

Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютера

Бережной Иван Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Основы работы с тс	7
3.2	Работа с языком ассемблера NASM	9
3.3	Подключение внешнего файла	12
3.4	Задание для самостоятельной работы	14
4	Вывод	20
	Список литературы	21

Список иллюстраций

3.1	Открытие Midnight Commander	8
3.2	Создание каталога lab05	8
3.3	Создание файла lab5-1.asm	9
3.4	Изменение lab5-1.asm	10
3.5	Просмотр lab5-1.asm	11
3.6	Создание исполняемого файла lab5-1.exe	11
3.7	Запуск исполняемого файла lab5-1.exe	12
3.8	Копирование in_out.asm	12
3.9	Создание и редактирование lab5-2.asm	13
3.10	Проверка работы lab5-2.exe	13
3.11	Запуск изменённого файла lab5-2.exe	14
3.12	Создание файла lab5-3.asm	14
3.13	Редактирование lab5-3.asm	15
3.14	Запуск lab5-3.exe	15
3.15	Создание файла lab5-4.asm	17
3.16	Редактирование lab5-4.asm	17
3.17	Запуск lab5-4.exe	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 тс
2. Работа с языком ассемблера NASM
3. Подключение внешнего файла
4. Задание для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Основы работы с tc

Откроем Midnight Commander командой `tc` и перейдём в каталог `~/work/arch-rc`, который мы создали в ходе выполнения лабораторной работы №4 (рис. 3.1).

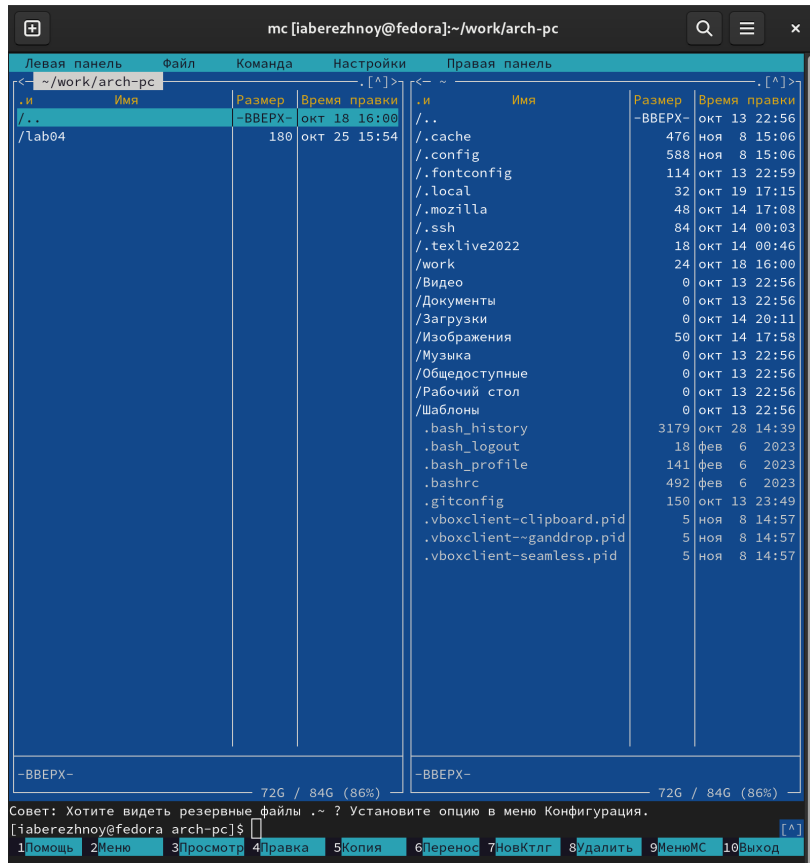


Рис. 3.1: Открытие Midnight Commander

Создадим папку lab05, в которой и будем работать в дальнейшем (рис. 3.2). Перейдём в неё.

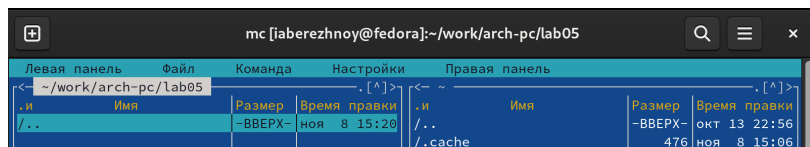


Рис. 3.2: Создание каталога lab05

С помощью команды touch создаём файл lab5-1.asm (рис. 3.3).

3.2 Работа с языком ассемблера NASM

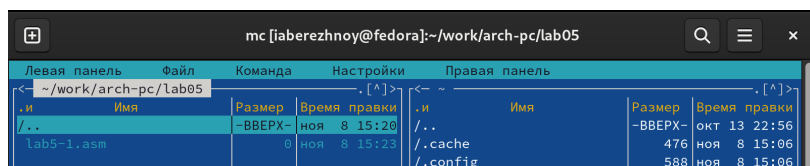
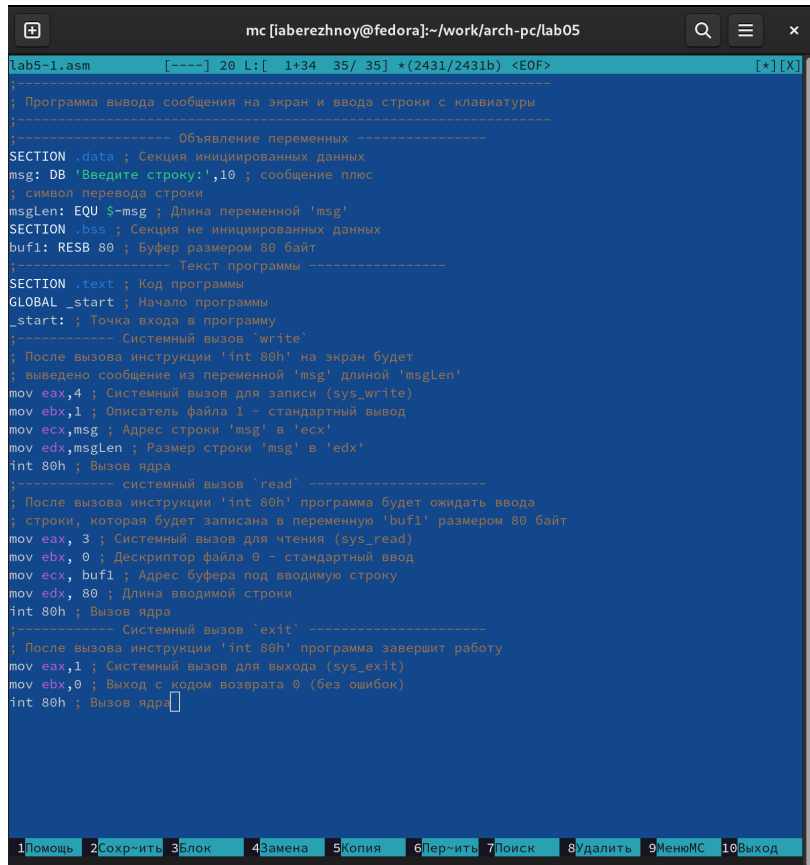


Рис. 3.3: Создание файла lab5-1.asm

Откроем lab5-1.asm, используя функциональную клавишу F4. Мы попали в редактор mcedit. Введём текст программы из предложенного листинга и сохраним изменения (рис. 3.4).



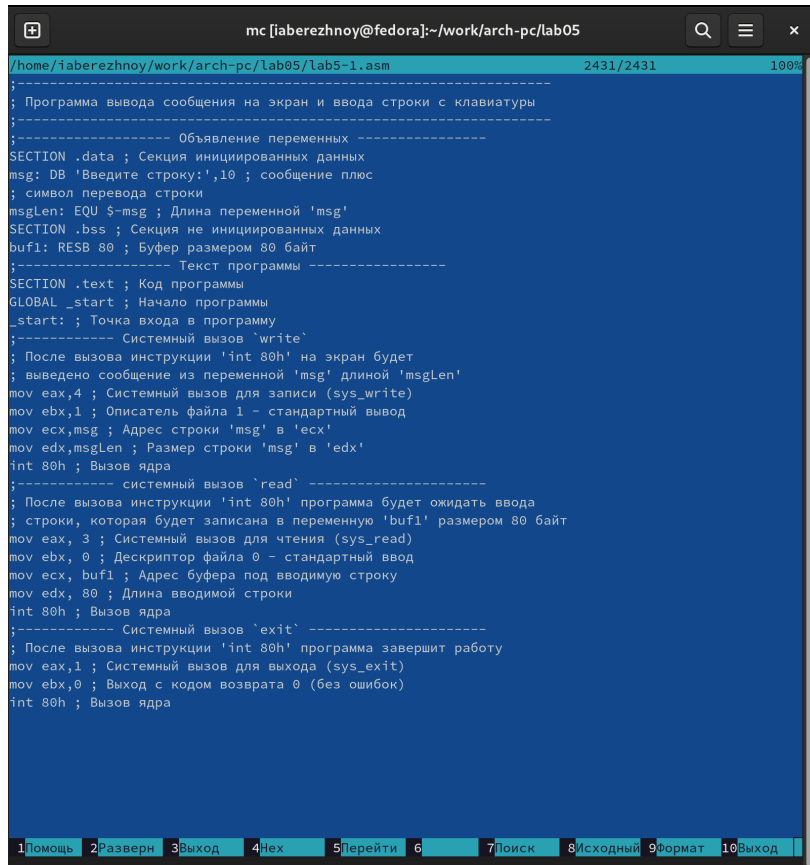
```
lab5-1.asm [-----] 20 L: [ 1+34 35/ 35] *(2431/2431b) <EOF> [*][X]

; 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 ;Descriptor 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. 3.4: Изменение lab5-1.asm

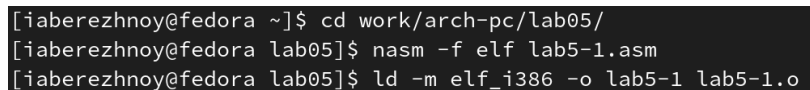
Откроем файл для просмотра и убедимся, что все изменения сохранились корректно (рис. 3.5).



```
mc [iaberezhnoy@fedora]:~/work/arch-pc/lab05
/home/iaberezhnoy/work/arch-pc/lab05/lab5-1.asm 2431/2431 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 ; 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
1Помощь 2Разверн 3Выход 4Hex 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 3.5: Просмотр lab5-1.asm

Оттранслируем текст программы в объектный файл, затем выполним его компоновку (рис. 3.6) и запустим новый исполняемый файл. Подаём на ввод строку “Бережной Иван Александрович” (рис. 3.7).



```
[iaberezhnoy@fedora ~]$ cd work/arch-pc/lab05/
[iaberezhnoy@fedora lab05]$ nasm -f elf lab5-1.asm
[iaberezhnoy@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Рис. 3.6: Создание исполняемого файла lab5-1.exe

```
[iaberezhnoy@fedora lab05]$ ./lab5-1
Введите строку:
Бережной Иван Алекснадрович
```

Рис. 3.7: Запуск исполняемого файла lab5-1.exe

3.3 Подключение внешнего файла

Скачаем файл `in_out.asm` и переместим его в каталог, в котором мы сейчас работаем. Для этого скопируем его из загрузок в `~/work/arch-pc/lab05` (рис. 3.8).

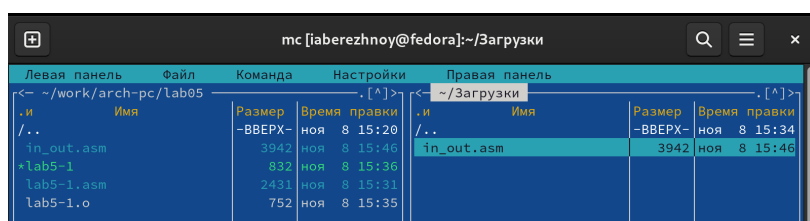
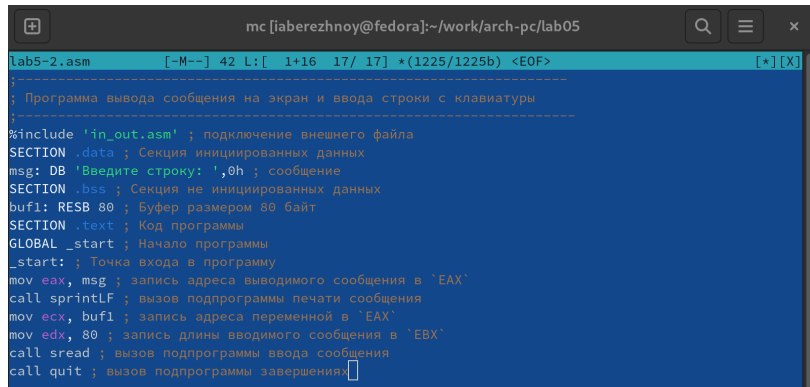


Рис. 3.8: Копирование `in_out.asm`

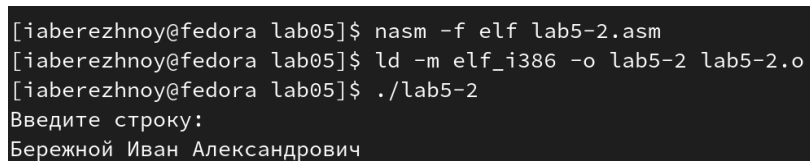
Создадим копию файла `lab5-1.asm`, назвав `lab5-2.asm`. Используя предложенный листинг, изменим текст программы в созданном файле с использованием подпрограмм из внешнего файла `in_out.asm` (рис. 3.9).



```
mc [iaberezhnoy@fedora]:~/work/arch-pc/lab05
lab5-2.asm [-M--] 42 L: [ 1+16 17/ 17] *(1225/1225b) <EOF> [*][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 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.9: Создание и редактирование lab5-2.asm

Проверим работу файла. Для этого оттранслируем объектный файл, выполним его компоновку и запустим исполняемый файл. Видим запрос на ввод строки. Введём “Бережной Иван Александрович” (рис. 3.10).



```
[iaberezhnoy@fedora lab05]$ nasm -f elf lab5-2.asm
[iaberezhnoy@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[iaberezhnoy@fedora lab05]$ ./lab5-2
Введите строку:
Бережной Иван Александрович
```

Рис. 3.10: Проверка работы lab5-2.exe

Немного изменим файл lab5-2.asm, заменив sprintf на sprintf. Снова создадим исполняемый файл по той же схеме и запустим его. Видим, что запрос на ввод строки находится на той же строке, что и приглашение типа “Введите строку:” (рис. 3.11). В предыдущей версии программы запрос на ввод был на следующей строке после приглашения.

```
[iaberezhnoy@fedora lab05]$ nasm -f elf lab5-2.asm
[iaberezhnoy@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[iaberezhnoy@fedora lab05]$ ./lab5-2
Введите строку: Бережной Иван Александрович
```

Рис. 3.11: Запуск изменённого файла lab5-2.exe

3.4 Задание для самостоятельной работы

Создадим копию файла lab5-1.asm, назвав lab5-3.asm (рис. 3.12). Редактируем файл так, чтобы программа выводила вводимую пользователем строку (так же изменим приглашение на “Введите свою фамилию:”) (рис. 3.13).

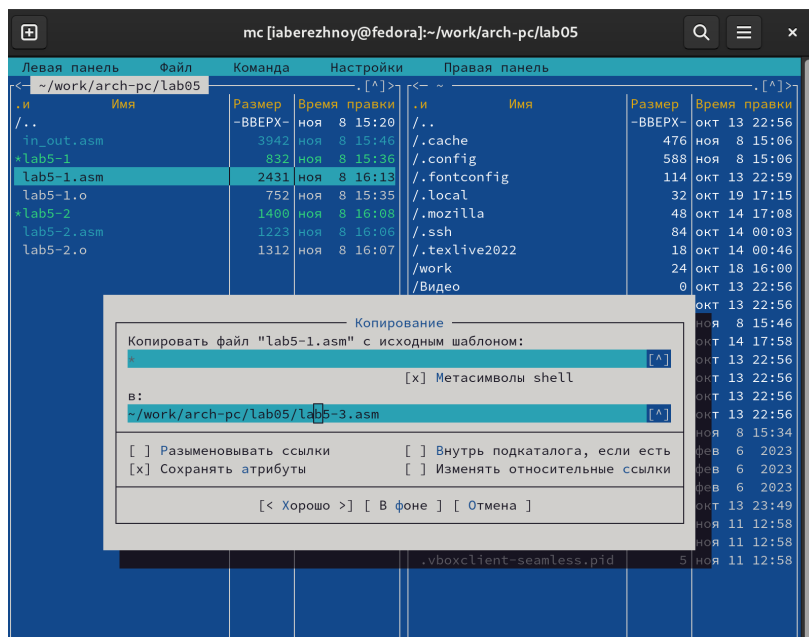
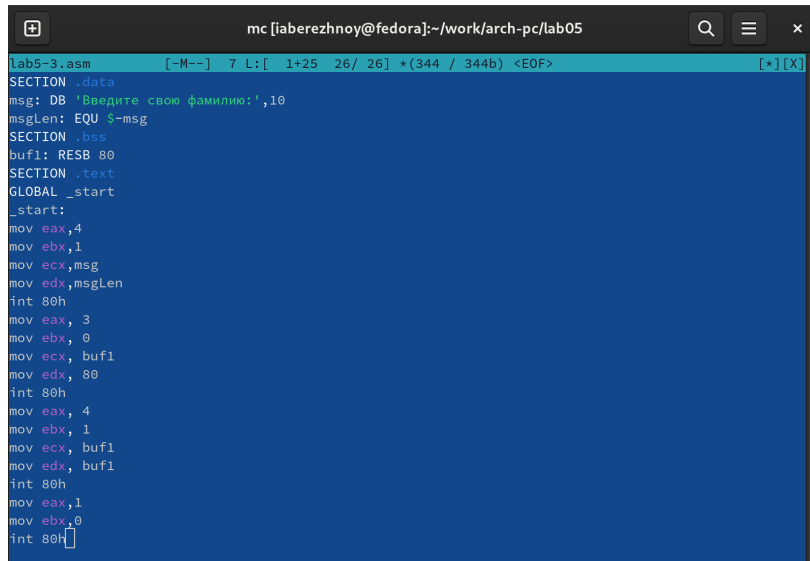


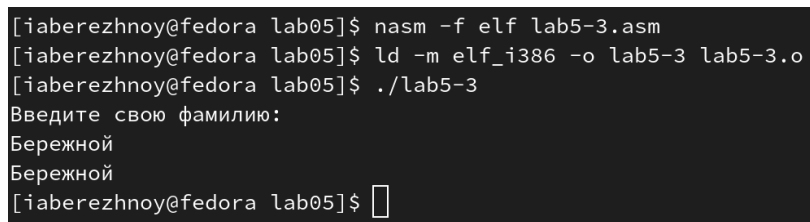
Рис. 3.12: Создание файла lab5-3.asm



```
lab5-3.asm [-M--] 7 L: [ 1+25 26/ 26] *(344 / 344b) <EOF> [*] [X]
SECTION .data
msg: DB 'Введите свою фамилию:',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
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 3.13: Редактирование lab5-3.asm

Теперь создадим исполняемый файл по знакомой нам схеме и запустим его (рис. 3.14). Всё работает корректно.



```
[iaberezhnoy@fedora lab05]$ nasm -f elf lab5-3.asm
[iaberezhnoy@fedora lab05]$ ld -m elf_i386 -o lab5-3 lab5-3.o
[iaberezhnoy@fedora lab05]$ ./lab5-3
Введите свою фамилию:
Бережной
Бережной
[iaberezhnoy@fedora lab05]$
```

Рис. 3.14: Запуск lab5-3.exe

Листинг 5.1

```
SECTION .data
msg: DB 'Введите свою фамилию:',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
mov edx, buf1
int 80h
mov eax,1
mov ebx,0
int 80h

```

Создадим копию файла lab5-2.asm, назвав lab5-4.asm (рис. 3.15). Редактируем файл так, чтобы он выполнял те же функции, что и lab5-3.asm

(рис. 3.16).

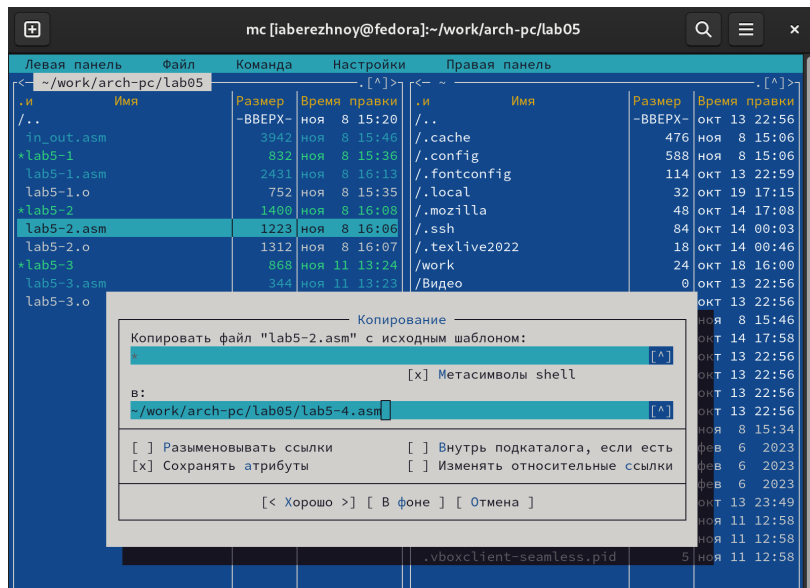


Рис. 3.15: Создание файла lab5-4.asm

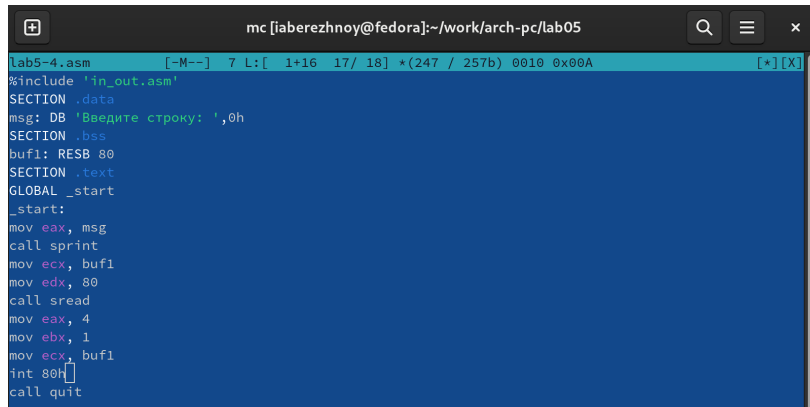


Рис. 3.16: Редактирование lab5-4.asm

Теперь создадим исполняемый файл по знакомой нам схеме и запустим его (рис. 3.17). Всё работает корректно.

```
[iaberezhnoy@fedora lab05]$ nasm -f elf lab5-4.asm
[iaberezhnoy@fedora lab05]$ ld -m elf_i386 -o lab5-4 lab5-4.o
[iaberezhnoy@fedora lab05]$ ./lab5-4
Введите строку: Бережной Иван
Бережной Иван
```

Рис. 3.17: Запуск lab5-4.exe

Листинг 5.2

```
%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
mov eax, 4
mov ebx, 1
mov ecx, buf1
```

```
int 80h  
call quit
```

4 Вывод

В ходе выполнения лабораторной работы мы приобрели практические навыки работы в Midnight Commander и освоили инструкции языка ассемблера `mov` и `int`.

Список литературы

::: Архитектура ЭВМ