

# **Отчёт по лабораторной работе 5**

Лобанова Екатерина Евгеньевна

# Содержание

<b>1 Выводы</b>	<b>12</b>
<b>Список литературы</b>	<b>13</b>

# Список иллюстраций

1	Рис 1 . . . . .	4
2	Рис 2 . . . . .	4
3	Рис 3 . . . . .	5
4	Рис 4 . . . . .	5
5	Рис 5 . . . . .	5
6	Рис 6 . . . . .	6
7	Рис 7 . . . . .	6
8	Рис 8 . . . . .	7
9	Рис 9 . . . . .	7
10	Рис 10 . . . . .	8
11	Рис 11 . . . . .	8
12	Рис 12 . . . . .	9
13	Рис 13 . . . . .	9
14	Рис 14 . . . . .	10
15	Рис 15 . . . . .	10
16	Рис 16 . . . . .	11
17	Рис 17 . . . . .	11

# Список таблиц

## #1. Цель работы

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

## #2. Порядок выполнения лабораторной работы

Открываем Midnight Commander (рис. 1).

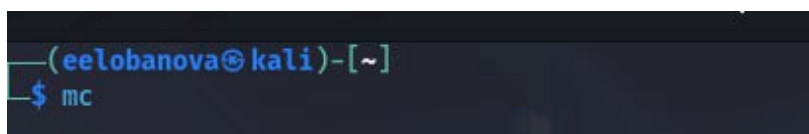


Рисунок 1: Рис 1

Переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной №4 с помощью клавиш (рис. 2).

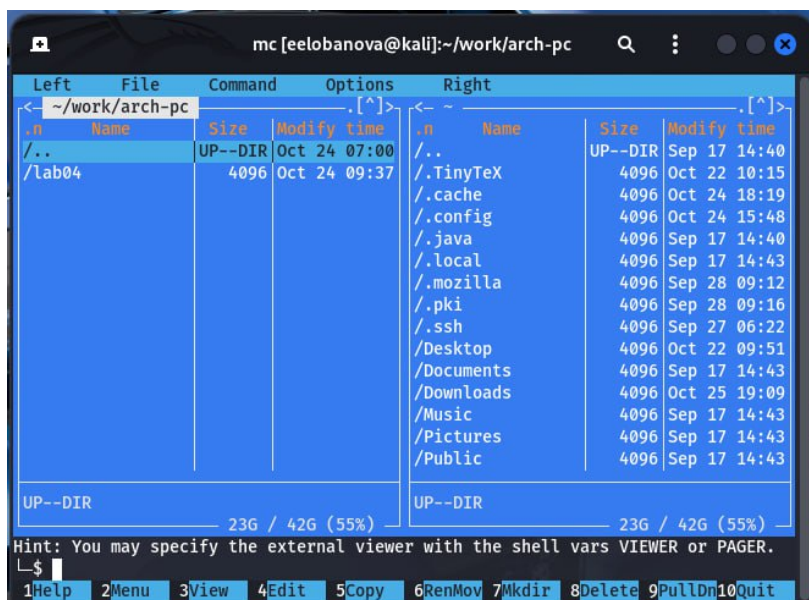


Рисунок 2: Рис 2

Создаем папку lab05 и переходим в нее. (рис. 3).

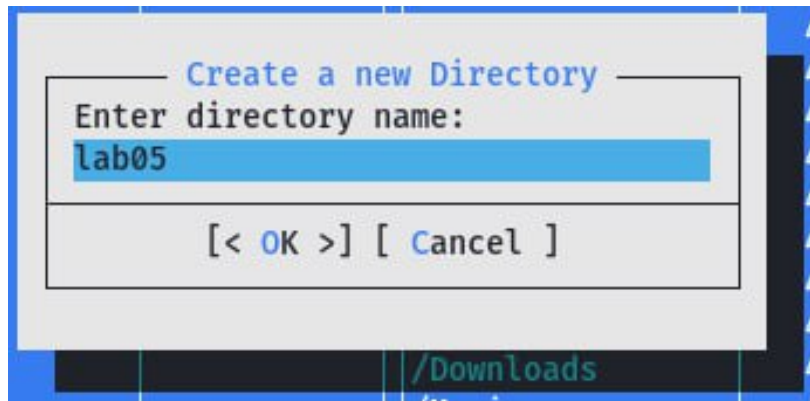


Рисунок 3: Рис 3

Создаем файл lab05-1.asm используя команду touch (рис. 4).

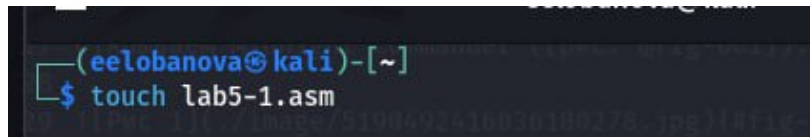


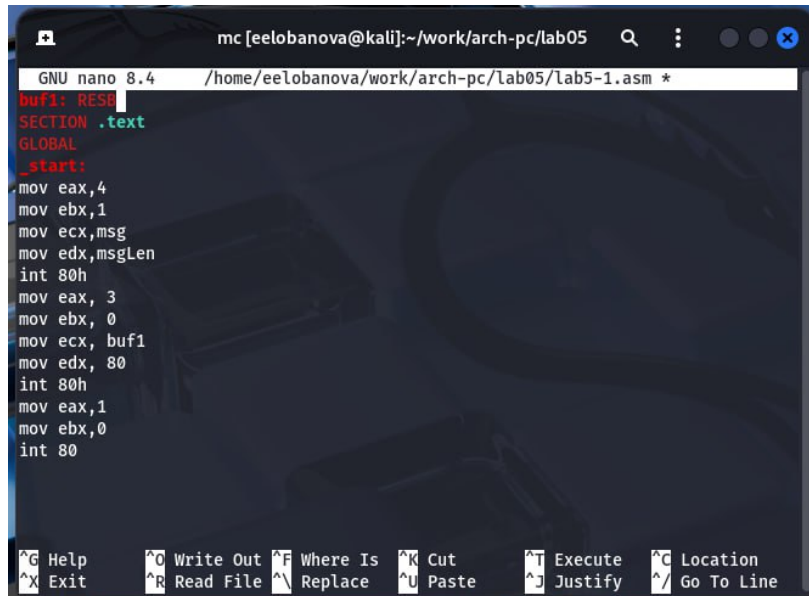
Рисунок 4: Рис 4

Открываем созданный файл и с помощью команды nano вводим текст из листинга 5.1 (рис. 5).



Рисунок 5: Рис 5

Открываем файл и убеждаемся в правильности введенного текста(рис. 6).

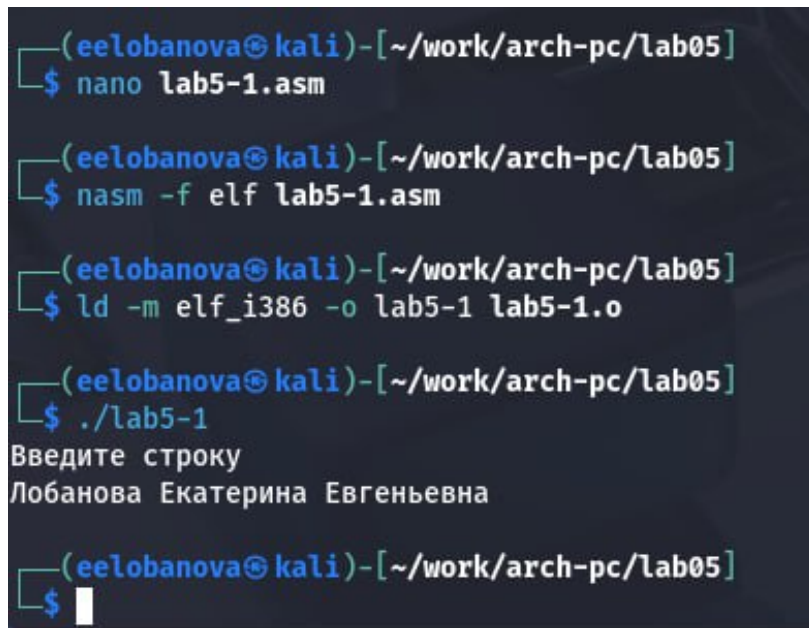
A screenshot of a terminal window with a dark background. The title bar shows 'mc [eelobanova@kali]:~/work/arch-pc/lab05'. The terminal content shows the GNU nano 8.4 editor editing a file at '/home/eelobanova/work/arch-pc/lab05/lab5-1.asm'. The code is as follows:

```
buf1: RESB
SECTION .text
GLOBAL
_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,1
mov ebx,0
int 80
```

At the bottom, there is a menu bar with various shortcuts: ^G Help, ^O Write Out, ^F Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\_ Replace, ^U Paste, ^J Justify, and ^\_ Go To Line.

Рисунок 6: Рис 6

Оттранслируем текст программы в объектный файл, выполняем компоновку файла и запускаем его(рис. 7).

A screenshot of a terminal window showing the compilation and execution of the assembly program. The terminal content is as follows:

```
(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nano lab5-1.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab5-1.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-1 lab5-1.o

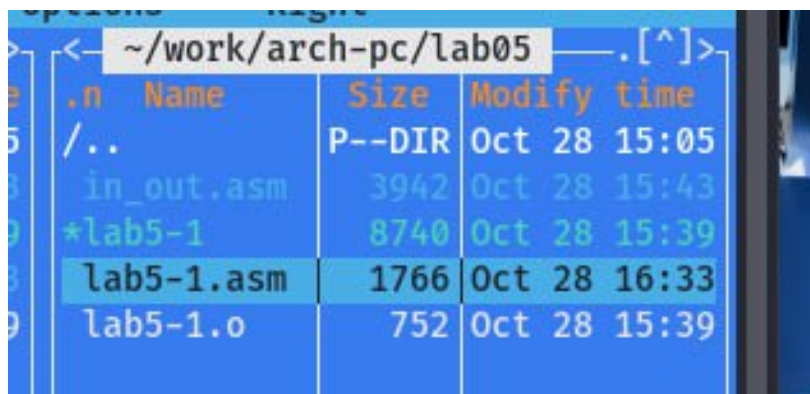
(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ./lab5-1
Введите строку
Лобанова Екатерина Евгеньевна

(eelobanova@kali)-[~/work/arch-pc/lab05]
$
```

Рисунок 7: Рис 7

## ##2.1. Подключение внешнего файла in\_out.asm

Скачиваем файл in\_out.asm со страницы в ТУИС и копируем его в каталог с файлом lab5-1.asm (рис. 8).



File	Name	Size	Modify time
../	P--DIR		Oct 28 15:05
in_out.asm		3942	Oct 28 15:43
*lab5-1		8740	Oct 28 15:39
lab5-1.asm		1766	Oct 28 16:33
lab5-1.o		752	Oct 28 15:39

Рисунок 8: Рис 8

Создаем копию файла lab5-1.asm и называем его lab5-2.asm (рис. 9).

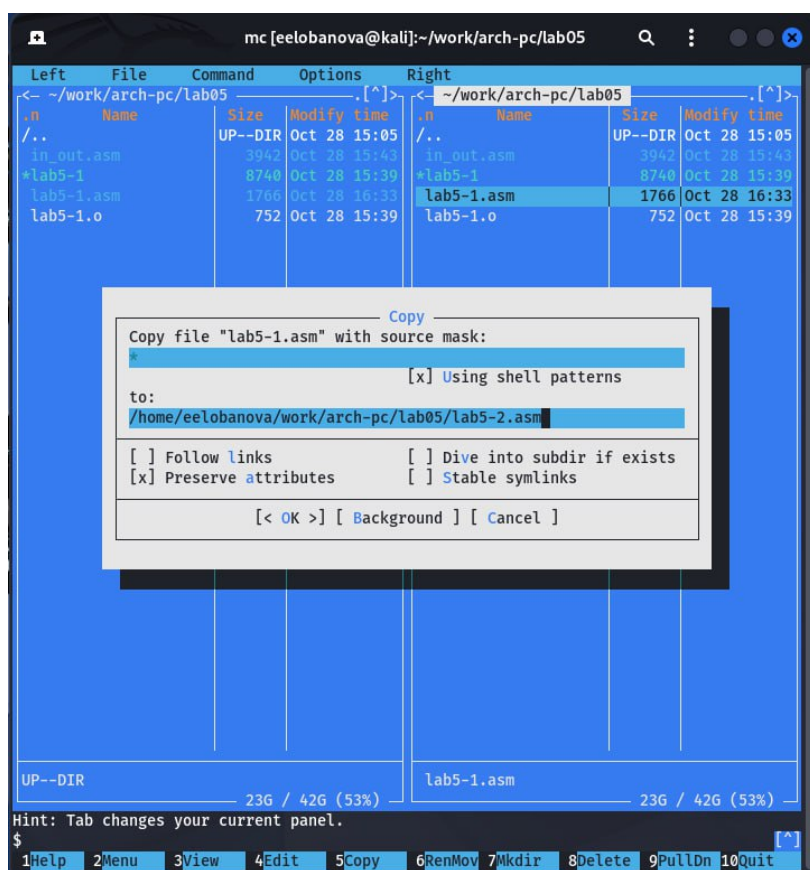


Рисунок 9: Рис 9

Исправляем текст программы lab5-2.asm используя программу in\_out и листинг 5.2 (рис. 10).

```
GNU nano 8.4 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
call quit ; вызов подпрограммы завершения
```

Рисунок 10: Рис 10

В файле lab5-2.asm заменяем подпрограмму sprintLF на sprint. Создаем исполняемый файл и проверяем его работу.(рис. 11).

```
(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab5-2.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-2 lab5-2.o

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ./lab5-2
Введите строку:
Лобанова Екатерина Евгеньевна

(eelobanova@kali)-[~/work/arch-pc/lab05]
$
```

Рисунок 11: Рис 11

При использовании sprintLF выводимые строчки видны с новой сторки, а с sprint на той же самой.

#3. Задание для самостоятельной работы

Копируем файл lab5-1.asm, переименовываем его в lab5-3.asm (рис. 12)



Left	File	Command	Options	Right
~/work/arch-pc/lab05	in_out.asm	UP--DIR	Oct 28 15:05	~/work/arch-pc/lab05
lab5-1	lab5-1.asm			lab5-1
lab5-1.o	lab5-1.o			lab5-1.o
lab5-2	lab5-2.asm			lab5-2
lab5-2.o	lab5-2.o			lab5-2.o
lab5-3.asm	lab5-3.asm			lab5-3.asm

Рисунок 12: Рис 12

Вносим изменения в код. (рис. 13)

```

; --- 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 0x80 ; Вызов ядра

; --- Вывод введенной строки ---
mov eax, 4
mov ebx, 1
mov ecx, buf1
mov edx, 80
int 0x80

```

Рисунок 13: Рис 13

Оттранслируем полученный текст программы lab5-3.asm, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Проверяем корректность работы. (рис. 14).

```

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab5-3.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-2 lab5-2.o

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-3 lab5-3.o

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ./lab5-3
Введите строку:
Лобанова Екатерина Евгеньевна
Лобанова Екатерина Евгеньевна

```

Рисунок 14: Рис 14

Копируем файл lab5-2.asm, переименовываем его в lab5-4.asm (рис. 15)

Left	File	Command	Options	Right
<-	~/work/arch-pc/lab05			<- ~/work/arch-pc/lab05
.n	Name	Size	Modify time	.n Name Size Modify time
./..		UP--DIR	Oct 28 15:05	./.. UP--DIR Oct 28 15:05
in_out.asm		3942	Oct 28 15:43	in_out.asm 3942 Oct 28 15:43
*lab5-1		8740	Oct 28 15:39	*lab5-1 8740 Oct 28 15:39
lab5-1.asm		1766	Oct 28 16:33	lab5-1.asm 1766 Oct 28 16:33
lab5-1.o		752	Oct 28 15:39	lab5-1.o 752 Oct 28 15:39
*lab5-2		9092	Oct 28 17:20	*lab5-2 9092 Oct 28 17:20
lab5-2.asm		1223	Oct 28 16:49	lab5-2.asm 1223 Oct 28 16:49
lab5-2.o		1312	Oct 28 16:56	lab5-2.o 1312 Oct 28 16:56
*lab5-3		8744	Oct 28 17:21	*lab5-3 8744 Oct 28 17:21
lab5-3.asm		2480	Oct 28 17:20	lab5-3.asm 2480 Oct 28 17:20
lab5-3.o		784	Oct 28 17:20	lab5-3.o 784 Oct 28 17:20
lab5-4.asm		1223	Oct 28 16:49	lab5-4.asm 1223 Oct 28 16:49

Рисунок 15: Рис 15

Вносим изменения в код. (рис. 15)

```

SECTION .data      ; Секция инициализированных данных
msg: db 'Введите строку: ',0      ; 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 `ECX`
    mov edx, 80     ; запись длины вводимого сообщения в `EDX`
    call sread      ; вызов подпрограммы ввода сообщения
    mov eax, buf1   ; запись введенной строки в `EAX`
    call sprint     ; вызов подпрограммы печати строки
    call quit       ; вызов подпрограммы завершения

```

Рисунок 16: Рис 16

Оттранслируем полученный текст программы lab5-4.asm, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Проверяем корректность работы. (рис. 17)

```

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nano lab5-4.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab5-4.asm

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-4 lab5-4.o

(eelobanova@kali)-[~/work/arch-pc/lab05]
$ ./lab5-4
Введите строку:
Лобанова Екатерина Евгеньевна
Лобанова Екатерина Евгеньевна

```

Рисунок 17: Рис 17

# 1 Выводы

В ходе лабораторной работы были приобретены навыки работы в Midnight Commander в операционной системе Linux. Освоены инструкции языка ассемблера `mov` и `int`.

## **Список литературы**