

Отчёт по лабораторной работе №5

Архитектура вычислительных систем

Зеленко Ирина Юрьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	создание lab05	8
4.2	создание lab5-1.asm	8
4.3	Открытие lab5-1.asm	9
4.4	редактирование	10
4.5	Ввод текста	11
4.6	оттранслирование текста	12
4.7	Скачивание	12
4.8	Подключение	13
4.9	Создание копии	14
4.10	Исправления текста программы	15
4.11	Изменение в файле	16
4.12	Запуск файла lab04.asm	16
4.13	Проверка	17
4.14	Создание копии	18
4.15	РЕЗУЛЬТАТ	18

Список таблиц

1 Цель работы

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

2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.
4. Создайте исполняемый файл и проверьте его работу.

3 Теоретическое введение

4 Выполнение лабораторной работы

1. Я открыл mc, в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No4, создал lab05

Левая панель				Правая панель			
~/work/arch-pc				~			
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	ноя 22 16:06	/..		-ВВЕРХ-	ноя 22 16:06
/lab04		2048	ноя 22 17:09	/.cache		2048	ноя 22 17:09
/lab05		2048	дек 8 16:59	/.config		2048	ноя 22 17:09
				/grub		2048	ноя 22 17:09

Рис. 4.1: создание lab05

2. Пользуясь строкой ввода и командой touch создал файл lab5-1.asm

/lab05				-ВВЕРХ-			
2048G / 2048G (100%)				2048G / 2048G (100%)			
Совет: Переходите к часто используемым каталогам из справочника, набрав C-\.							
iyzelenko@dk1n22 ~/work/arch-pc \$ touch lab5-1.asm							
1Помощь	2Меню	3Просмотр	4Правка	5Копия	6Перенос	7НаКтлог	8Удалить

Рис. 4.2: создание lab5-1.asm

3. Я открыл файл lab5-1.asm для редактирования во встроенном редакторе и проверил его


```

SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
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
;----- 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. 4.3: Открытие lab5-1.asm

```

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
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: редактирование

4. Ввёл текст программы из листинга 6.1, сохранил изменения и закрыл файл.

```

/afs/.dk.sci.pfu.edu.ru/home/i/y/iyenko/work/arch-pc/lab05/lab5-1.asm 202
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
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
;----- 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. 4.5: Ввод текста

5. Я Оттранслировал текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустил получившийся исполняемый файл

```

iyzelenko@dk1n22 ~ $ cd work
iyzelenko@dk1n22 ~/work $ cd arch-pc
iyzelenko@dk1n22 ~/work/arch-pc $ cd lab05
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Зеленко Ирина Юрьевна

```

Рис. 4.6: оттранслирование текста

6. Скачал и подключил in_out.asm

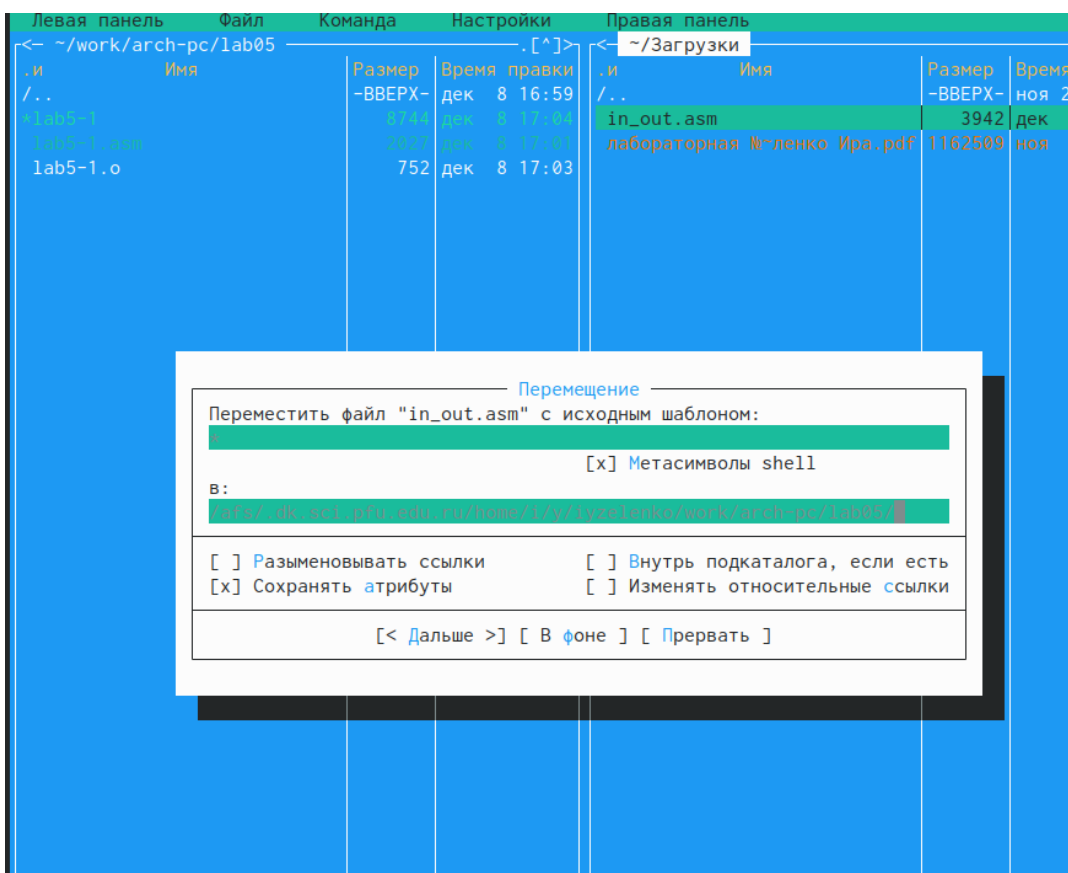


Рис. 4.7: Скачивание

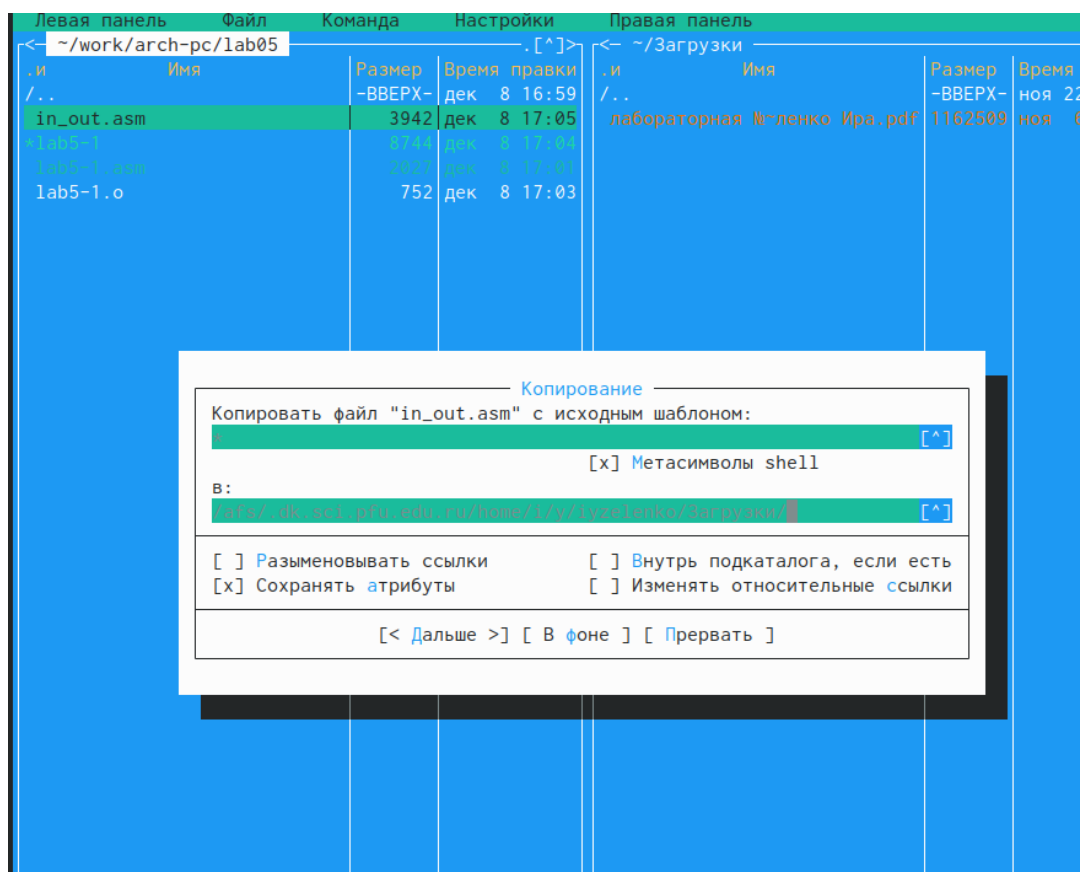


Рис. 4.8: Подключение

7. Создал копию файла `lab6-1.asm` с именем `lab6-2.asm`

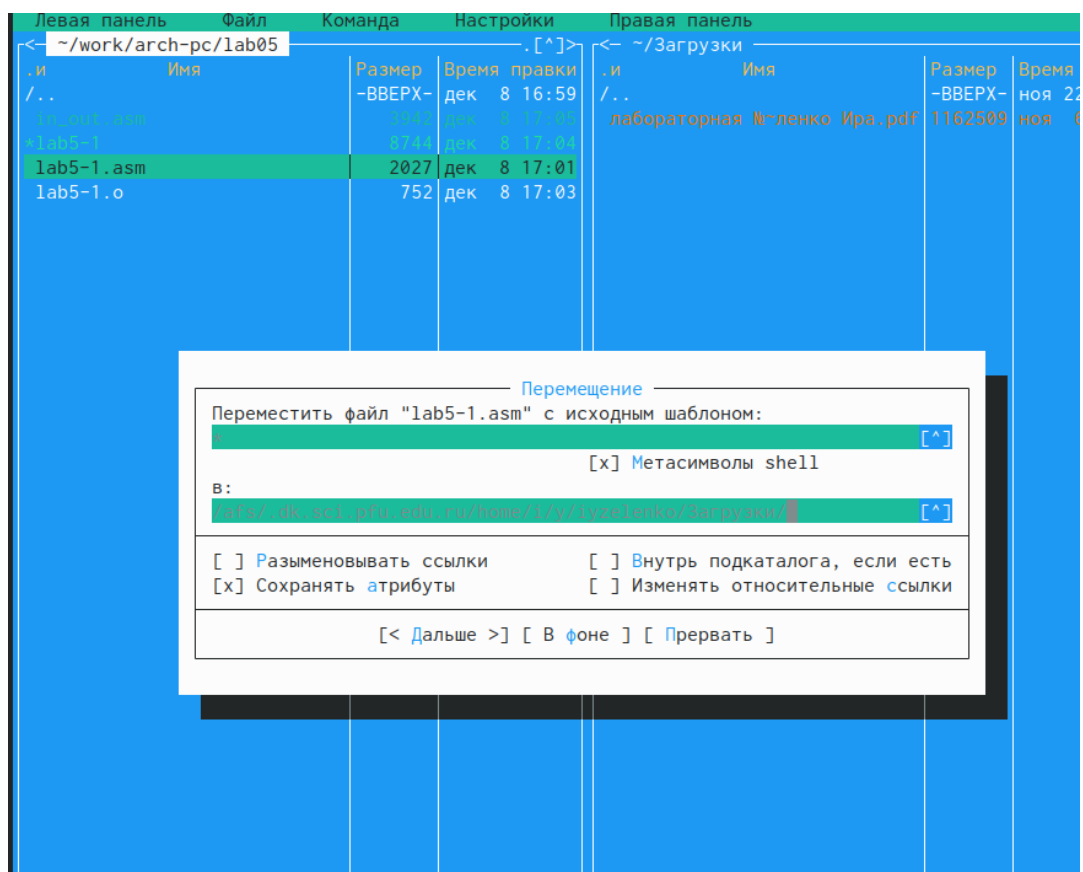


Рис. 4.9: Создание копии

- Исправил текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm

```
GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/i/y/iyzelenko/work/arch-pc/lab05/lab5-2.asm
#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 printf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 4.10: Исправления текста программы

9. В lab6-2.asm заменил подпрограмму printf на sprintf, создал исполняемый файл и выявил разницу - она заключается в выводе текста (отступ для ввода либо пробелом, либо новой строкой)

< ~/work/arch-pc/lab05 .[^]> <-				
.и	Имя	Размер	Время	правки
/..		-ВВЕРХ-	дек	8 16:59
	in_out.asm	3942	дек	8 17:05
	*lab5-1	8744	дек	8 17:04
	lab5-1.asm	2027	дек	8 17:01
	lab5-1.o	752	дек	8 17:03
	lab5-2.asm	964	дек	8 17:19

Рис. 4.11: Изменение в файле

10. Я создал копию файла lab6-1.asm и внёс изменения в программу

```

iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Зеленко Ирина Юрьевна

```

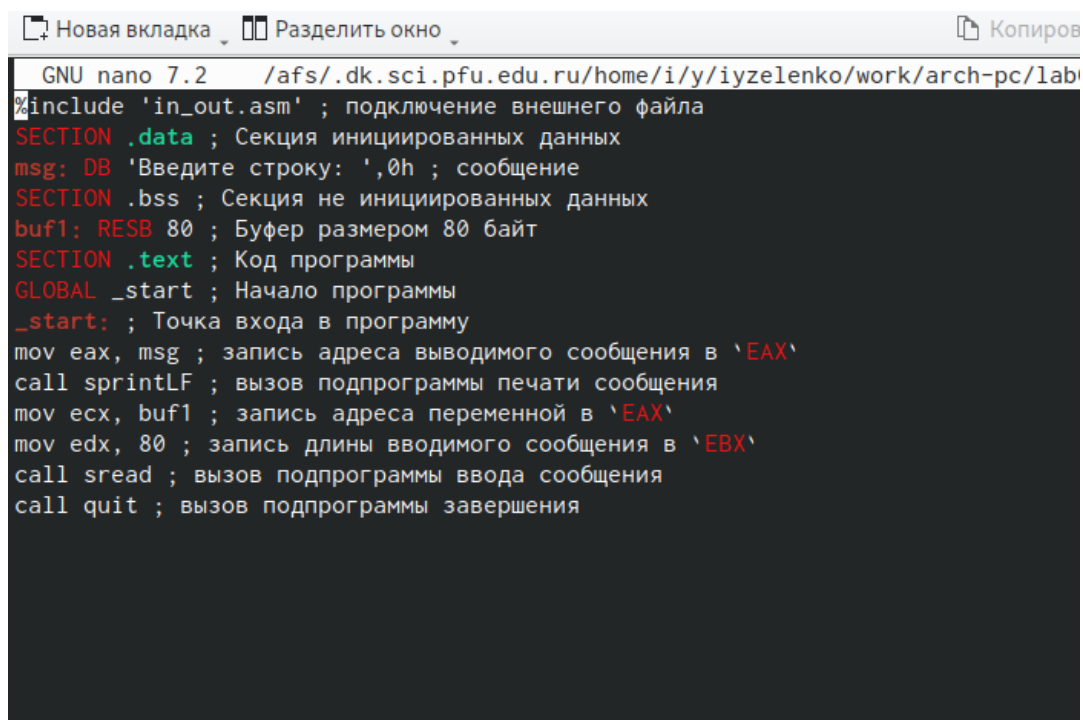
Рис. 4.12: Запуск файла lab04.asm

11. Я получил исполняемый файл и проверил его работу

Левая панель	Файл	Команда	Настройки
< ~/work/arch-pc/lab05			. [^] >
.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	дек 8 16:59
in_out.asm		3942	дек 8 17:05
*lab5-1		8744	дек 8 17:04
lab5-1.asm		2027	дек 8 17:01
lab5-1.o		752	дек 8 17:03
*lab5-2		9092	дек 8 17:21
lab5-2.asm		964	дек 8 17:19
lab5-2.o		1312	дек 8 17:20

Рис. 4.13: Проверка

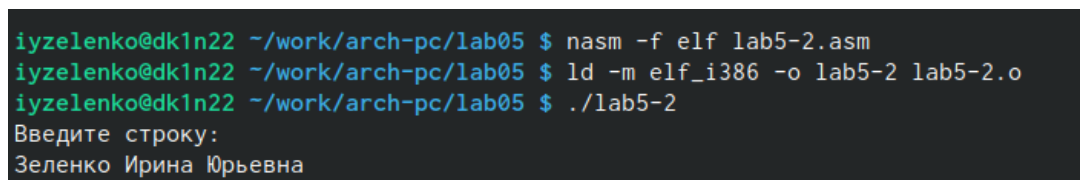
12. Создал копию файла lab6-2.asm. Исправил текст программы с использованием подпрограмм из внешнего файла in_out.asm



```
GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/i/y/iyzelenko/work/arch-pc/lab5-2.asm
%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 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. 4.14: Создание копии

12. Создал исполняемый файл и проверил его работу



```
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
iyzelenko@dk1n22 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Зеленко Ирина Юрьевна
```

Рис. 4.15: РЕЗУЛЬТАТ

5 Выводы

Мы приобрели практические навыки работы в Midnight Commander. И освоили инструкции языка ассемблера `mov` и `int`.

Список литературы