

Отчёт по лабораторной работе №6

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux.

Аскеров Александр Эдуардович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Подключение внешнего файла in_out.asm	10
2.2	Задания для самостоятельной работы	14
3	Выводы	18

Список иллюстраций

2.1	Открытый Midnight Commander	5
2.2	Переход в каталог ~/work/archpc	6
2.3	Создание каталога lab06	6
2.4	Создание файла lab6-1.asm	7
2.5	Открытие файла lab6-1.asm в mcedit	8
2.6	Текст программы	9
2.7	Проверка наличия текста программы	9
2.8	Создание объектного файла, его компоновка и ввод имени	10
2.9	Копирование файла in_out.asm в рабочий каталог	11
2.10	Создание копированием файла lab6-2.asm	12
2.11	Внесение изменений в файл lab6-2.asm	13
2.12	Убеждаемся в работоспособности программы	13
2.13	Работа изменённой программы	14
2.14	Внесённые в копию файла lab6-1.asm изменения	15
2.15	Результат работы программы lab6-3.asm	15
2.16	Внесённые в копию файла lab6-2.asm изменения	16
2.17	Результат работы программы lab6-4.asm	17

1 Цель работы

Приобрести практические навыки работы в Midnight Commander. Освоить инструкции языка ассемблера mov и int.

2 Выполнение лабораторной работы

1. Откроем Midnight Commander.

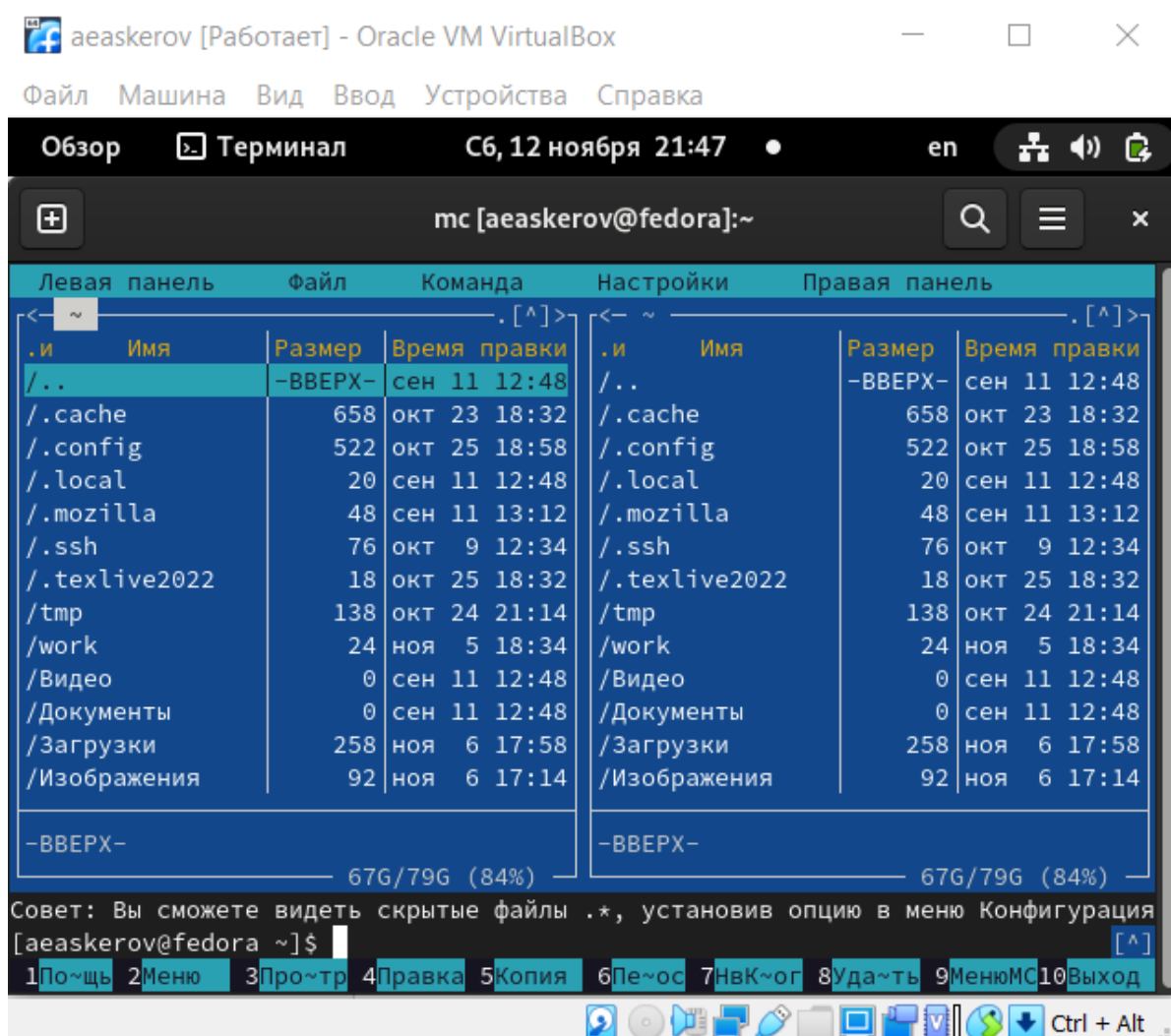


Рис. 2.1: Открытый Midnight Commander

2. Пользуясь клавишами Up, Down и Enter перейдём в каталог ~/work/archpc, созданный при выполнении лабораторной работы №5.

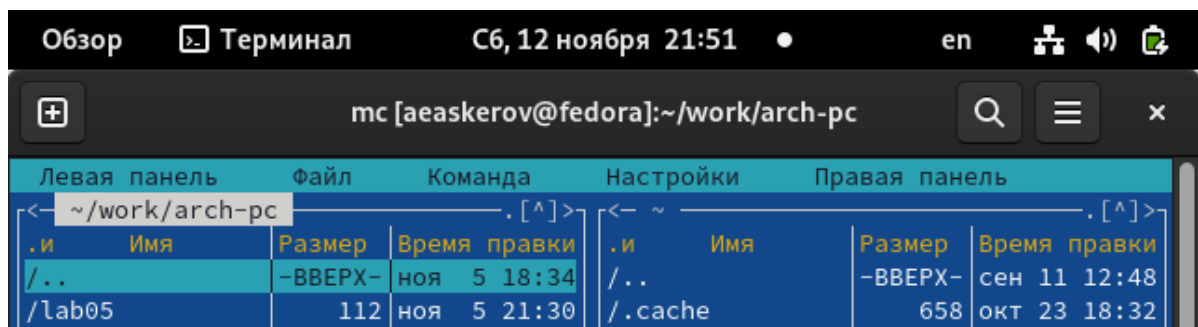


Рис. 2.2: Переход в каталог ~/work/archpc

3. С помощью функциональной клавиши F7 создадим папку lab06 и перейдём в созданный каталог.

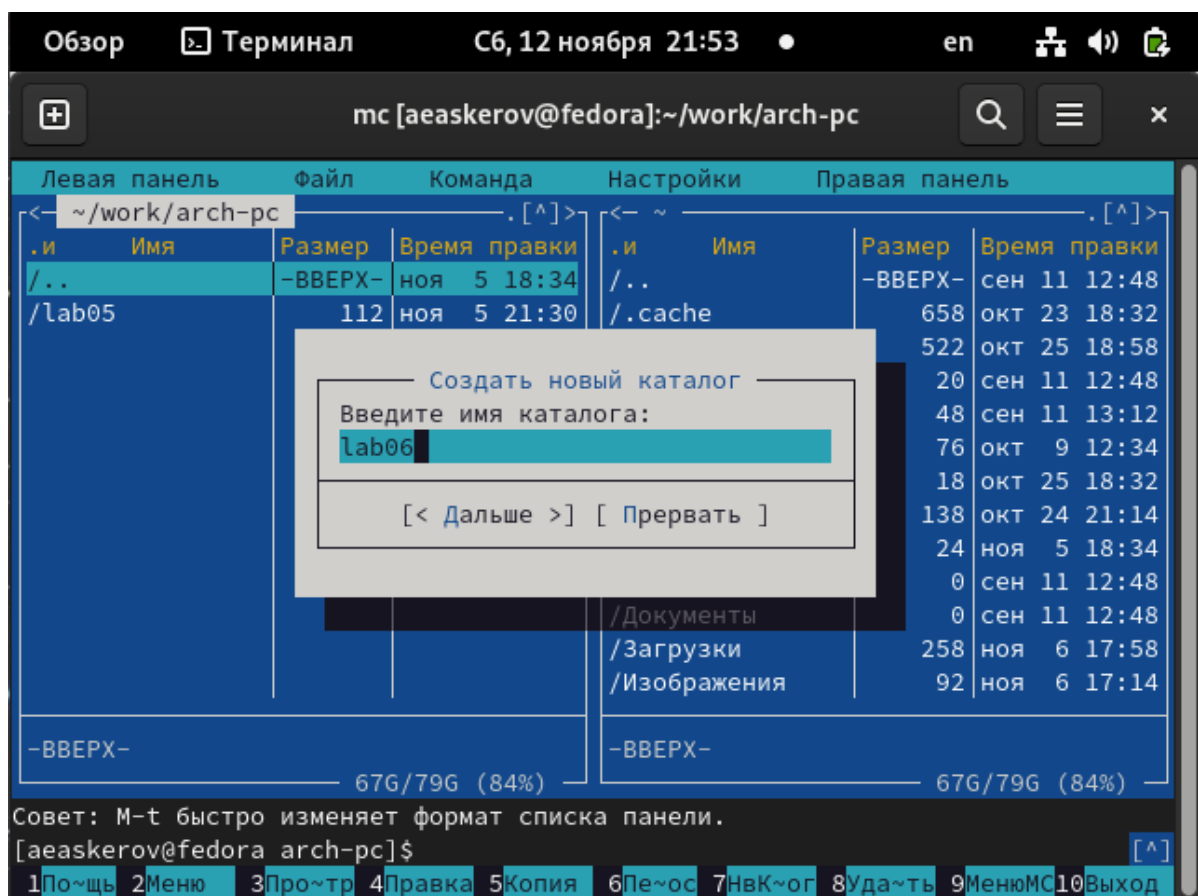


Рис. 2.3: Создание каталога lab06

4. Пользуясь строкой ввода и командой touch создадим файл lab6-1.asm.

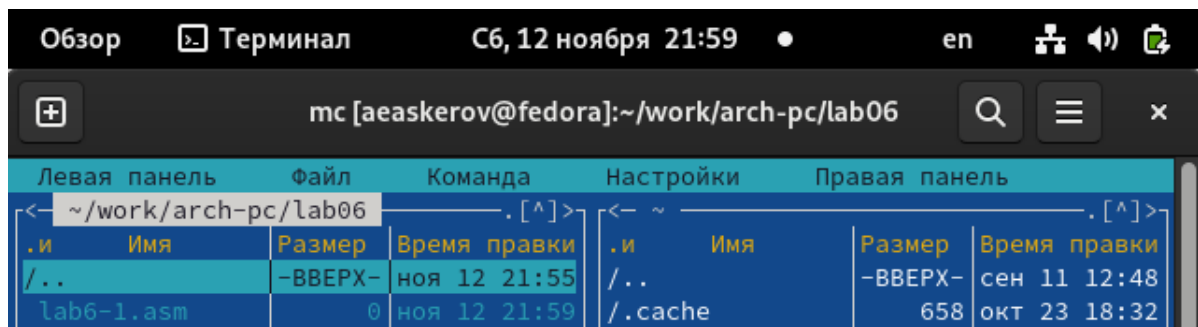


Рис. 2.4: Создание файла lab6-1.asm

5. С помощью функциональной клавиши F4 откроем файл lab6-1.asm для редактирования во встроенном редакторе.

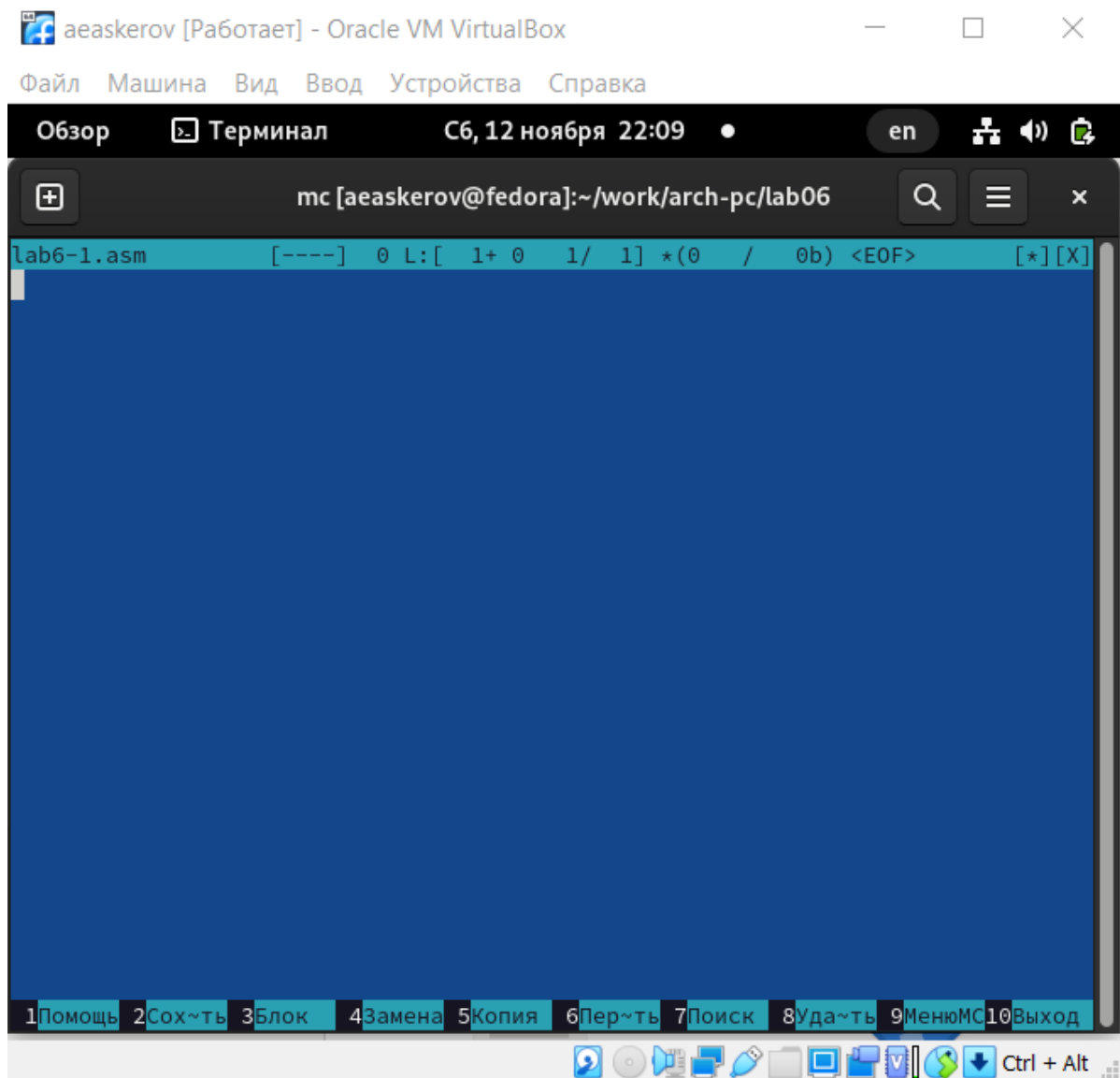
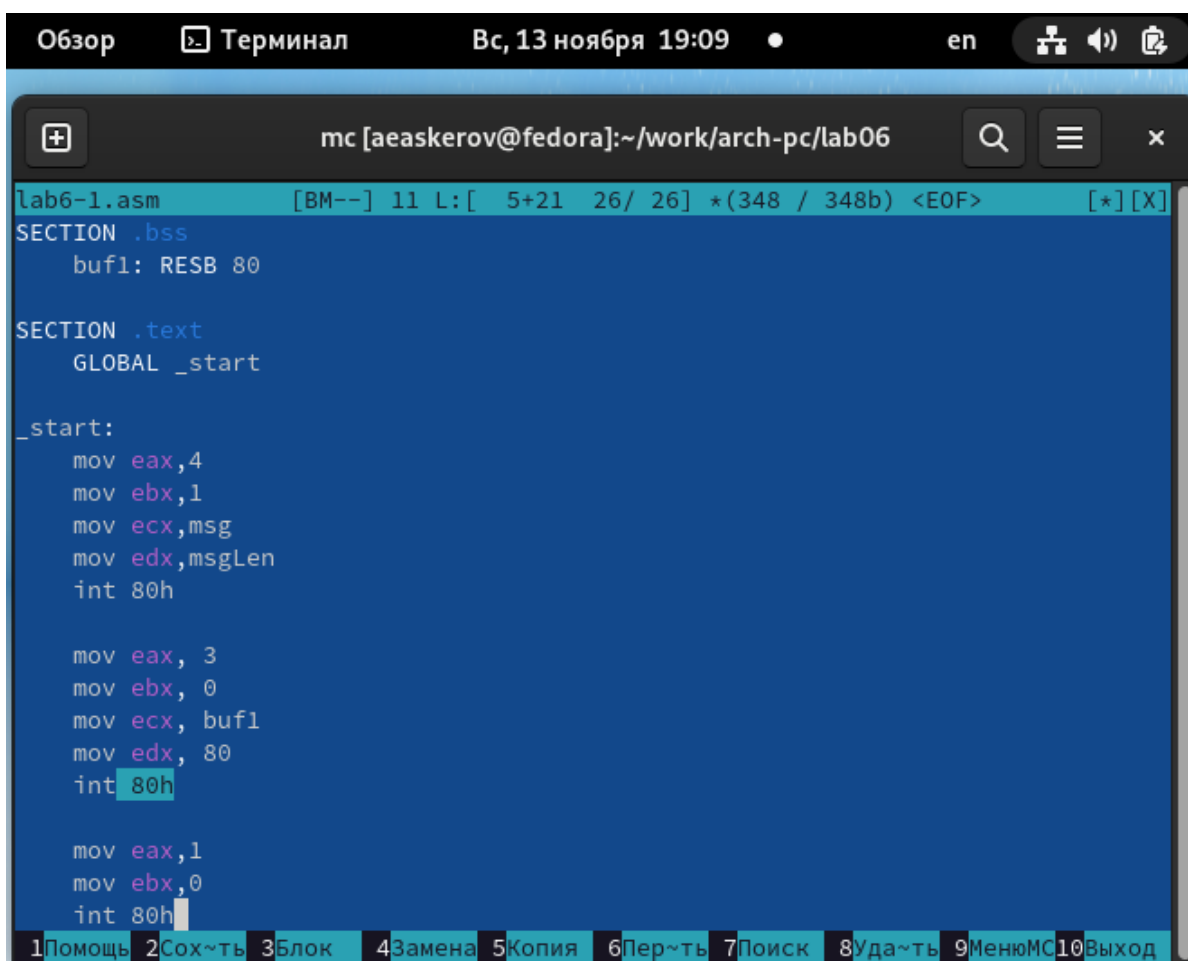


Рис. 2.5: Открытие файла lab6-1.asm в mcedit

6. Введём текст программы из листинга 6.1, сохраним изменения и закроем файл.



```
lab6-1.asm [BM--] 11 L: [ 5+21 26/ 26] *(348 / 348b) <EOF> [*] [X]
SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

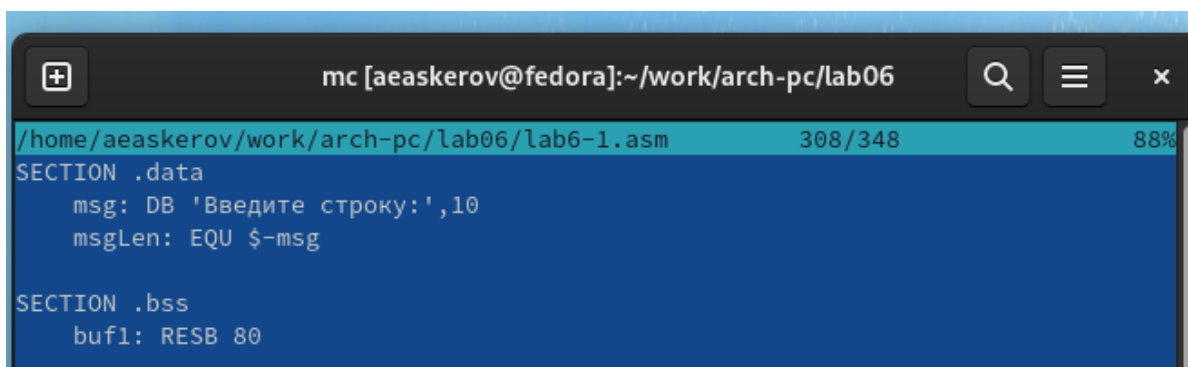
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

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

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС10Выход

Рис. 2.6: Текст программы

7. С помощью функциональной клавиши F3 откроем файл lab6-1.asm для просмотра. Убедимся, что файл содержит текст программы.



```
/home/aeaskerov/work/arch-pc/lab06/lab6-1.asm 308/348 88%
SECTION .data
    msg: DB 'Введите строку:',10
    msgLen: EQU $-msg

SECTION .bss
    buf1: RESB 80
```

Рис. 2.7: Проверка наличия текста программы

8. Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введём ФИО.

```
[aeaskerov@fedora lab06]$ nasm -f elf lab6-1.asm
[aeaskerov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[aeaskerov@fedora lab06]$ ./lab6-1
Введите строку:
Александр Аскеров
[aeaskerov@fedora lab06]$
```

Рис. 2.8: Создание объектного файла, его компоновка и ввод имени

2.1 Подключение внешнего файла in_out.asm

9. Скачаем файл in_out.asm со страницы курса в ТУИС.
10. Скопируем файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5.

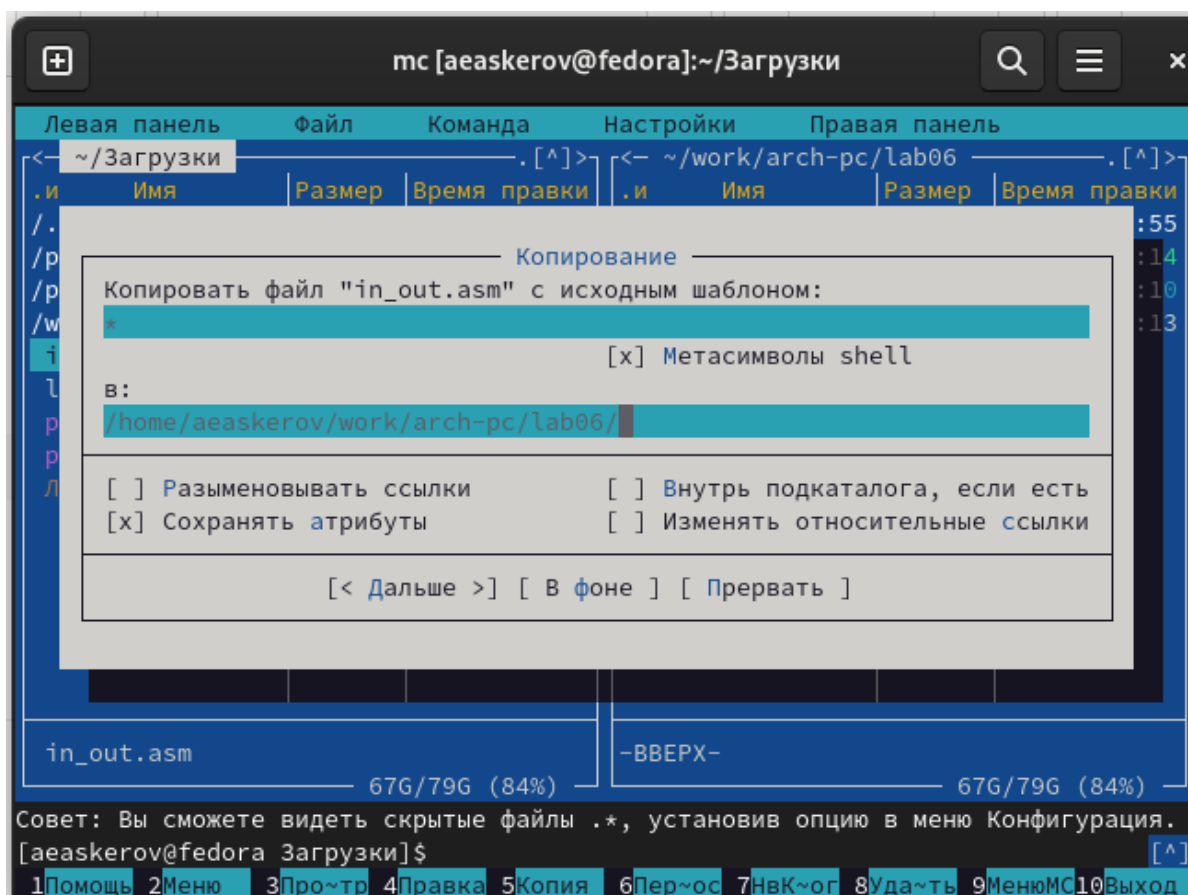


Рис. 2.9: Копирование файла in_out.asm в рабочий каталог

11. С помощью функциональной клавиши F6 создадим копию файла lab6-1.asm с именем lab6-2.asm. Выделим файл lab6-1.asm, нажмём клавишу F6, введём имя файла lab6-2.asm и нажмём клавишу Enter.

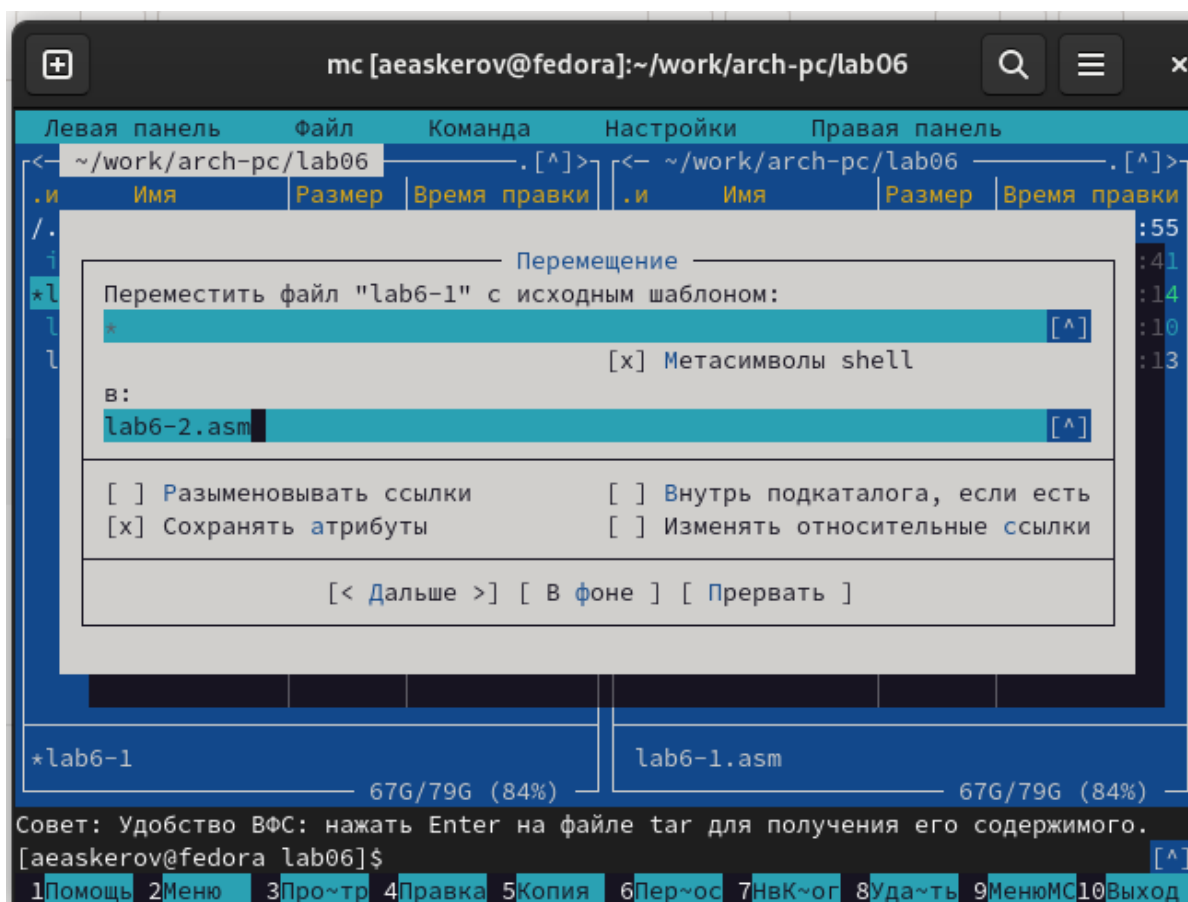
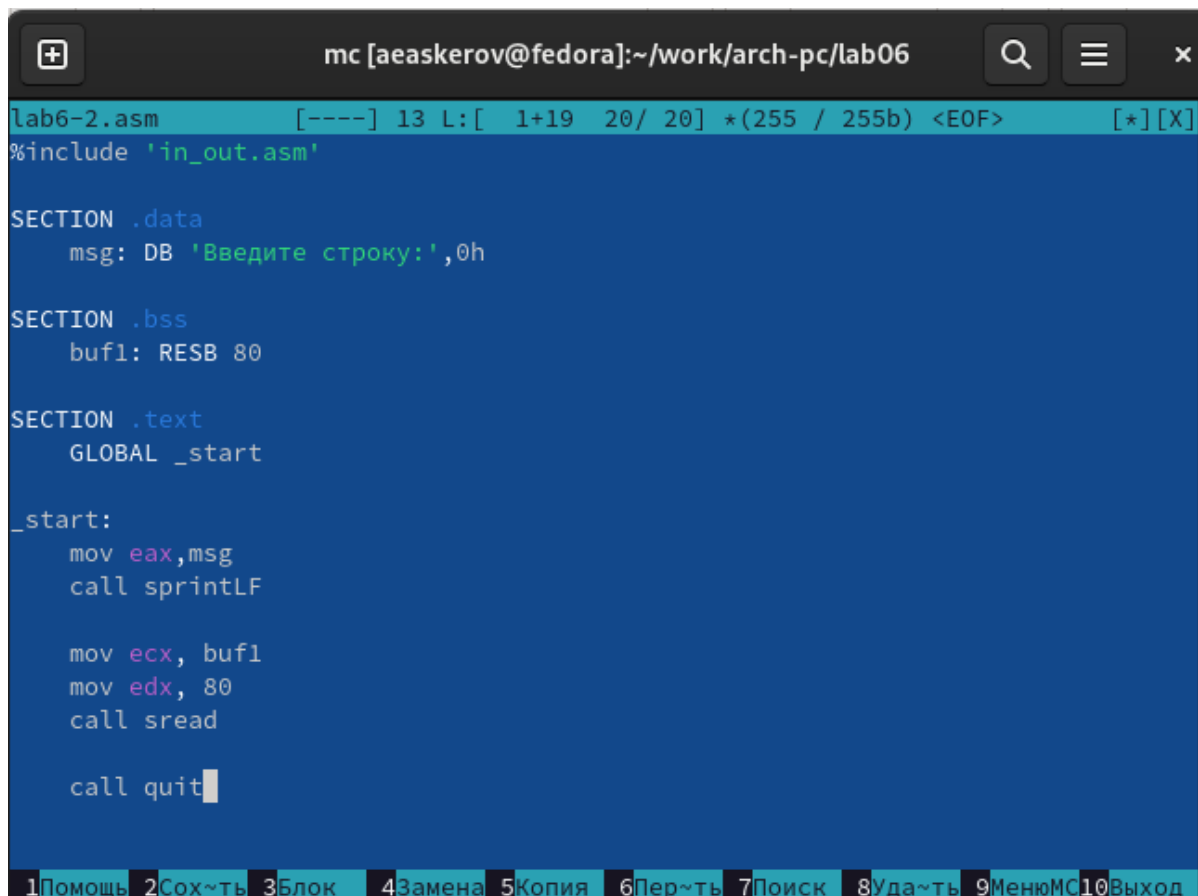


Рис. 2.10: Создание копированием файла lab6-2.asm

12. Исправим текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используем подпрограммы sprintf, sread и quit). Создадим исполняемый файл и проверим его работу.



```
lab6-2.asm [----] 13 L: [ 1+19 20/ 20] *(255 / 255b) <EOF> [*] [X]
#include 'in_out.asm'

SECTION .data
    msg: DB 'Введите строку:',0h

SECTION .bss
    buf1: RESB 80

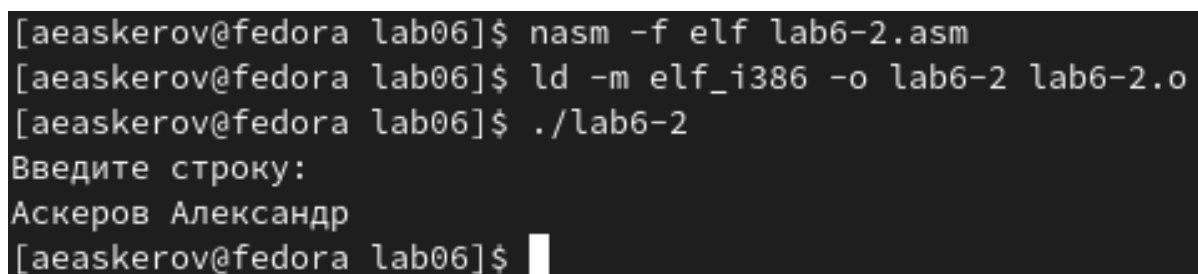
SECTION .text
    GLOBAL _start

_start:
    mov eax,msg
    call sprintLF

    mov ecx, buf1
    mov edx, 80
    call sread

    call quit
```

Рис. 2.11: Внесение изменений в файл lab6-2.asm



```
[aeaskerov@fedora lab06]$ nasm -f elf lab6-2.asm
[aeaskerov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[aeaskerov@fedora lab06]$ ./lab6-2
Введите строку:
Аскеров Александр
[aeaskerov@fedora lab06]$
```

Рис. 2.12: Убеждаемся в работоспособности программы

13. В файле lab6-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу. Разница в отсутствии переноса строки.

```
[aeaskerov@fedora lab06]$ ./lab6-2  
Введите строку:Александр Аскеров  
[aeaskerov@fedora lab06]$
```

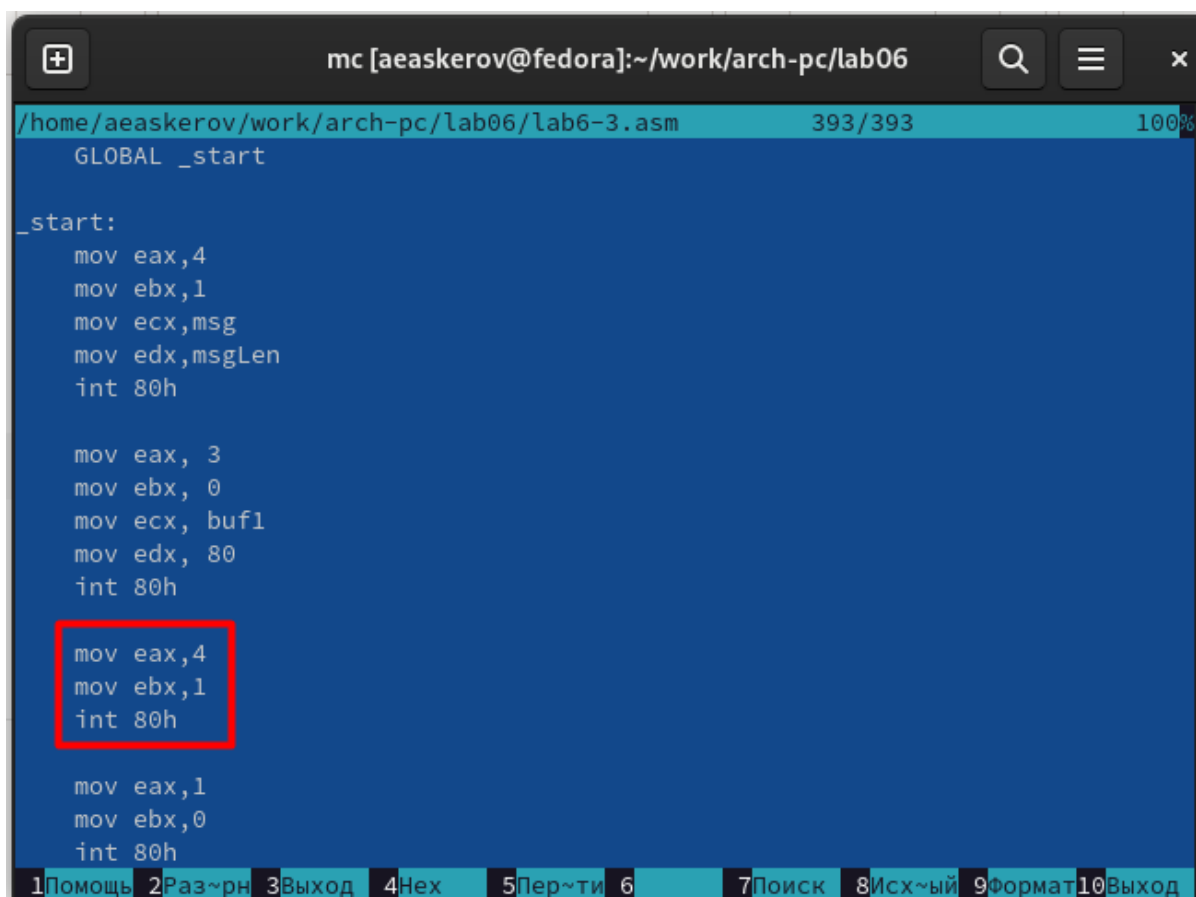
Рис. 2.13: Работа изменённой программы

2.2 Задания для самостоятельной работы

1. Создадим копию файла lab6-1.asm. Внесём 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 введённую строку на экран.

Регистры `ecx` и `edx` остались без изменений по той причине, что мы записали в них необходимые значения (текстовую строку и ограничение по длине, соответственно) в предыдущем блоке программы. После выполнения этого блока значения этих регистров не менялись.



```
mc [aeaskerov@fedora]:~/work/arch-pc/lab06
/home/aeaskerov/work/arch-pc/lab06/lab6-3.asm 393/393 100%
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

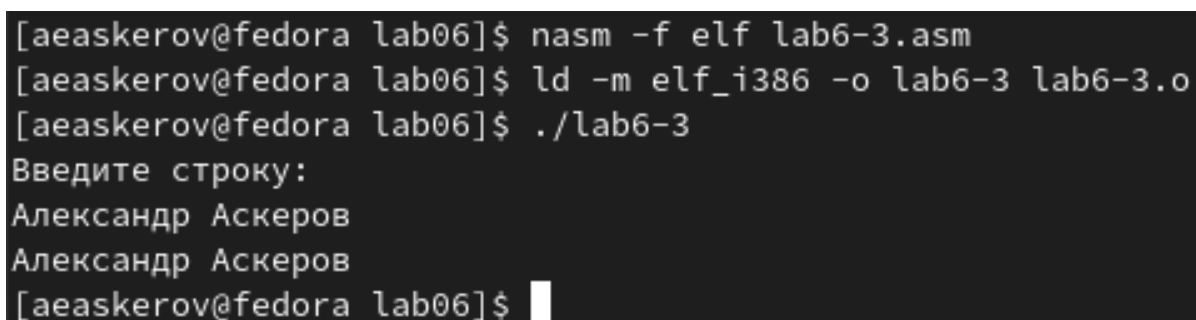
    mov eax, 4
    mov ebx, 1
    int 80h

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

1Помощь 2Развернуть 3Выход 4Hex 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

Рис. 2.14: Внесённые в копию файла lab6-1.asm изменения

2. Получим исполняемый файл lab6-3 и проверим его работу. На приглашение введём свои имя и фамилию.



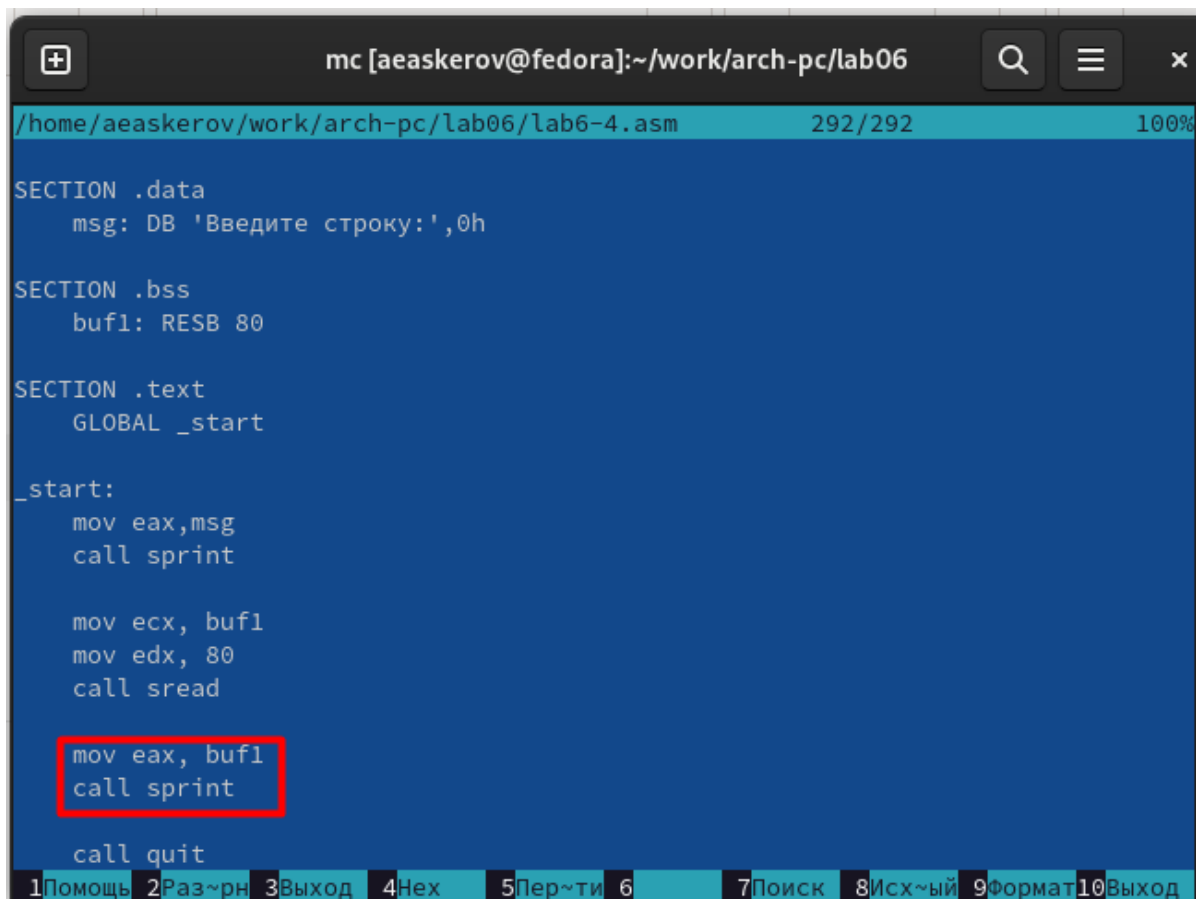
```
[aeaskerov@fedora lab06]$ nasm -f elf lab6-3.asm
[aeaskerov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[aeaskerov@fedora lab06]$ ./lab6-3
Введите строку:
Александр Аскеров
Александр Аскеров
[aeaskerov@fedora lab06]$
```

Рис. 2.15: Результат работы программы lab6-3.asm

3. Создадим копию файла lab6-2.asm. Исправим текст программы с использованием 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.



```
mc [aeaskerov@fedora]:~/work/arch-pc/lab06
/home/aeaskerov/work/arch-pc/lab06/lab6-4.asm 292/292 100%

SECTION .data
    msg: DB 'Введите строку:',0h

SECTION .bss
    buf1: RESB 80

SECTION .text
    GLOBAL _start

_start:
    mov eax,msg
    call sprint

    mov ecx, buf1
    mov edx, 80
    call sread

    mov eax, buf1
    call sprint

    call quit
```

1Помощь 2Развернуть 3Выход 4Hex 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

Рис. 2.16: Внесённые в копию файла lab6-2.asm изменения

4. Создадим исполняемый файл lab6-4 и проверим его работу.


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

Рис. 2.17: Результат работы программы lab6-4.asm

3 Выводы

Приобретены практические навыки работы в Midnight Commander. Освоены инструкции языка ассемблера `mov` и `int`.