

# **Отчёт по лабораторной работе №6**

**Дисциплина: Архитектура компьютера**

**Батова Ирина Сергеевна, НММбд-01-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>13</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

2.1	Команда для открытия Midnight Commander . . . . .	6
2.2	Переход в каталог arch-pc . . . . .	6
2.3	Создание каталога lab06 . . . . .	7
2.4	Создание файла lab6-1.asm . . . . .	7
2.5	Файл lab6-1.asm . . . . .	8
2.6	Ввод листинга программы . . . . .	8
2.7	Исполнение программы из файла 'lab6-1.asm' . . . . .	9
2.8	Скачивание файла 'in_out.asm' . . . . .	9
2.9	Копирование файла 'in_out.asm' в нужный каталог . . . . .	10
2.10	Копирование файла 'lab6-1.asm' с именем 'lab6-2.asm' . . . . .	10
2.11	Листинг программы с использованием внешнего файла . . . . .	11
2.12	Исполнение программы из файла 'lab6-2.asm' . . . . .	11
2.13	Изменение подпрограммы из файла 'lab6-2.asm' . . . . .	12
2.14	Исполнение новой программы из файла 'lab6-2.asm' . . . . .	12
3.1	Копирование файла 'lab6-1.asm' с именем 'lab6-3.asm' . . . . .	13
3.2	Изменение текста программы в 'lab6-3.asm' . . . . .	14
3.3	Исполнение программы из файла 'lab6-3.asm' . . . . .	14
3.4	Копирование файла 'lab6-2.asm' с именем 'lab6-4.asm' . . . . .	15
3.5	Изменение текста программы в 'lab6-4.asm' . . . . .	15
3.6	Исполнение программы из файла 'lab6-4.asm' . . . . .	16

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Вводим команду 'mc' в терминале и открываем Midnight Commander (рис. 2.1).

```
[isbatova@fedora ~]$ mc
```

Рис. 2.1: Команда для открытия Midnight Commander

2. Переходим в каталог arch-pc, который был создан при выполнении пятой лабораторной работы (рис. 2.2).

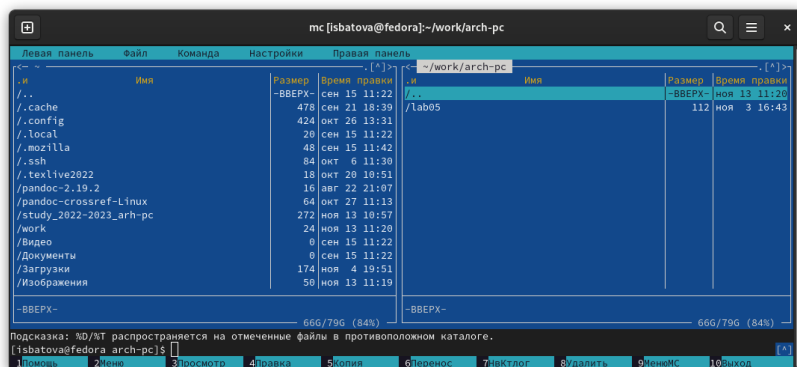


Рис. 2.2: Переход в каталог arch-pc

3. Нажимаем клавишу F7 и создаем каталог с именем lab06 (рис. 2.3).

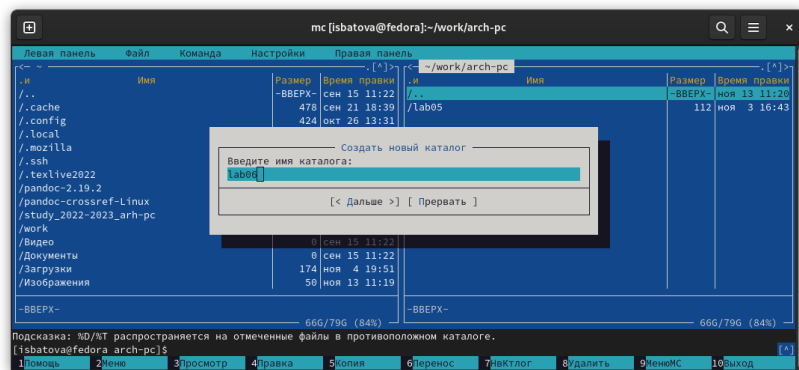


Рис. 2.3: Создание каталога lab06

4. С помощью команды 'touch' создаем в этом каталоге файл 'lab6-1.asm' (рис. 2.4).

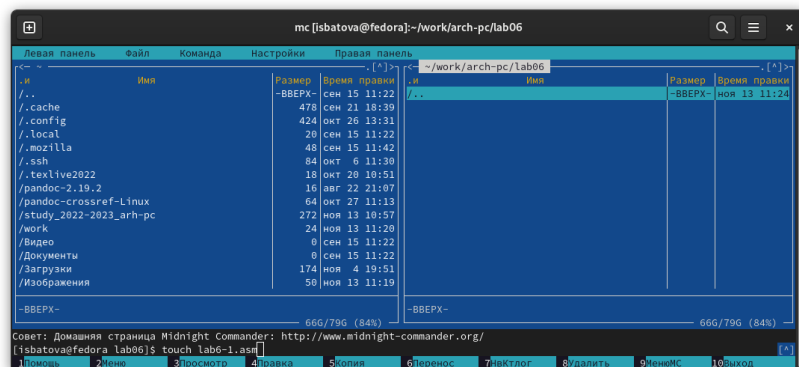


Рис. 2.4: Создание файла lab6-1.asm

5. Нажимаем команду F4 и открываем файл 'lab6-1.asm' (рис. 2.5).

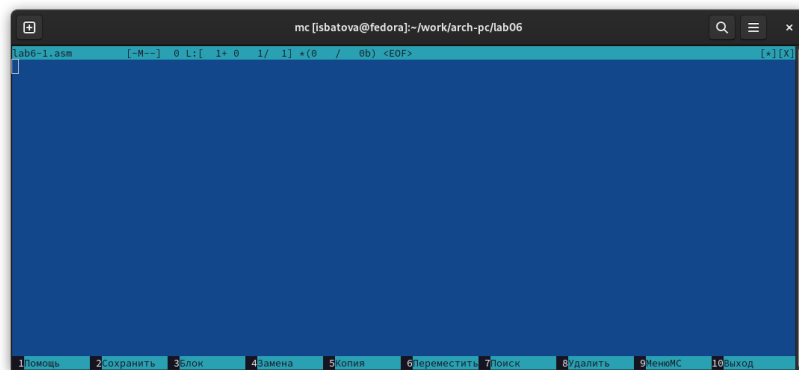


Рис. 2.5: Файл lab6-1.asm

6. Вводим в файл программу. Листинг программы берем из лабораторной работы (рис. 2.6).

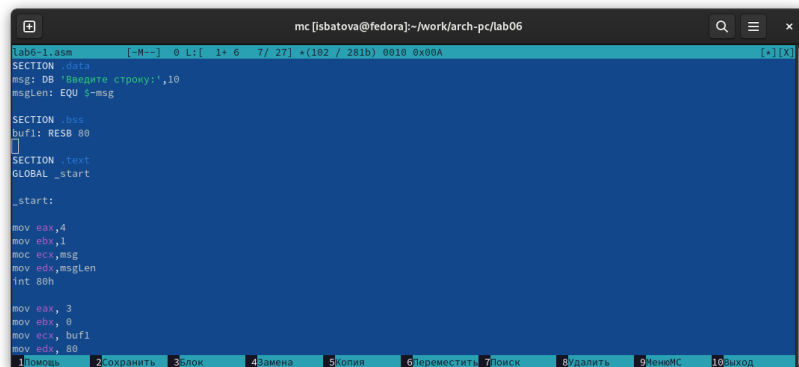


Рис. 2.6: Ввод листинга программы

Далее нажимаем F2 для сохранения файла и F10 для выхода из редактора.

7. Нажимаем клавишу F3 и проверяем, что в файле содержится текст программы.
8. Закрываем Midnight Commander. Далее нужно сделать из 'lab6-1.asm' объектный файл (команда 'nasm'), выполнить его компоновку (команда 'ld') и запустить получившийся исполняемый файл (команда './lab6-1'). Программа работает корректно: выводит строку "Введите строку", на запрос вводим имя и фамилию (рис. 2.7).



```

[isbatova@fedora lab06]$ nasm -f elf lab6-1.asm
[isbatova@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[isbatova@fedora lab06]$ ./lab6-1
Введите строку:
Ирина Батова
[isbatova@fedora lab06]$ 

```

Рис. 2.7: Исполнение программы из файла 'lab6-1.asm'

9. Заходим в ТУИС и скачиваем файл 'in\_out.asm', используемый для упрощения написания программ (рис. 2.8).

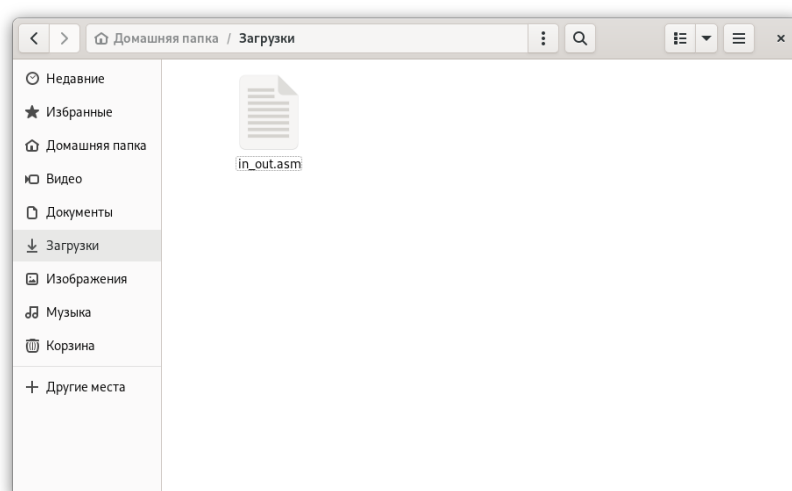


Рис. 2.8: Скачивание файла 'in\_out.asm'

10. Для корректной работы этот подключаемый файл должен лежать в одном каталоге с файлом с программы. Открываем тмс и на одной панели открываем каталог lab06 (где лежит файл 'lab6-1.asm'), а на другой панели открываем каталог загрузок (где лежит файл 'in\_out.asm'). Далее нажимаем клавишу F5 и копируем наш внешний файл в каталог 'lab06' (рис. 2.9).

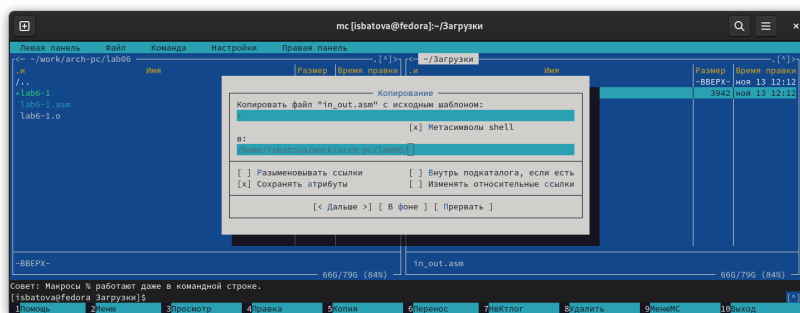


Рис. 2.9: Копирование файла 'in\_out.asm' в нужный каталог

11. Далее выделяем файл 'lab6-1.asm', нажимаем клавишу F6 и копируем этот файл с именем 'lab6-2.asm' (рис. 2.10).

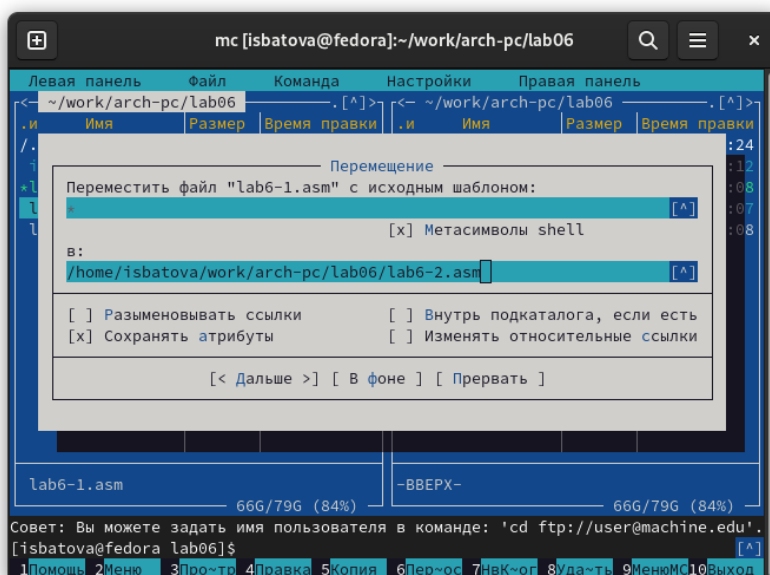


Рис. 2.10: Копирование файла 'lab6-1.asm' с именем 'lab6-2.asm'

12. Открываем файл 'lab6-2.asm'. В начале вводим '%include 'in\_out.asm'' (подключаем внешний файл). Далее также заменяем подпрограммами из файла печать сообщения, ввод сообщения и завершение (рис. 2.11).



```
GNU nano 6.0 /home/isbatova/work/arch-pc/lab06/lab6-2.asm
%include 'in_out.asm'

SECTION .data
    msg:    db "Введите строку:",0h
    msgLen: equ $ - msg

SECTION .bss
    buf1:   RESB 80

SECTION .text
    global _start

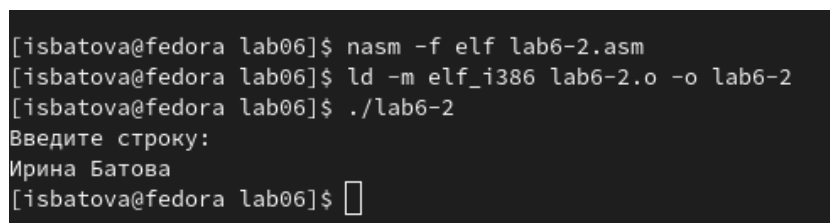
_start:
    mov eax, msg
    call sprintf

    mov ecx, buf1
    mov edx, 80
    call sread

    call quit
```

Рис. 2.11: Листинг программы с использованием внешнего файла

Далее сохраняем файл и выходим из Midnight Commander. Вводим поочередно команды 'nasm', 'ld' и './lab6-1' (оттранслировать программу в объектный файл, выполнить компоновку исполняемого файла и запустить его). Видим, что программа работает корректно (рис. 2.12).



```
[isbatova@fedora lab06]$ nasm -f elf lab6-2.asm
[isbatova@fedora lab06]$ ld -m elf_i386 lab6-2.o -o lab6-2
[isbatova@fedora lab06]$ ./lab6-2
Введите строку:
Ирина Батова
[isbatova@fedora lab06]$
```

Рис. 2.12: Исполнение программы из файла 'lab6-2.asm'

13. Открываем файл 'lab6-2.asm' и заменяем подпрограмму 'sprintf' на 'sprint' (рис. 2.13).



```
GNU nano 6.0 /home/isbatova/work/arch-pc/lab06/lab6-2.asm
#include 'in_out.asm'

SECTION .data
msg: db "Введите строку:",0h
msgLen: equ $ - msg

SECTION .bss
buf1: RESB 80

SECTION .text
global _start

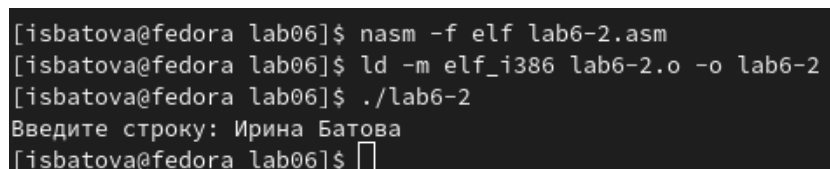
_start:
mov eax, msg
call sprint

mov ecx, buf1
mov edi, 80
call sread

call quit
```

Рис. 2.13: Изменение подпрограммы из файла 'lab6-2.asm'

Создаем исполняемый файл и запускаем его с помощью последовательности команд, описанных выше (рис. 2.14).



```
[isbatova@fedora lab06]$ nasm -f elf lab6-2.asm
[isbatova@fedora lab06]$ ld -m elf_i386 lab6-2.o -o lab6-2
[isbatova@fedora lab06]$ ./lab6-2
Введите строку: Ирина Батова
[isbatova@fedora lab06]$
```

Рис. 2.14: Исполнение новой программы из файла 'lab6-2.asm'

Разница между 'sprint' и 'sprintLF' в том, что первая просто выводит сообщение на экран, а вторая добавляет к выводу сообщения переход на новую строку.

### 3 Задание для самостоятельной работы

1. Открываем Midnight Commander, выделяем файл 'lab6-1.asm' и копируем его с именем 'lab6-3.asm' (рис. 3.1).

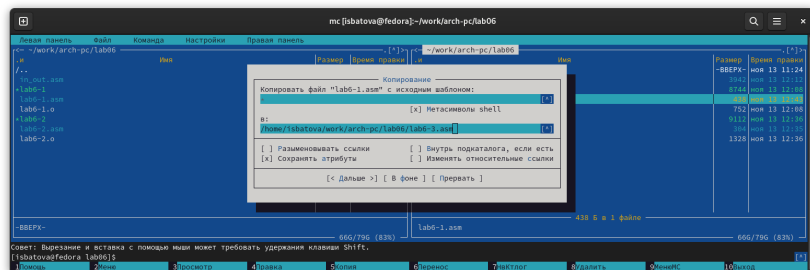


Рис. 3.1: Копирование файла 'lab6-1.asm' с именем 'lab6-3.asm'

Нам необходимо внести изменения в программу так, чтобы помимо вывода приглашения “Введите строку:” и возможности ввести строку с клавиатуры, она выводила введенную строку на экран. Для этого в тексте программы перед командой завершения вводим:

```
mov eax, 4 mov ebx, 1 mov ecx, buf1 int 80h
```

Далее сохраняем и закрываем файл (рис. 3.2).

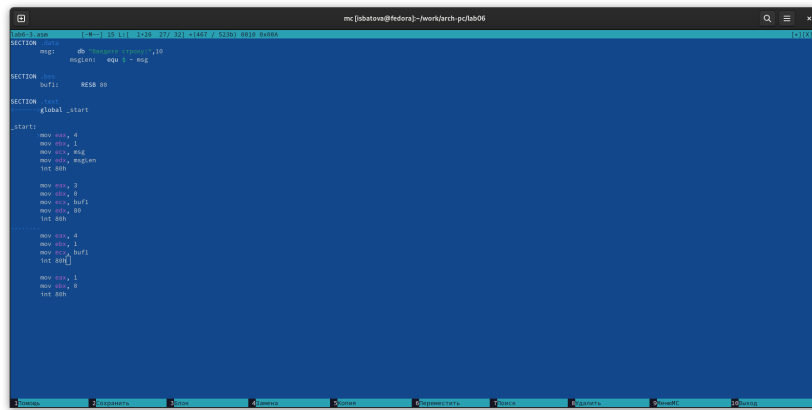


Рис. 3.2: Изменение текста программы в 'lab6-3.asm'

Выполняем создание исполняемого файла и запускаем его (рис. 3.3). На приглашение вводим фамилию - команда выводит нашу фамилию еще раз. Значит, программа написана корректно.

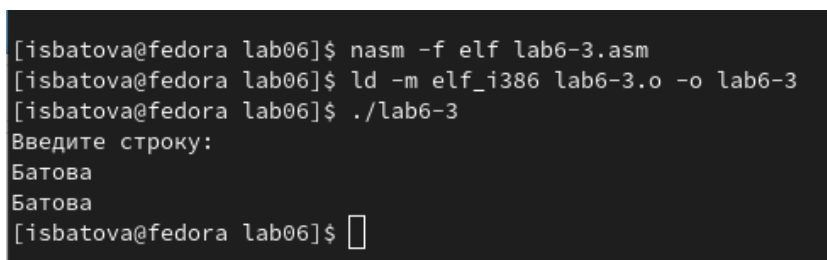


Рис. 3.3: Исполнение программы из файла 'lab6-3.asm'

2. Далее нам необходимо изменить программу из файла 'lab6-2.asm' так, чтобы она работала аналогично программы из первого пункта. Для начала открываем Midnight Commander и копируем файл 'lab6-2.asm' с именем 'lab6-4.asm' (рис. 3.4).

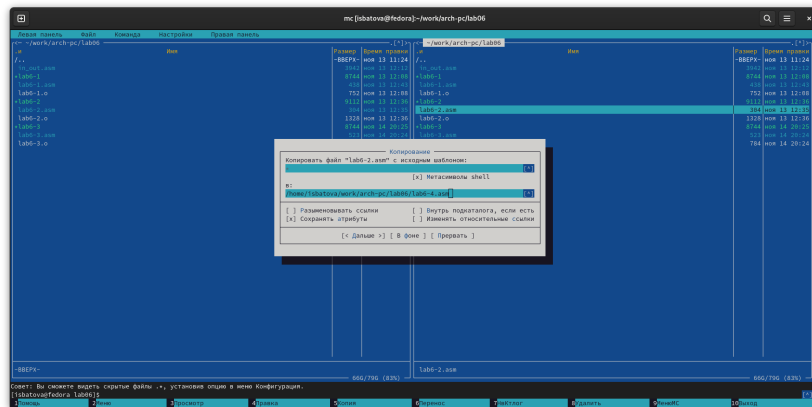


Рис. 3.4: Копирование файла 'lab6-2.asm' с именем 'lab6-4.asm'

Открываем файл. Изменяем его аналогично первому, но тут нам необходимо использовать внешний файл с подпрограммами. Поэтому в тексте программы перед командой завершения вводим:

```
mov eax, buf1 call sprint
```

Далее сохраняем и закрываем файл (рис. 3.5).

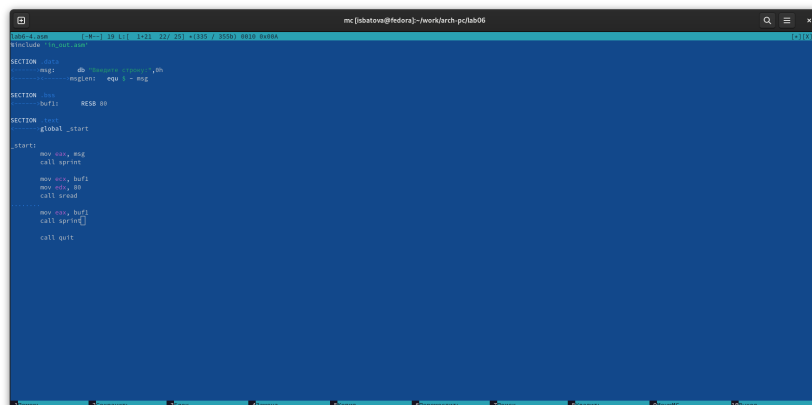


Рис. 3.5: Изменение текста программы в 'lab6-4.asm'

После этого создаем исполняемый файл и запускаем его (рис. 3.6). Видим, что программа работает корректно.

```
[isbatova@fedora lab06]$ nasm -f elf lab6-4.asm
[isbatova@fedora lab06]$ ld -m elf_i386 lab6-4.o -o lab6-4
[isbatova@fedora lab06]$ ./lab6-4
Введите строку: Батова
Батова
[isbatova@fedora lab06]$
```

Рис. 3.6: Исполнение программы из файла 'lab6-4.asm'



## 4 Выводы

В данной лабораторной работе мной были приобретены практические навыки работы в Midnight Commander, а также освоены инструкции языка ассемблера `mov` и `int`.