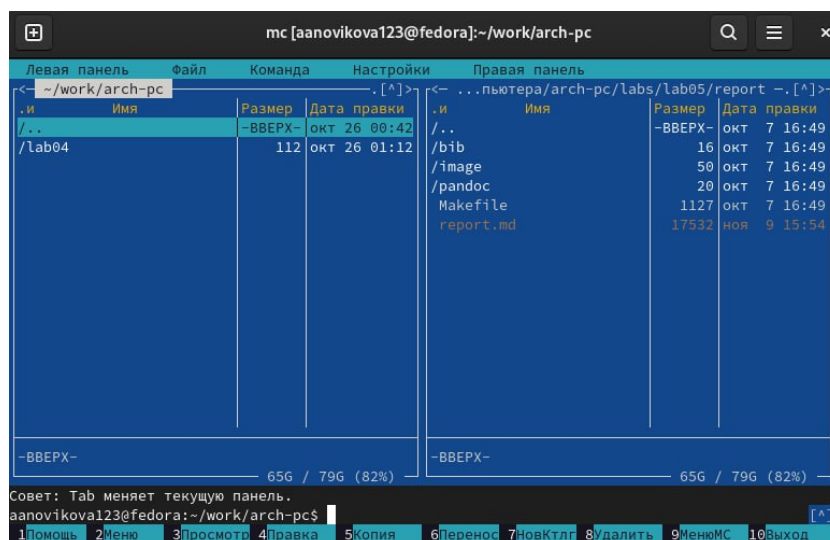


Рис. 3.2: Перемещение между директориями

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. 3.3).

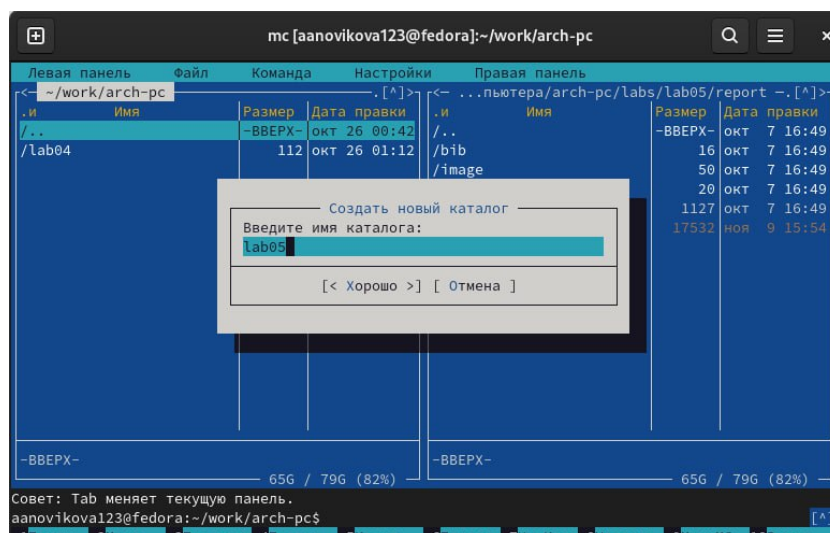


Рис. 3.3: Создание каталога

Переходу в созданный каталог (рис. 3.4).

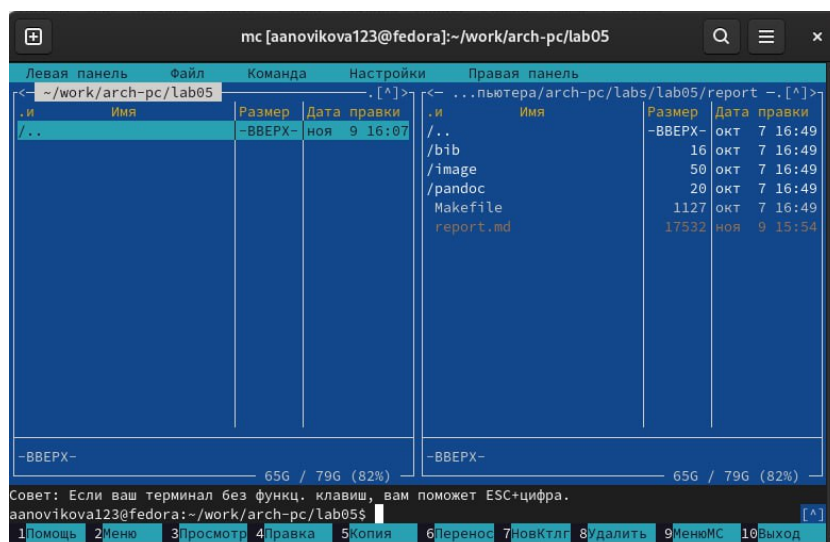


Рис. 3.4: Перемещение между директориями

В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать (рис. 3.5).

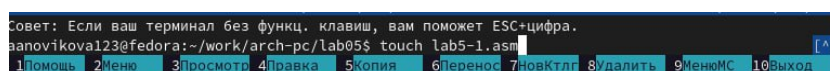


Рис. 3.5: Создание файла

3.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе mcedit (рис. 3.6).

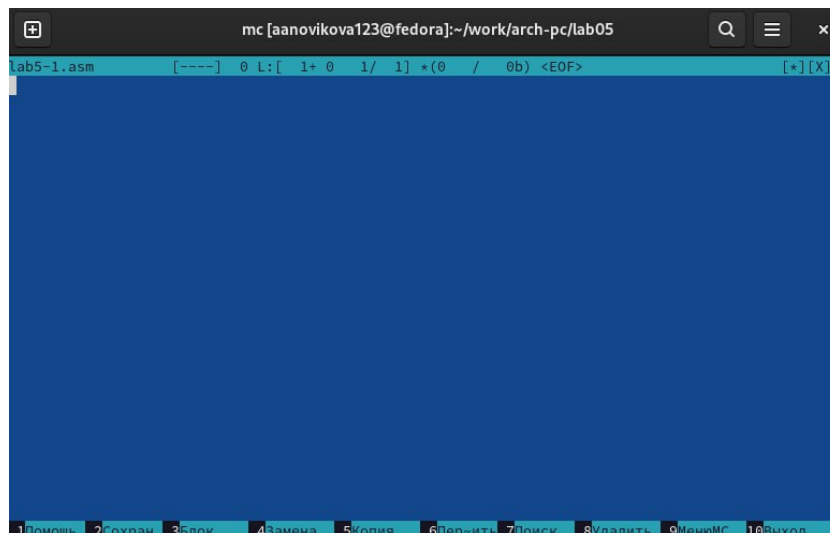


Рис. 3.6: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 3.7).
Далее клавишей F2 сохраняю изменения и выхожу из файла (клавиша F10).

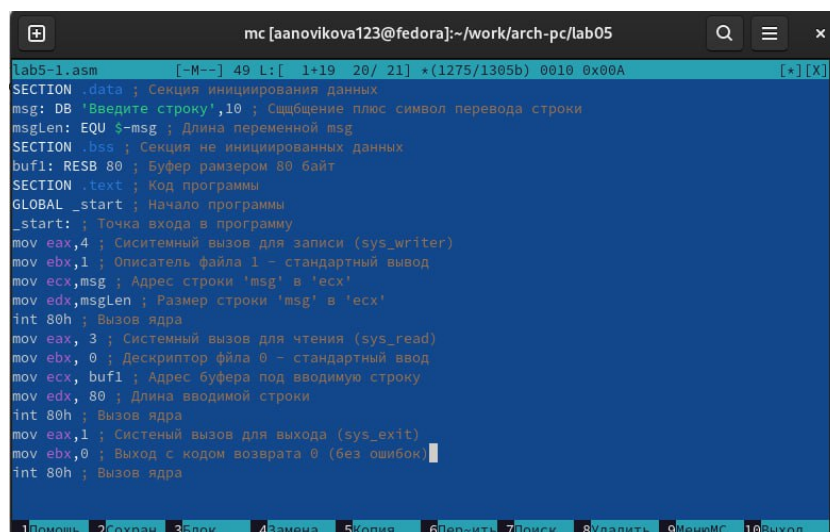


Рис. 3.7: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 3.8).

```

/home/aanovikova123/work/arch-pc/lab05/lab5-1.asm 1305/1305 100%
SECTION .data ; Секция инициализации данных
msg: DB 'Введите строку',10 ; Символ перевода строки
msgLen: EQU $-msg ; Длина переменной msg
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,4 ; Системный вызов для записи (sys_writer)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'ecx'
int 80h ; Вызов ядра
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
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
1Помощь 2Разверн 3Выход 4Чех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

```

Рис. 3.8: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. (рис. 3.9) Создался объектный файл `lab5-1.o`. (рис. 3.10)

```

Совет: Если ваш терминал без функц. клавиш, вам поможет ESC+цифра.
aanovikova123@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

```

Рис. 3.9: Трансляция в объектный файл

| Левая панель | | Правая панель | |
|--------------|--------|---------------|---------|
| Имя | Размер | Имя | Размер |
| lab5-1.asm | 1305 | .. | -BВЕРХ- |
| lab5-1.o | 752 | ./bib | 16 |
| | | ./image | 50 |
| | | ./pandoc | 20 |
| | | Makefile | 1127 |
| | | report.md | 17532 |

Рис. 3.10: Проверка создания объектного файла

Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o` (рис. 3.11). Создался исполняемый файл `lab5-1`. (рис. 3.12)

```

Совет: Если ваш терминал без функц. клавиш, вам поможет ESC+цифра.
aanovikova123@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

```

Рис. 3.11: Компиляция файла и передача на обработку компоновщику

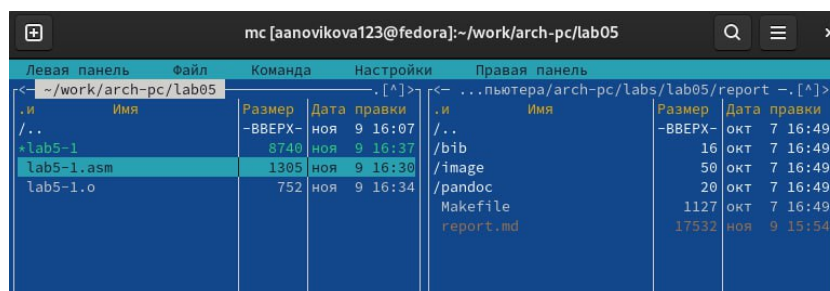


Рис. 3.12: Исполняемый файл

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 3.13).

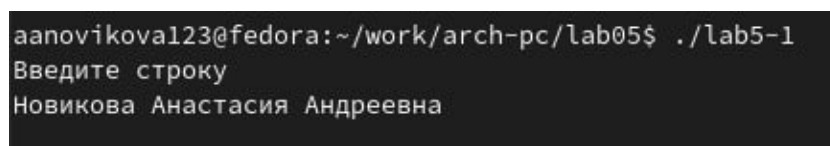


Рис. 3.13: Исполнение файла

3.3 Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки”. С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 3.14).

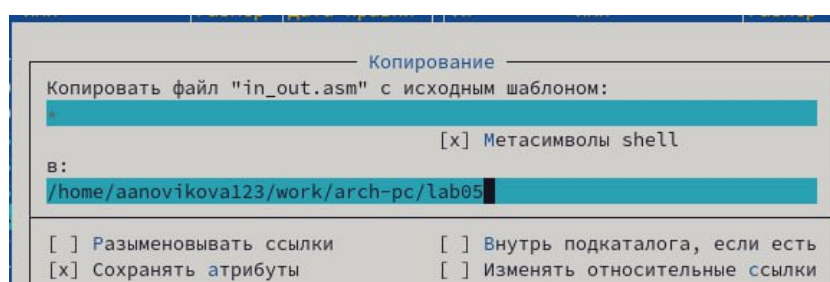


Рис. 3.14: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для

копии файла (рис. 3.15).

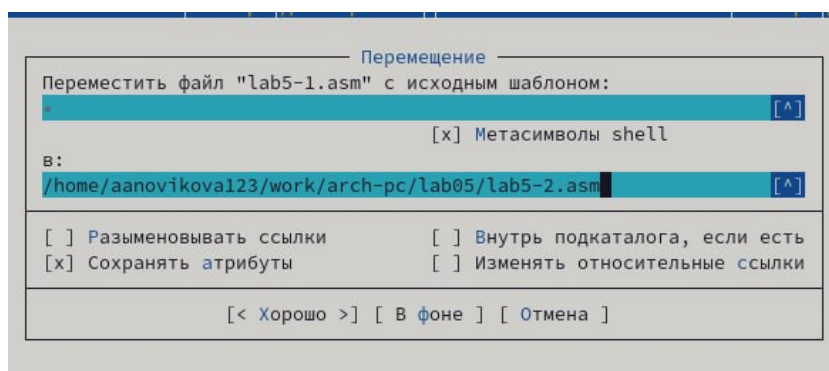


Рис. 3.15: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе mcedit (рис. 3.16), чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.

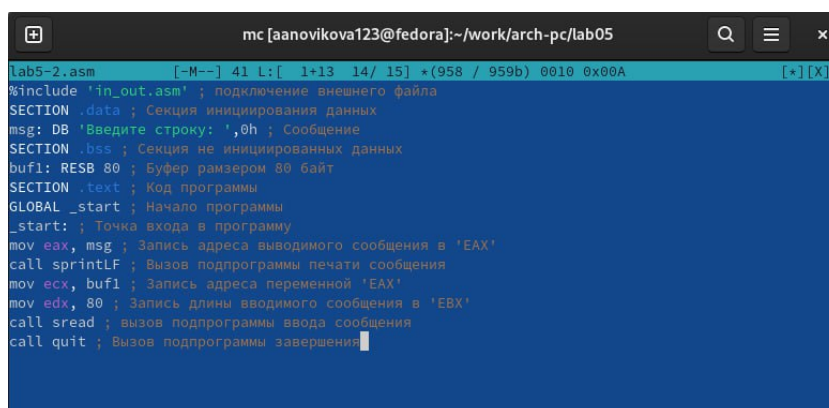


Рис. 3.16: Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл lab5-2.o. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o` Создался исполняемый файл lab5-2. Запускаю исполняемый файл (рис. 3.17).

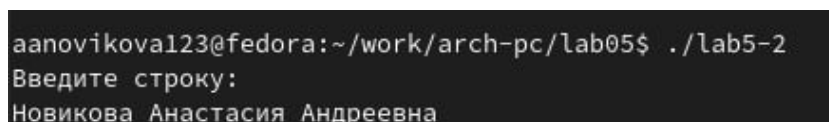


Рис. 3.17: Исполнение файла

Открываю файл lab5-2.asm для редактирования в mcedit функциональной клавишей F4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения.

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 3.18).

```
aanovikova123@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aanovikova123@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab5-2.o
aanovikova123@fedora:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку: Новикова Анастасия Андреевна
```

Рис. 3.18: Исполнение файла

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами sprintLF и sprint.

3.4 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 3.19).

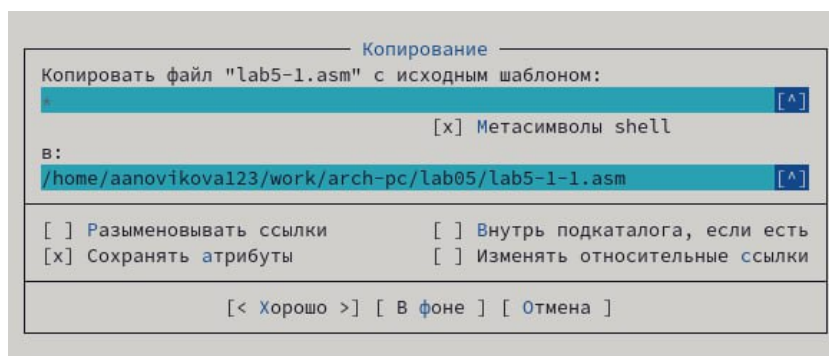
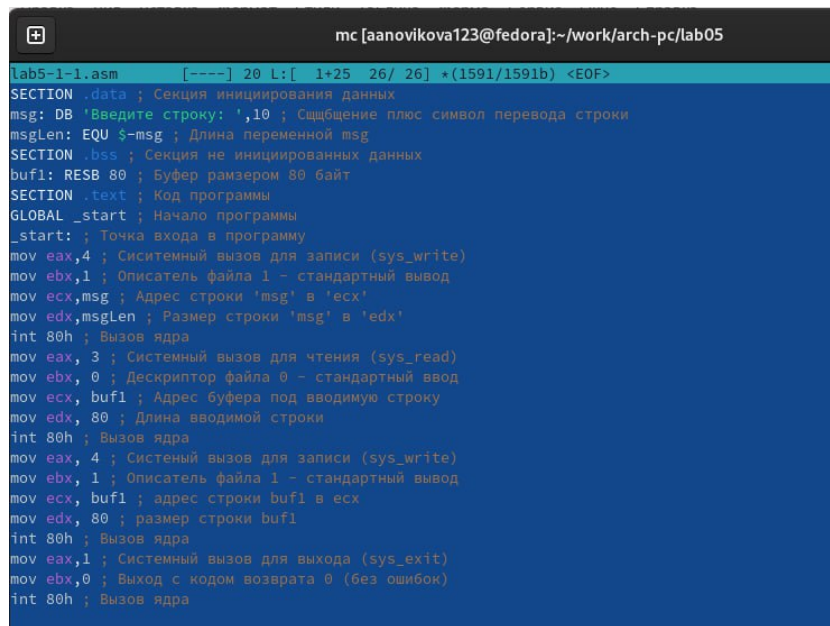


Рис. 3.19: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и

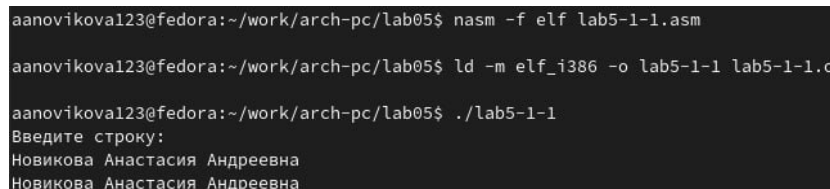
запроса ввода, она выводила вводимую пользователем строку (рис. 3.20).



```
mc [aanovikova123@fedora]:~/work/arch-pc/lab05
lab5-1-1.asm  [----] 20 L: [ 1+25 26/ 26] *(1591/1591b) <EOF>
SECTION .data ; Секция инициализации 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
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,80 ; 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
```

Рис. 3.20: Редактирование файла

Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.21).



```
aanovikova123@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
aanovikova123@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
aanovikova123@fedora:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Новикова Анастасия Андреевна
Новикова Анастасия Андреевна
```

Рис. 3.21: Исполнение файла

Код программы:

```
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10
msgLen: EQU $-msg ; Длина переменной 'msg'
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,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
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 ; Описание файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov buf1, 80 ; 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

```

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 3.22).

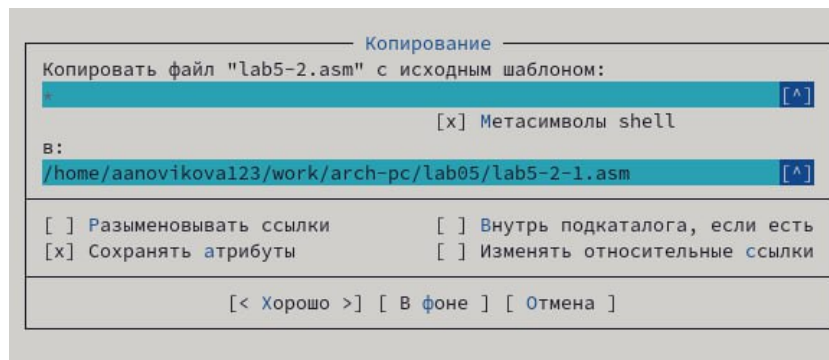


Рис. 3.22: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 3.23).

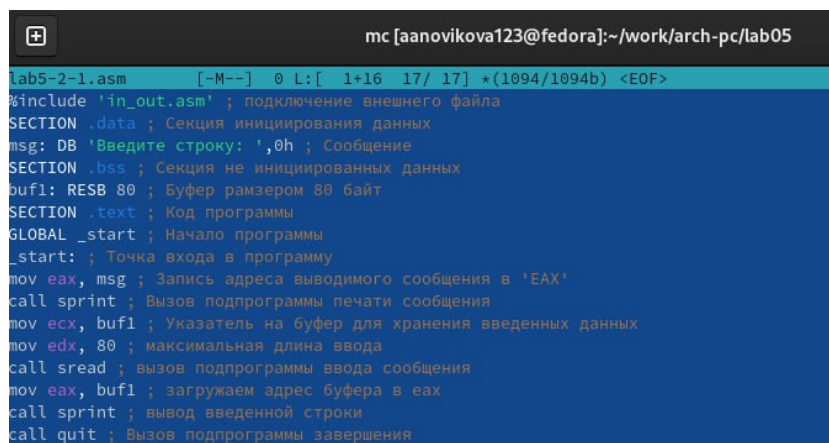


Рис. 3.23: Редактирование файла

Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.24).

```

aanovikova123@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
aanovikova123@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
aanovikova123@fedora:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку: Новикова Анастасия Андреевна
Новикова Анастасия Андреевна

```

Рис. 3.24: Исполнение файла

Код программы:

```

%include 'in_out.asm'

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 ; указатель на буфер для хранения введенных данных
mov edx, 80 ; максимальная длина ввода
call sread ; вызов подпрограммы ввода сообщения

mov eax, buf1; загружаем адрес буфера в eax
call sprint ; вывод введенной строки
call quit ; вызов подпрограммы завершения

```

4 Выводы

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