

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	12
4	Выводы	17

Список иллюстраций

2.1	Midnight Commander	6
2.2	Переход в каталог	7
2.3	Создание папки	7
2.4	Создание файла	8
2.5	Открытие файла	8
2.6	Ввод текста программы	9
2.7	Проверка	9
2.8	Запуск программы	10
2.9	Создание файла lab5-2.asm	10
2.10	Исправление текста программы	11
2.11	Проверка программы	11
2.12	Проверка программы после замены	11
3.1	Копирование файла	12
3.2	Изменение программы	12
3.3	Создание исполняемого файла	13
3.4	Проверка	13
3.5	Копирование файла	14
3.6	Изменение файла	15
3.7	Запуск программы	15

Список таблиц

1 Цель работы

Цель данной лабораторной работы - приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера `mov` и `int`.

2 Выполнение лабораторной работы

Открываю Midnight Commander (рис. 2.1).

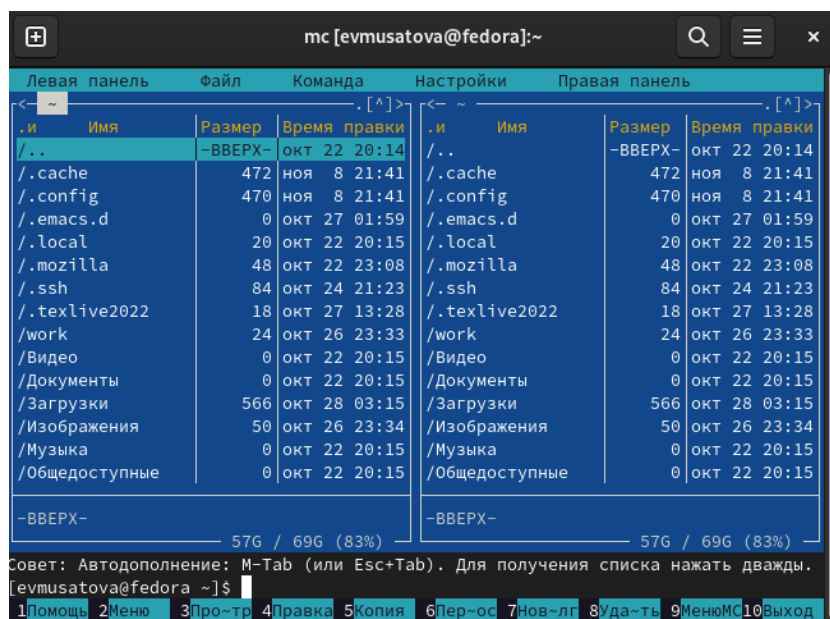


Рис. 2.1: Midnight Commander

Пользуясь нужными клавишами, перехожу в каталог arch-рс (рис. 2.2).

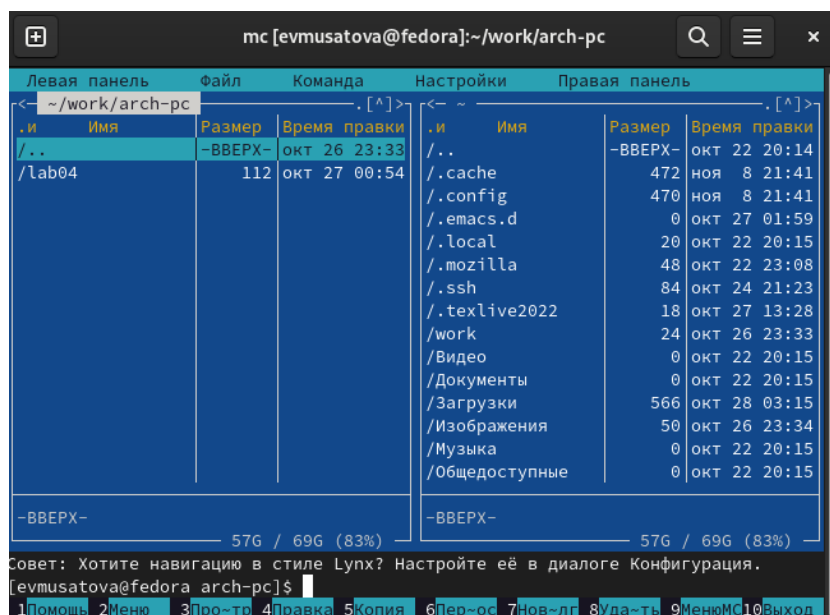


Рис. 2.2: Переход в каталог

С помощью клавиши f7 создаю папку lab05 и перехожу в созданный каталог (рис. 2.3).

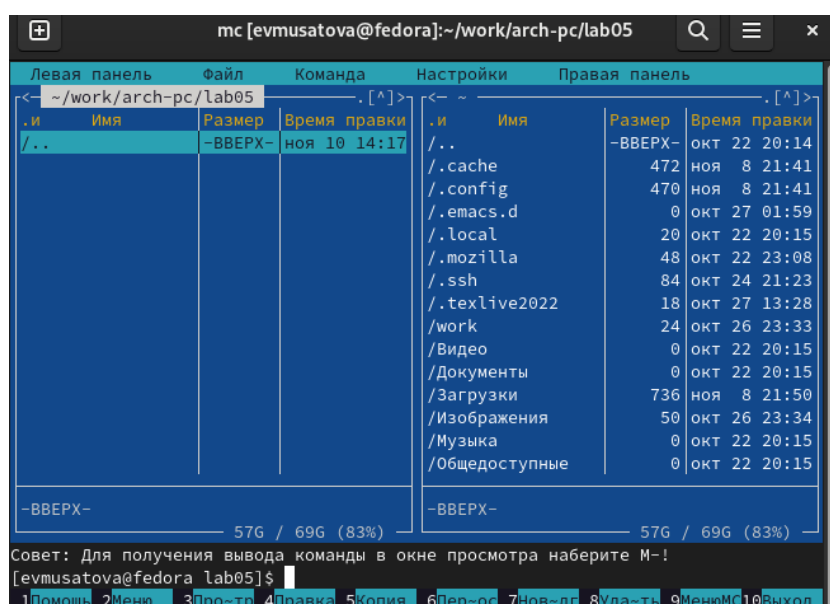


Рис. 2.3: Создание папки

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. 2.4).

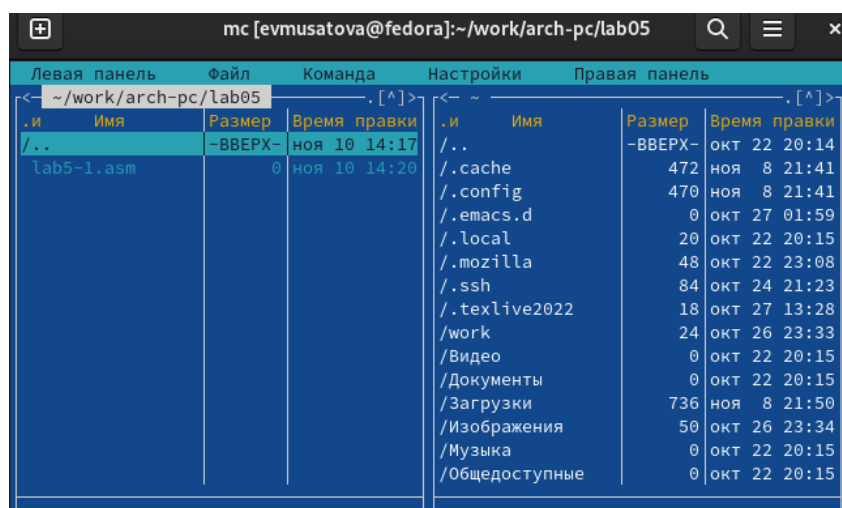


Рис. 2.4: Создание файла

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе mcedit (рис. 2.5).

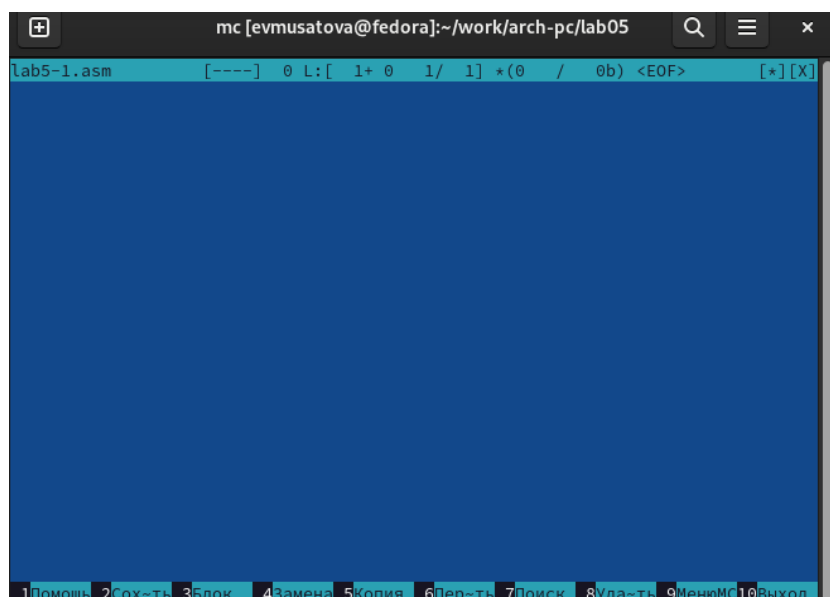


Рис. 2.5: Открытие файла

Ввожу текст программы из листинга 5.1 (рис. 2.6). Затем сохраняю изменения и закрываю файл.


```

lab5-1.asm  [---]  0 L: [ 1+ 0 1/ 35] *(0 /2431b) 0059 0x03B  [*] [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 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
1Помощь 2Скопировать 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 2.6: Ввод текста программы

С помощью функциональной клавиши f3 открываю файл, чтобы убедиться, что он содержит текст программы (рис. 2.7).

```

/home/evmusatova/work/arch-pc/lab05/lab5-1.asm  2041/2431  83%
;-----
; 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

```

Рис. 2.7: Проверка

Оттранслировала текст программы lab5-1.asm в объектный файл, выполнила компоновку объектного файла и запустила получившийся исполняемый файл, на запрос ввела свою фамилию и имя (рис. 2.8).

```
[evmusatova@fedora lab05]$ nasm -f elf hello.asm
nasm: fatal: unable to open input file 'hello.asm' No such file or directory
[evmusatova@fedora lab05]$ nasm -f elf lab5-1.asm
[evmusatova@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[evmusatova@fedora lab05]$ ./lab5-1
Введите строку:
Мусатова Екатерина
[evmusatova@fedora lab05]$
```

Рис. 2.8: Запуск программы

Скачиваю файл in_out.asm со страницы курса в ТУИС и помещаю его в тот же каталог, где находится файл с программой. Затем с помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. (рис. 2.9).

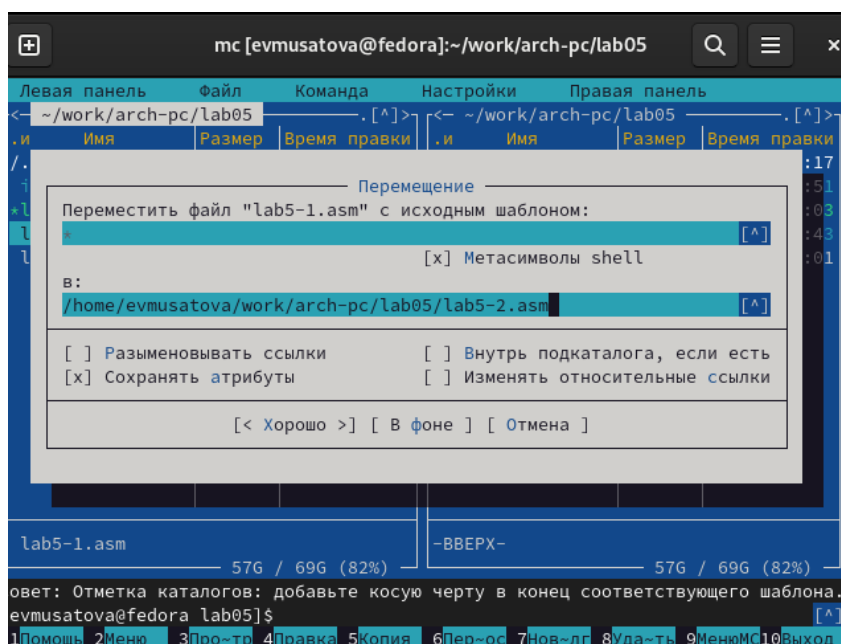
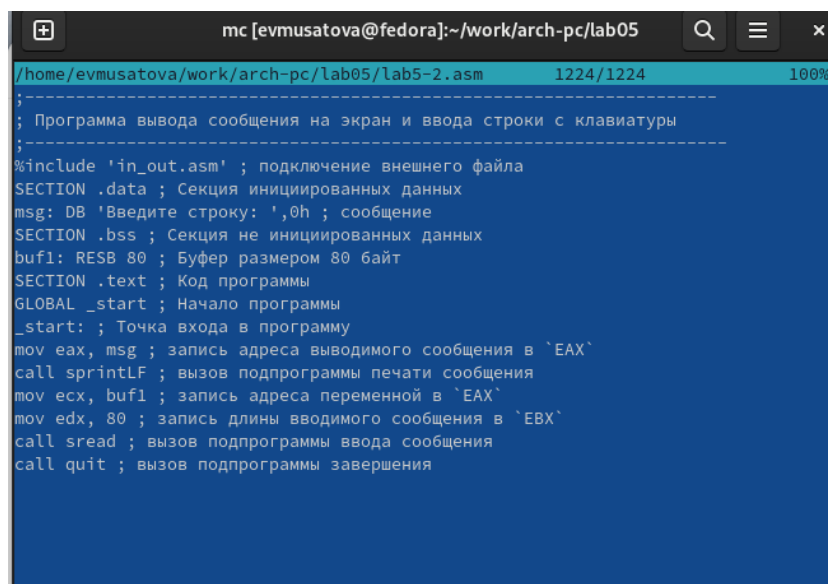


Рис. 2.9: Создание файла lab5-2.asm

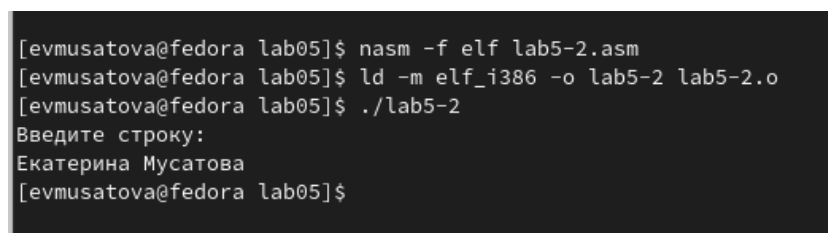
Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2 (рис. 2.10).



```
mc [evmusatova@fedora]:~/work/arch-pc/lab05 1224/1224 100%
/home/evmusatova/work/arch-pc/lab05/lab5-2.asm
;-----
; 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 sprintf ; 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. 2.10: Исправление текста программы

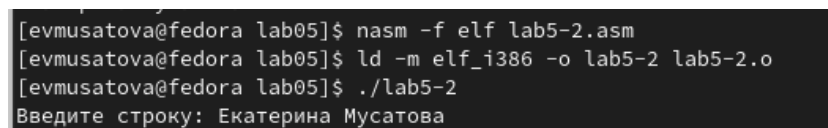
Создаю исполняемый файл и проверяю работу программы (рис. 2.11).



```
[evmusatova@fedora lab05]$ nasm -f elf lab5-2.asm
[evmusatova@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[evmusatova@fedora lab05]$ ./lab5-2
Введите строку:
Екатерина Мусатова
[evmusatova@fedora lab05]$
```

Рис. 2.11: Проверка программы

В файле lab5-2.asm заменяю подпрограмму sprintf на printf. Создаю исполняемый файл и проверяю его работу (рис. 2.12). Разница заключается в том, что теперь ввод производится на той же строке, что и вывод, убран символ перевода строки после вывода.



```
[evmusatova@fedora lab05]$ nasm -f elf lab5-2.asm
[evmusatova@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[evmusatova@fedora lab05]$ ./lab5-2
Введите строку: Екатерина Мусатова
```

Рис. 2.12: Проверка программы после замены

3 Выполнение заданий для самостоятельной работы

1. Копирую файл lab5-1.asm с именем lab5-3.asm (рис. [3.1]).



Рис. 3.1: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. [3.2]).

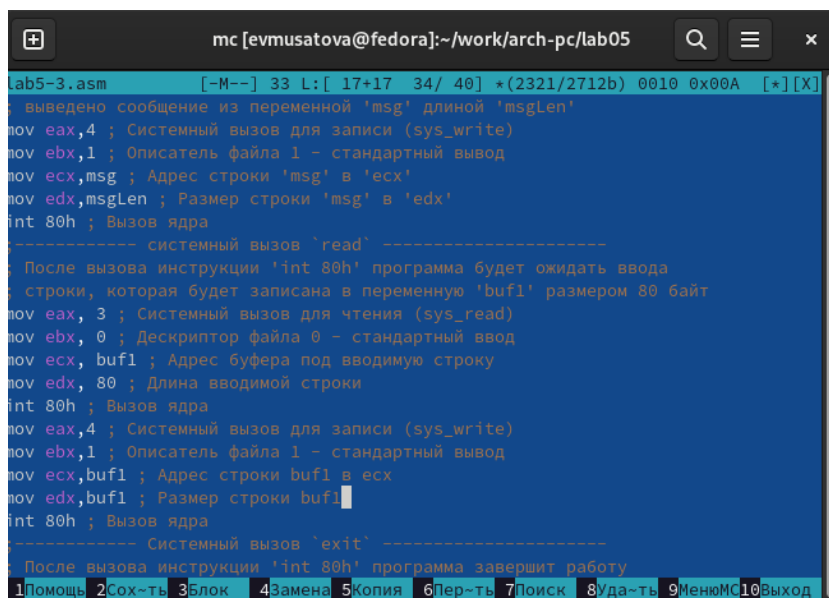


Рис. 3.2: Изменение программы

Создаю объектный файл lab5-3.o и компоную его в исполняемый файл (рис. [3.3]).

```
[evmusatova@fedora lab05]$ nasm -f elf lab5-3.asm
[evmusatova@fedora lab05]$ ld -m elf_i386 -o lab5-3 lab5-3.o
[evmusatova@fedora lab05]$ ./lab5-3
```

Рис. 3.3: Создание исполняемого файла

Запускаю программу и ввожу свою фамилию для проверки (рис. [3.4]).

```
[evmusatova@fedora lab05]$ ./lab5-3
Введите строку:
Мусатова
Мусатова
[evmusatova@fedora lab05]$
```

Рис. 3.4: Проверка

Программа из пункта 1:

;----- ; 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 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, buf1 ; Размер строки buf1 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

2. Копирую файл lab5-2.asm с именем lab5-4.asm (рис. [3.5]).

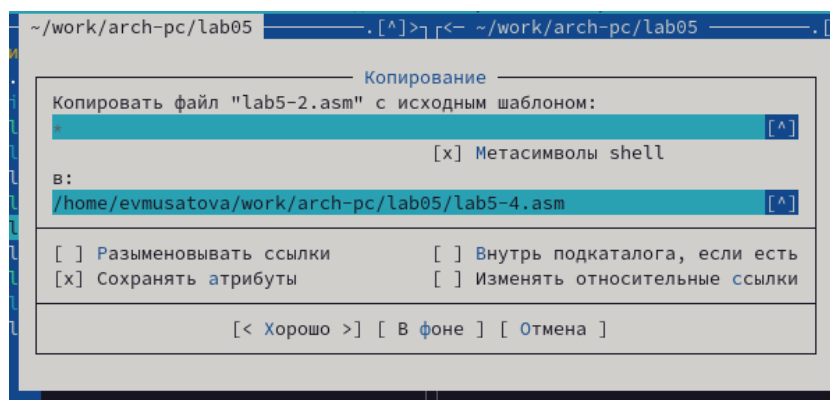
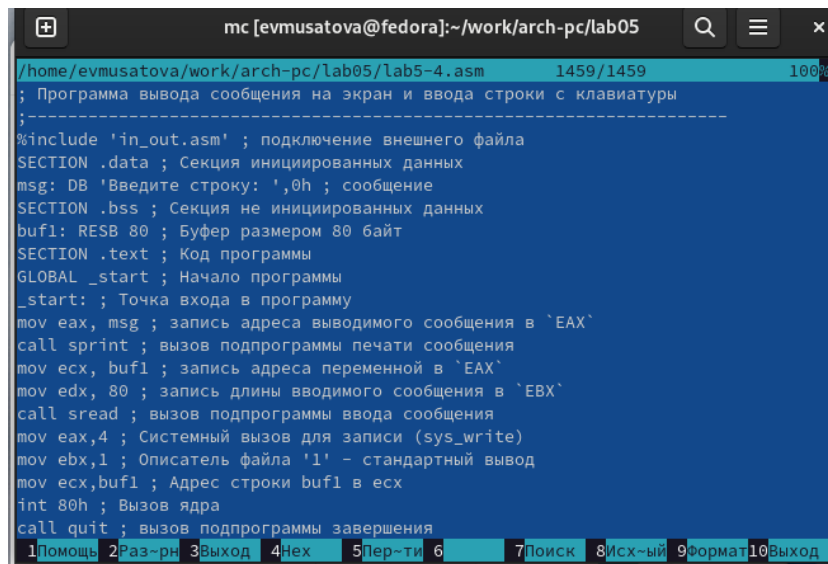


Рис. 3.5: Копирование файла

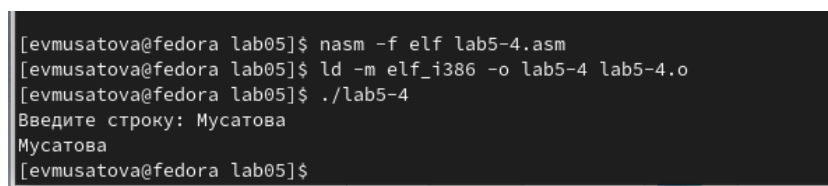
Изменяю код программы, добавляя вывод введенной строки (рис. [3.6]).



```
mc [evmusatova@fedora]: ~/work/arch-pc/lab05
/home/evmusatova/work/arch-pc/lab05/lab5-4.asm 1459/1459 100%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных 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
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
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```

Рис. 3.6: Изменение файла

Создаю исполняемый файл и запускаю его (рис. [3.7]).



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

Рис. 3.7: Запуск программы

Программа из пункта 2:

;----- ; Программа вывода сообщения на
экран и ввода строки с клавиатуры ;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция
инициированных данных msg: DB 'Введите строку:',0h ; сообщение
SECTION .bss
; Секция не инициированных 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 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 int 80h ; Вызов ядра call quit ; вызов подпрограммы завершения

4 Выводы

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.