

Отчёт по лабораторной работе №5

Архитектура вычислительных систем

Федоров Андрей Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	16
	Список литературы	17

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	11
4.7	Скачивание	12
4.8	Подключение	12
4.9	Создание копии	13
4.10	Исправления текста программы	13
4.11	Запуск файла lab04.asm	14
4.12	Проверка	14
4.13	Создание копии	14
4.14	Исправление	15
4.15	РЕЗУЛЬТАТ	15

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

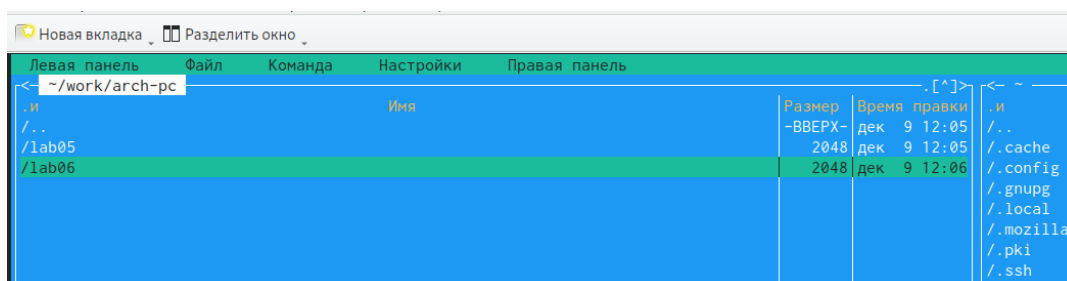


Рис. 4.1: создание lab05

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

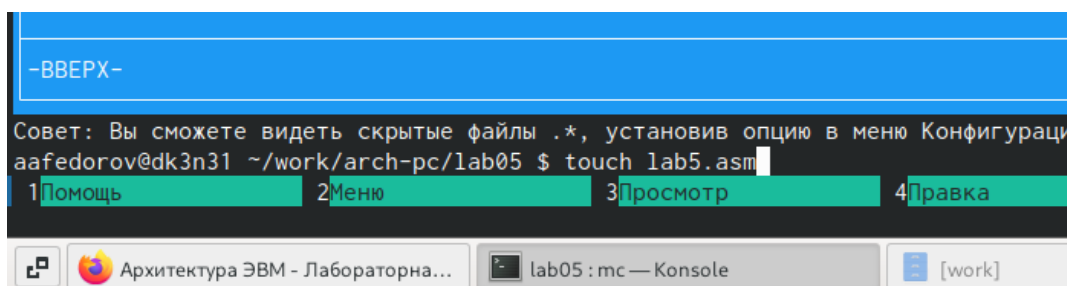
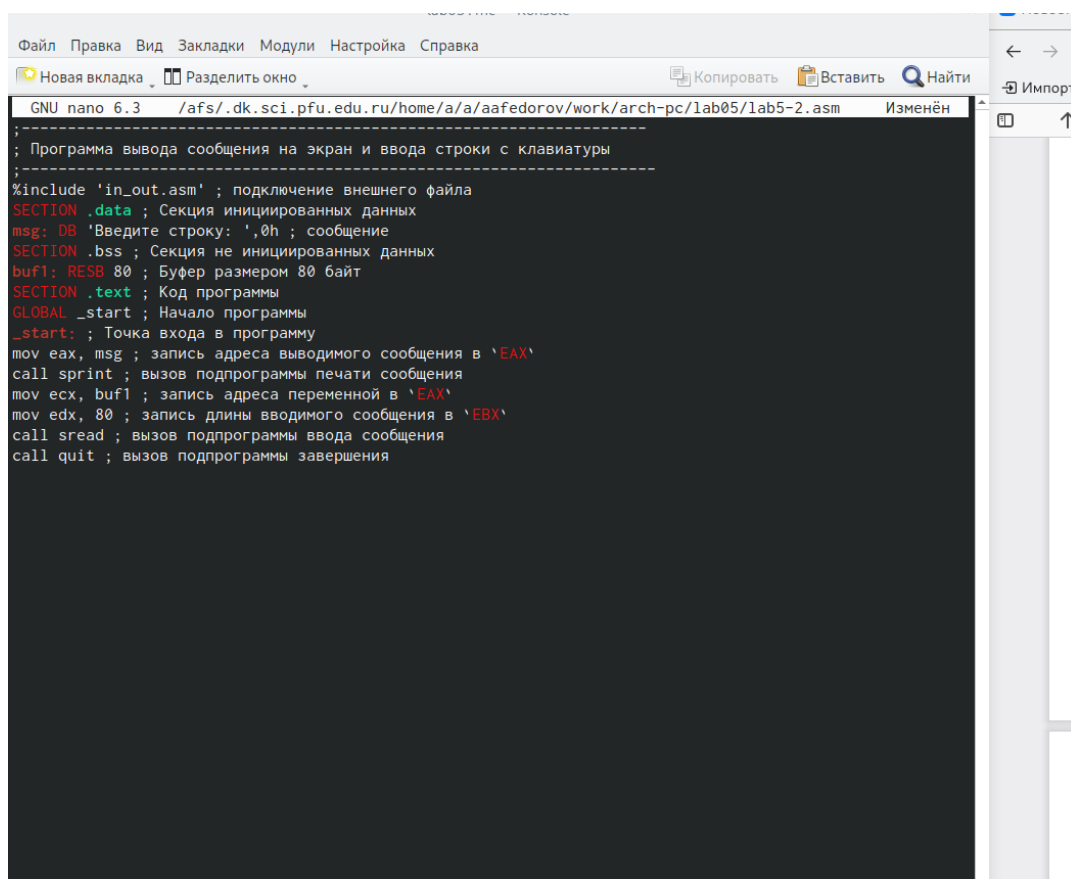


Рис. 4.2: создание lab5-1.asm

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


```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/a/aafedorov/wo
; 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
```

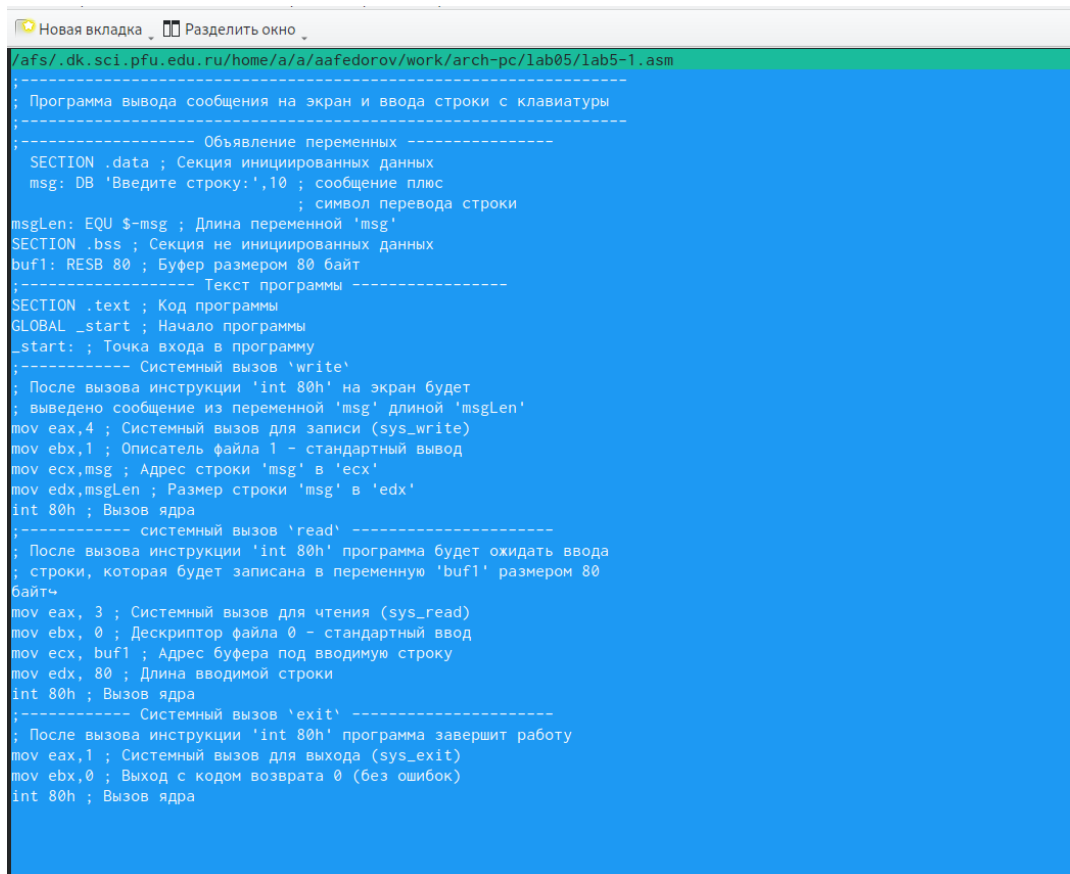
Рис. 4.3: Открытие lab5-1.asm



```
-----  
; 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 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  
call quit ; вызов подпрограммы завершения
```

Рис. 4.4: редактирование

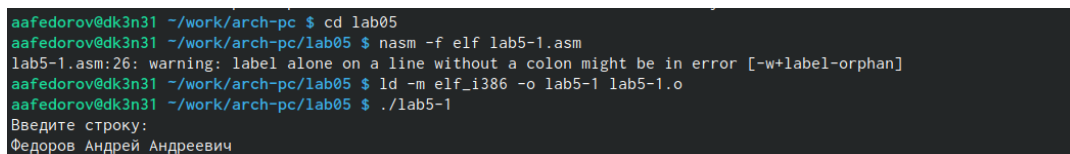
4. Ввёл текст программы из листинга 6.1, сохранил изменения и закрыл файл.



```
-----  
; 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
```

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



```
aafedorov@dk3n31 ~/work/arch-pc $ cd lab05  
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm  
lab5-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]  
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o  
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ./lab5-1  
Введите строку:  
Федоров Андрей Андреевич
```

Рис. 4.6: оттранслирование текста

6. Скачал и подключил in_out.asm

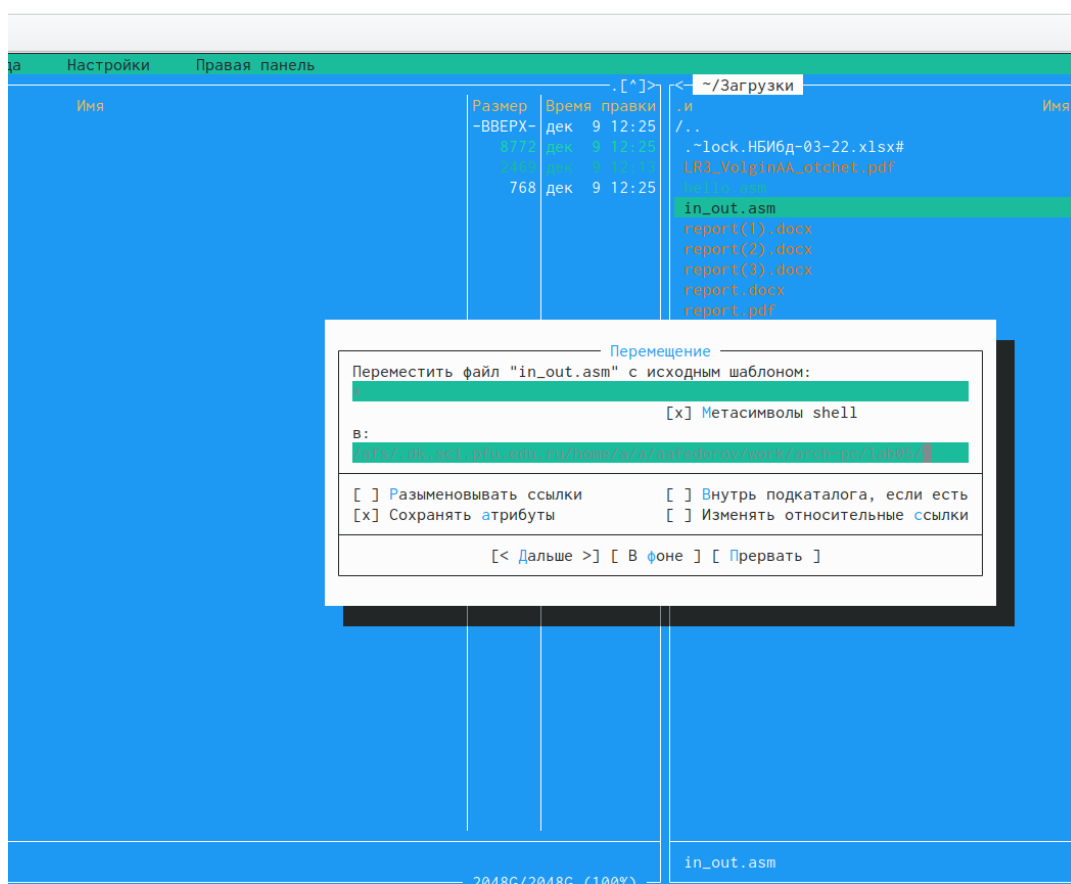


Рис. 4.7: Скачивание

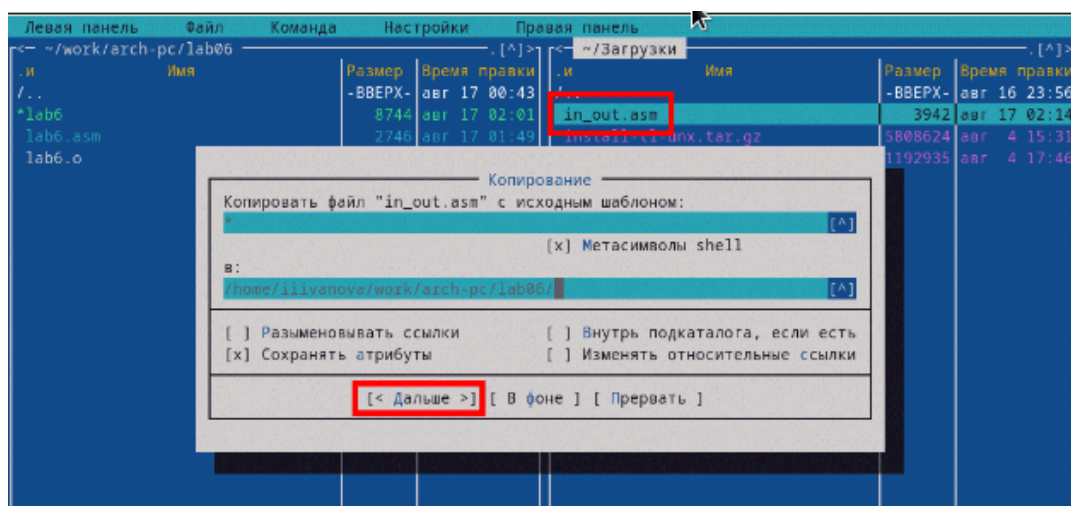


Рис. 4.8: Подключение

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

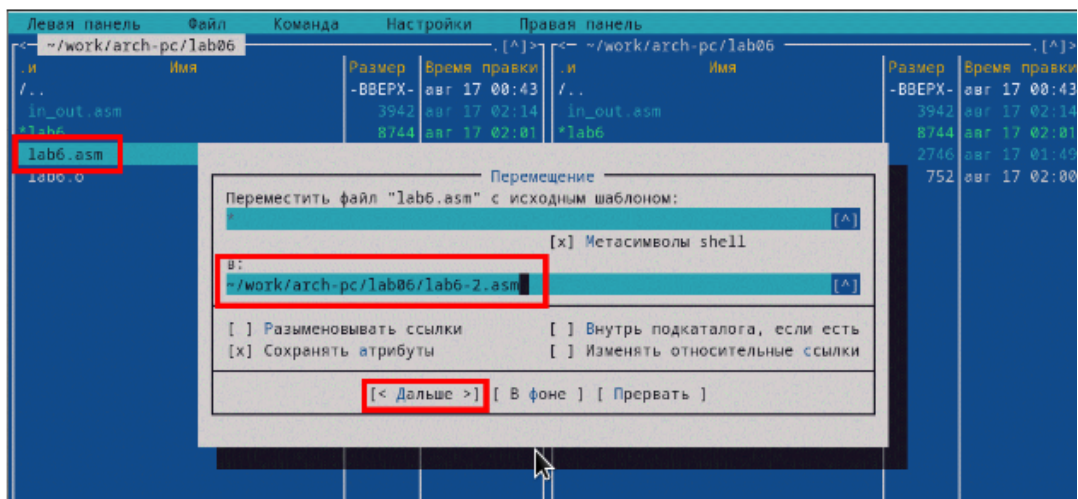


Рис. 4.9: Создание копии

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

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/a/a/aafedorov/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 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
quit ; вызов подпрограммы завершения
```

Рис. 4.10: Исправления текста программы

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

Изменение в файле

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

...	ВВЕРХ	дек	9 12:25	...
in_out.asm	194	дек	9 12:25	in_out.asm
lab5-1	877	дек	9 12:25	lab5-1
*lab5-1s.asm	2435	дек	9 12:54	*lab5-1s.o
lab5-2	877	дек	9 12:25	lab5-2
lab5-2.asm	1223	дек	9 12:38	lab5-2.asm
lab5-2.o	1312	дек	9 12:39	lab5-2.o

Рис. 4.11: Запуск файла lab04.asm

11. Я получил исполняемый файл и проверил его работу

```

aafedorov@dk3n31 ~/work/arch-pc/lab05 $ lm -m elf_i386 -o lab5-1s lab5-1s.o
bash: lm: команда не найдена
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1s lab5-1s.o
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ./lab5-1s
Введите строку:
Федоров
aafedorov@dk3n31 ~/work/arch-pc/lab05 $

```

Рис. 4.12: Проверка

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

и	Имя	Размер	Время	правки	и
...	-ВВЕРХ-	дек	9 12:25
in_out.asm	194	дек	9 12:25	in_out.asm	in_out.asm
lab5-1	877	дек	9 12:25	lab5-1	lab5-1
lab5-1s	877	дек	9 12:25	lab5-1s	lab5-1s
lab5-1s.asm	2435	дек	9 12:54	lab5-1s.asm	lab5-1s.asm
lab5-1s.o	768	дек	9 12:55	lab5-1s.o	lab5-1s.o
lab5-2.asm	1223	дек	9 12:38	lab5-2.asm	lab5-2.asm
lab5-2.o	1312	дек	9 12:39	lab5-2.o	lab5-2.o
lab5-2s.asm	1223	дек	9 12:38	lab5-2s.asm	lab5-2s.asm

Рис. 4.13: Создание копии

```

;-----
; 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. 4.14: Исправление

12. Создал исполняемый файл и проверил его работу

```

aafedorov@dk3n31 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2s.asm
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2s lab5-2s.o
aafedorov@dk3n31 ~/work/arch-pc/lab05 $ ./lab5-2s
Введите строку:
Федоров
aafedorov@dk3n31 ~/work/arch-pc/lab05 $

```

Рис. 4.15: РЕЗУЛЬТАТ

5 Выводы

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

Список литературы