

Лабораторная работа №6

Руслан Шухратович Исмаилов

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Задание для самостоятельной работы	8
5	Выводы	16

Список иллюстраций

4.1	интерфейс tc и файл созданный файл lab6-1	9
4.2	редактирование и сохранение файла	10
4.3	компилирование и работа исполняемого файла	11
4.4	перемещение файла	11
4.5	копирование файла с новым именем	12
4.6	код файла lab6-2	13
4.7	Разница sprint и sprintLF	13
4.8	код lab6-3	14
4.9	исполнение новой программы	15
4.10	код lab6-4	15

1 Цель работы

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

2 Задание

Используя Midnight Commander (MC) и языки ассемблера `mov` и `int` сделать базовые программы

3 Выполнение лабораторной работы

Шаг 1

Открываем midnight commander, и с помощью стрелок клавиатуры и 'enter' переходим в рабочий каталог и создаём каталог lab06, где мы будем проводить дальнейшую работу, также создаём (с помощью клавиши F4) файл lab6-1.asm, и редактируем его

(рис. 4.1)

Шаг 2

вводим в lab6-1.asm текст из *листинга 6.1*

(рис. 4.2)

сохраним изменения, и проверим наш файл, все с помощью клавиш F2, F3.

Шаг 3

Оттранслируем текст программы в объектный файл. Выполним компоновку объектного файла и запустим получившийся файл:

(рис. 4.3)

Шаг 4

Скопируем файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5

(рис. 4.4)

Скопируем файл lab6-1.asm с названием lab6-2.asm с помощью клавиши F6

(рис. 4.5)

Шаг 5

вводим в lab6-2.asm текст из *листинга 6.2*

(рис. 4.6)

Выполним компоновку объектного файла и запустим получившийся исполняемый файл (выше), а также заменим подпрограмму `sprintLF` на `sprint`, и снова проделаем прошлые шаги, и сравним разницу:

(рис. 4.7)

во втором файле отсутствует функция перевода строки после вывода сообщения на экран, поэтому введённое сообщение будет на одном уровне с “Введите строку:”

4 Задание для самостоятельной работы

Шаг 1

Копируем файл lab6-1.asm как lab6-3.asm и изменим его для того, чтобы на экран выводилась введённая строка:

(рис. 4.8)

Для этого в новом параграфе кода укажем:

- 1 Системный вызов для записи
- 2 Стандартный вывод
- 3 Адрес строки buf1 (ранее введённое сообщение)
- 4 Длину строки (80)

После вызова инструкции `int 80h` на экране будет сообщение buf1, что и является введённым текстом

(рис. 4.9)

Шаг 2

Прделаем схожую работу с копией lab6-2, lab6-4:

(рис. 4.10)

Здесь мы просто вписываем наше сообщение в регистр `eax` (необходим для работы `sprintf`) и используем подпрограмму `sprintf` для того чтобы вывести на экран введённое сообщение (buf1)

(рис. 4.9)

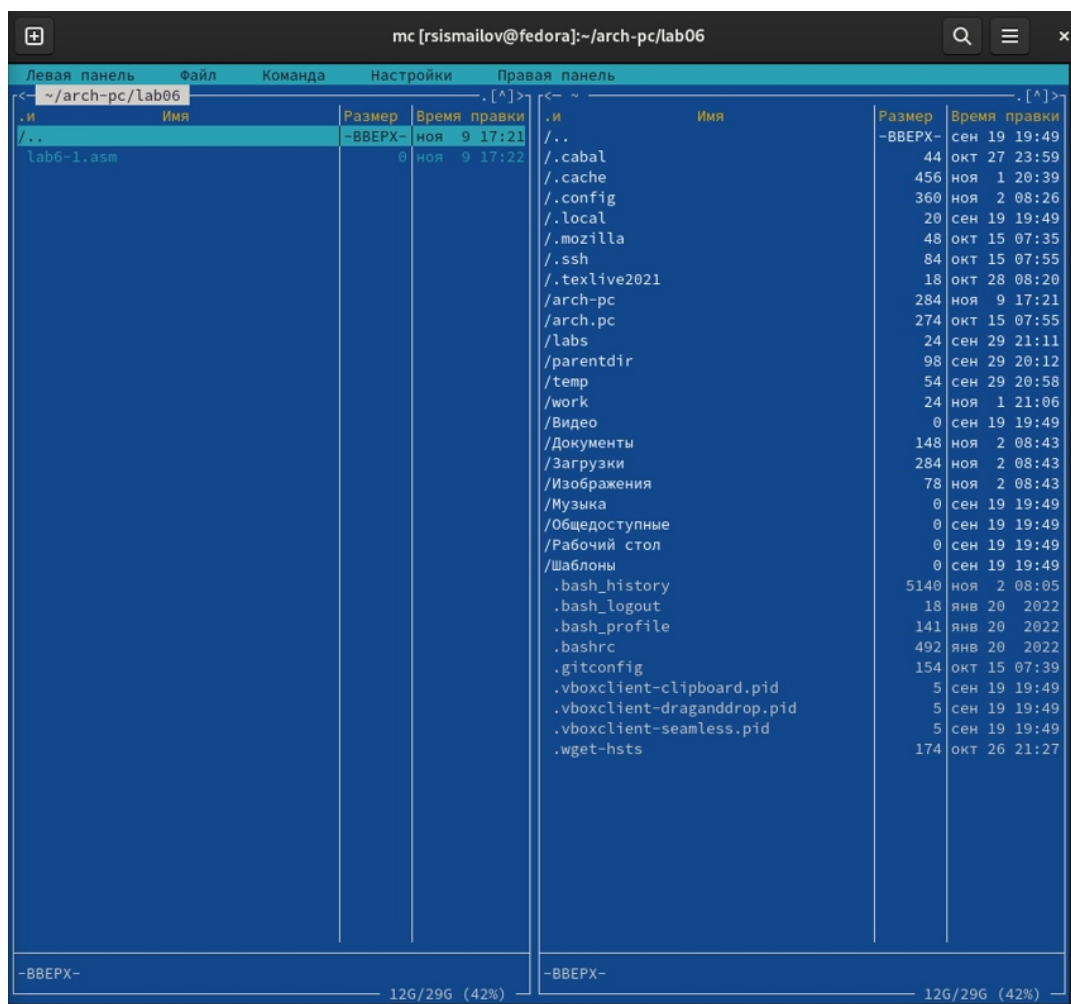


Рис. 4.1: интерфейс mc и файл созданный файл lab6-1

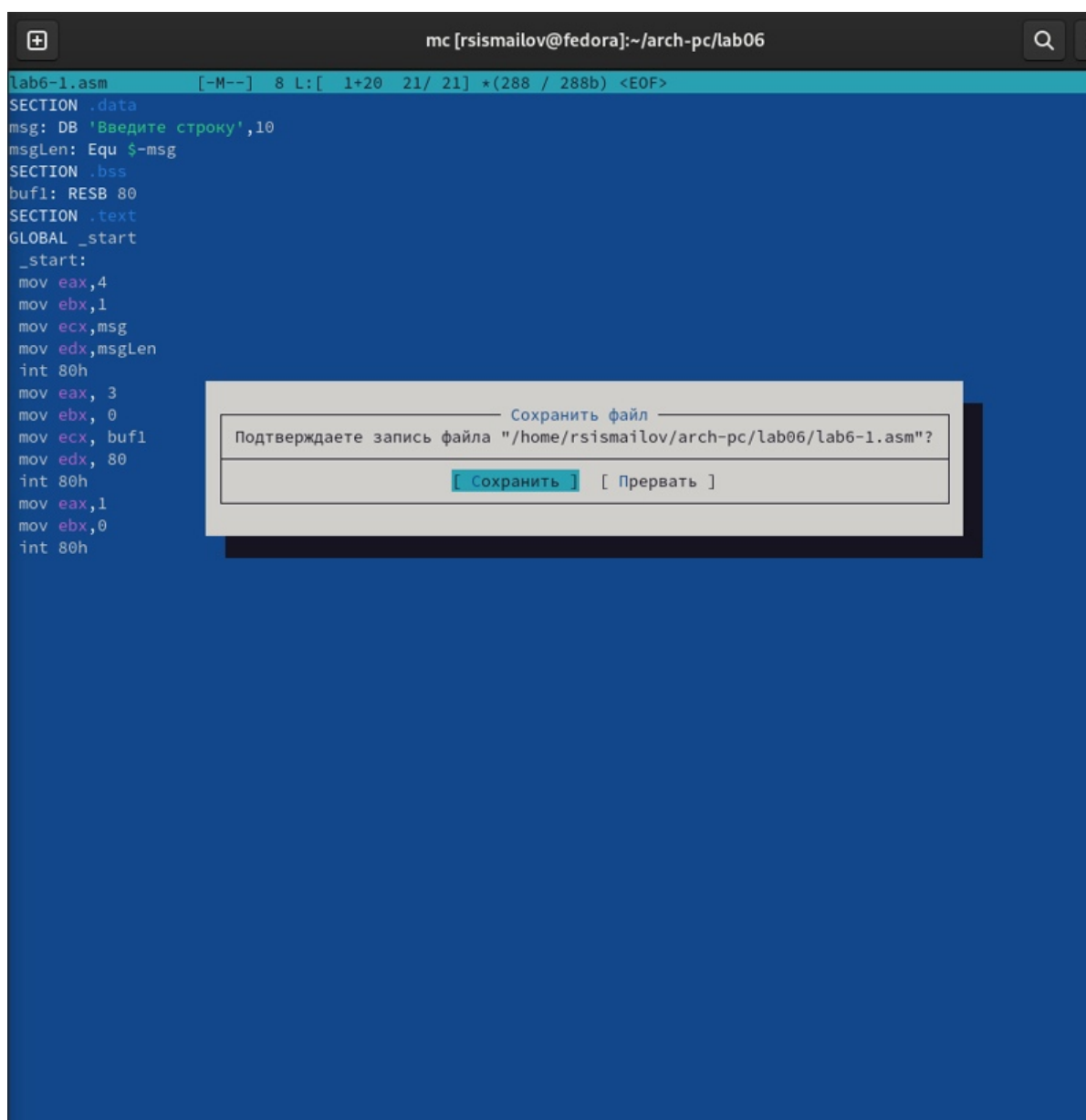


Рис. 4.2: редактирование и сохранение файла

```

[rsismailov@fedora ~]$ mc

[rsismailov@fedora lab06]$ touch lab6-1.asm

[rsismailov@fedora lab06]$ nasm -f elf lab6-1.asm

[rsismailov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o

[rsismailov@fedora lab06]$ ./lab6-1
Введите строку
Исмаилов Руслан Шухратович

```

Рис. 4.3: компилирование и работа исполняемого файла

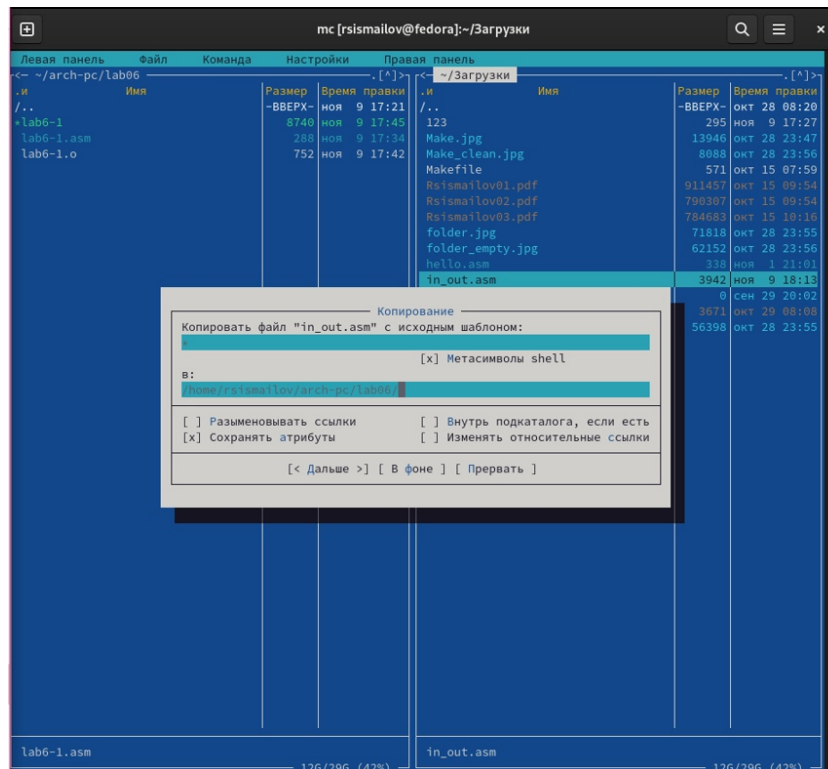


Рис. 4.4: перемещение файла

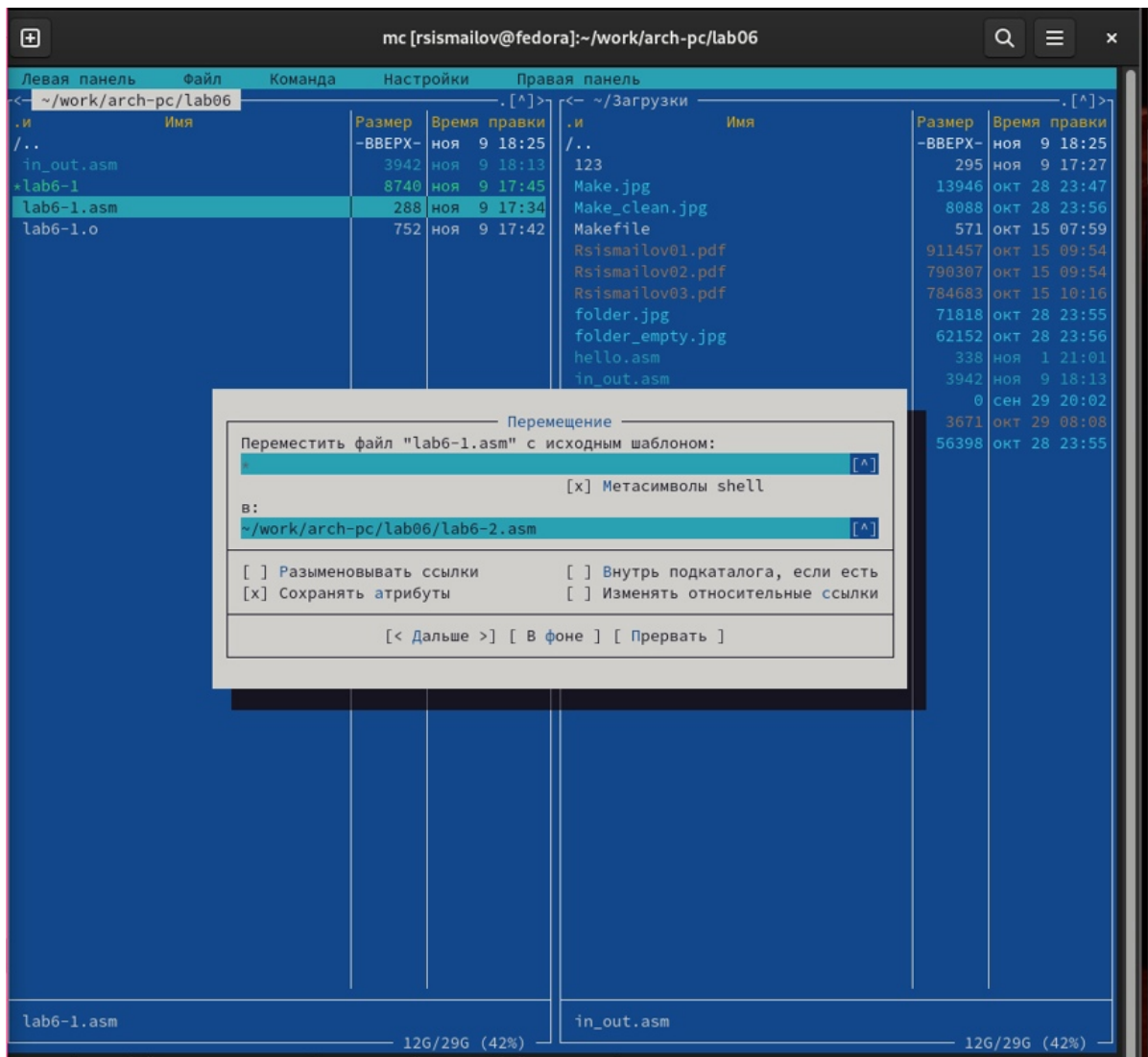


Рис. 4.5: копирование файла с новым именем



The screenshot shows a terminal window with the nano text editor open. The file being edited is `lab6-2.asm` located at `/home/rsismailov/work/arch-pc/lab06/`. The code in the editor is as follows:

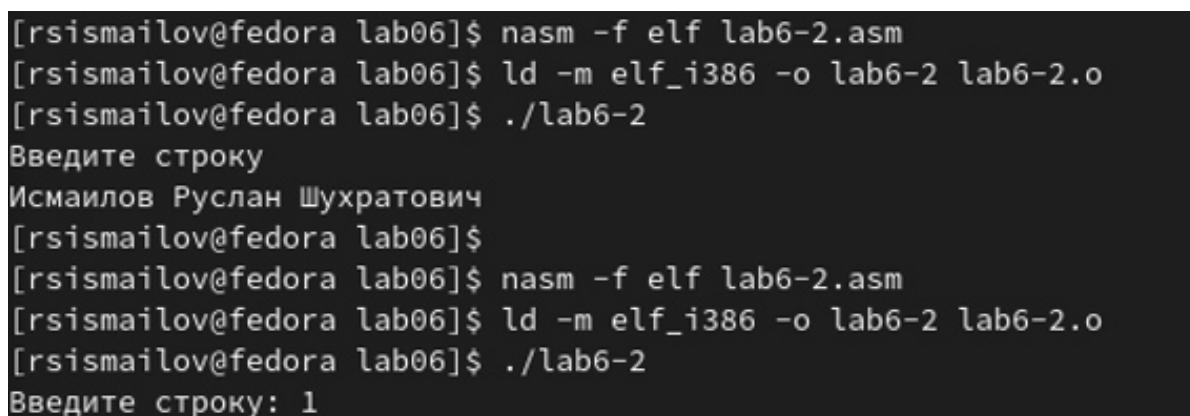
```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintf

mov ecx, buf1
mov edx, 80
call sread

call quit
```

Рис. 4.6: код файла lab6-2



The screenshot shows a terminal window with the following commands and output:

```
[rsismailov@fedora lab06]$ nasm -f elf lab6-2.asm
[rsismailov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[rsismailov@fedora lab06]$ ./lab6-2
Введите строку
Исмаилов Руслан Шухратович
[rsismailov@fedora lab06]$
[rsismailov@fedora lab06]$ nasm -f elf lab6-2.asm
[rsismailov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[rsismailov@fedora lab06]$ ./lab6-2
Введите строку: 1
```

Рис. 4.7: Разница sprint и sprintf

```
rsismaïlov@fedora:~/work/arch-pc/lab06
GNU nano 6.0 /home/rsismaïlov/work/arch-pc/lab06/lab6-3.asm
SECTION .data
msg: DB 'Введите строку:',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,80
int 80h

mov eax,1
mov ebx,0
int 80h
```


Рис. 4.8: код lab6-3

```

[rsismailov@fedora lab06]$ ./lab6-3
Введите строку:
Исмаилов Руслан Шухратович
Исмаилов Руслан Шухратович
[rsismailov@fedora lab06]$ ./lab6-4
Введите строку:
Исмаилов Руслан Шухратович
Исмаилов Руслан Шухратович

```

Рис. 4.9: исполнение новой программы



The screenshot shows a terminal window with the title bar "mc [rsismailov@fedora]:~/work/arch-pc/lab06". The editor is GNU nano 6.0, editing the file /home/rsismailov/work/arch-pc/lab06/lab6-4.asm. The code is as follows:

```

#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 sprintLF

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprintLF

call quit

```

Рис. 4.10: код lab6-4

5 Выводы

Я освоил инструкции языка ассемблера `mov` и `int`, успешно использовал подпрограммы из внешнего файла и использовал `midnight commander` для навигации по каталогам и созданию и изменению файлов