

# **Отчёт по лабораторной работе 5**

**Архитектура компьютера**

Саид Курбанов

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Знакомство с Midnight Commander . . . . .	6
2.2	Подключение внешнего файла in_out.asm . . . . .	11
2.3	Задание для самостоятельной работы . . . . .	16
<b>3</b>	<b>Выводы</b>	<b>20</b>

## Список иллюстраций

2.1	Запуск Midnight Commander . . . . .	6
2.2	Создание каталога . . . . .	7
2.3	Создание файла lab05-1.asm . . . . .	8
2.4	Программа в файле lab05-1.asm . . . . .	9
2.5	Просмотр файла lab05-1.asm . . . . .	10
2.6	Запуск программы lab05-1.asm . . . . .	11
2.7	Копирование файла in_out.asm . . . . .	12
2.8	Копирование файла lab05-1.asm . . . . .	13
2.9	Программа в файле lab05-2.asm . . . . .	14
2.10	Запуск программы lab05-2.asm . . . . .	14
2.11	Программа в файле lab05-2.asm . . . . .	15
2.12	Запуск программы lab05-2.asm . . . . .	15
2.13	Копирование файла lab05-1.asm . . . . .	16
2.14	Программа в файле lab05-3.asm . . . . .	17
2.15	Запуск программы lab05-3.asm . . . . .	17
2.16	Копирование файла lab05-2.asm . . . . .	18
2.17	Программа в файле lab05-4.asm . . . . .	19
2.18	Запуск программы lab05-4.asm . . . . .	19

## **Список таблиц**

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Выполнение лабораторной работы

### 2.1 Знакомство с Midnight Commander

Открыл Midnight Commander, с помощью клавишь со стрелками и Enter перешел в каталог ~/work/arch-pc. Далее нажал F7 и создал каталог lab05

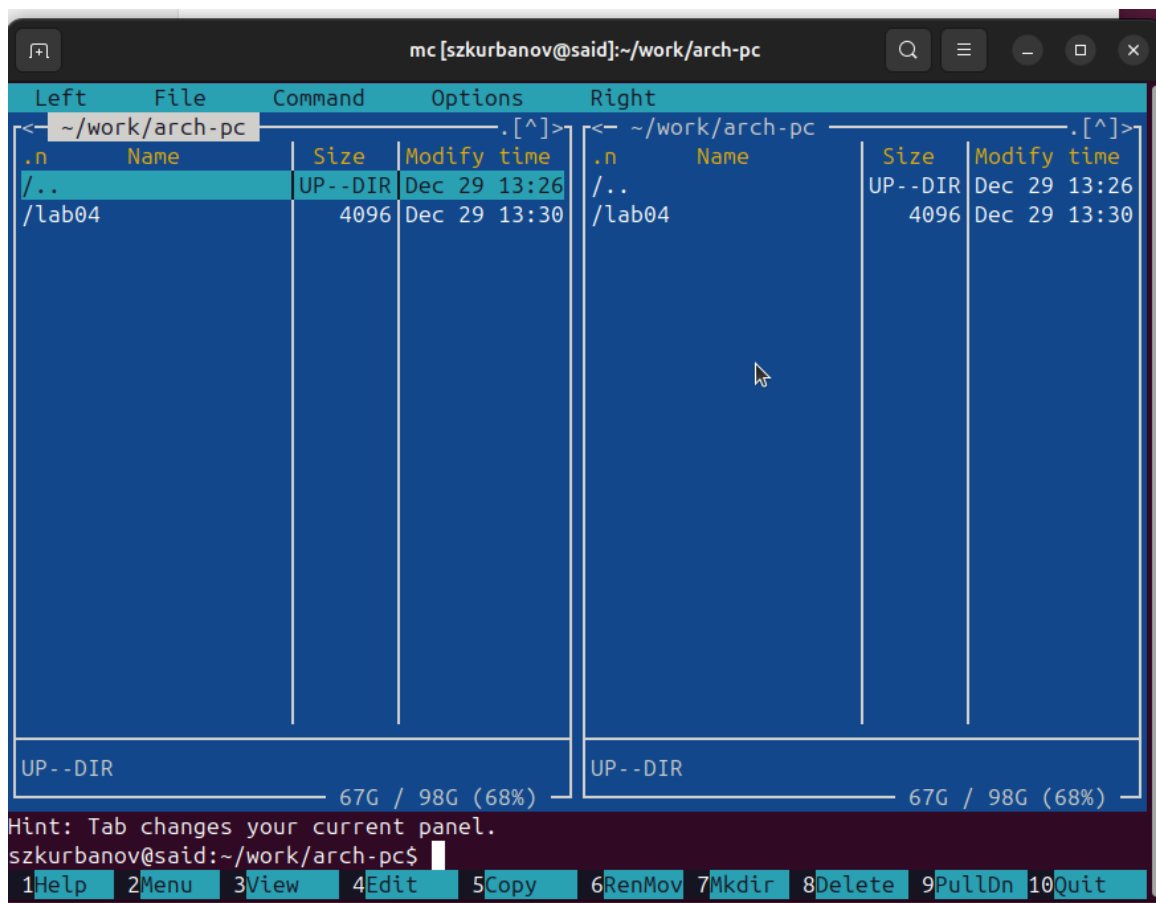


Рисунок 2.1: Запуск Midnight Commander

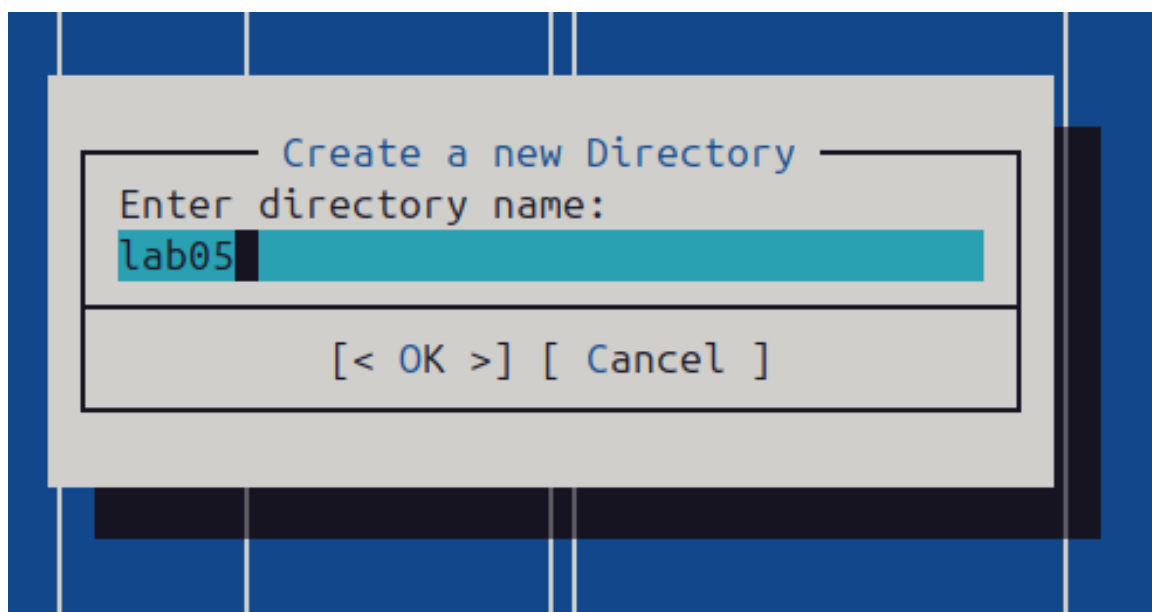


Рисунок 2.2: Создание каталога

При помощи touch создал файл lab05-1.asm

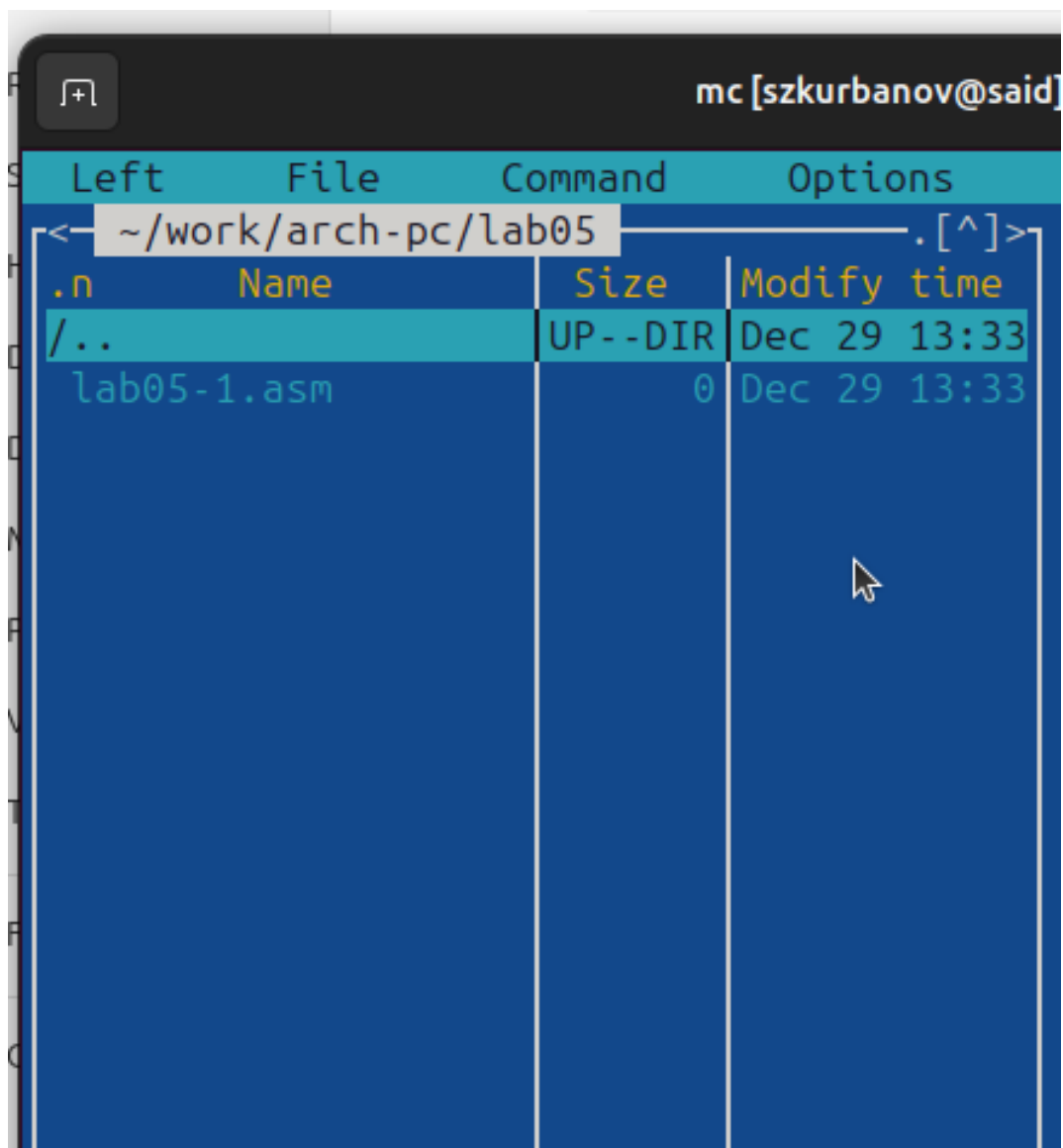
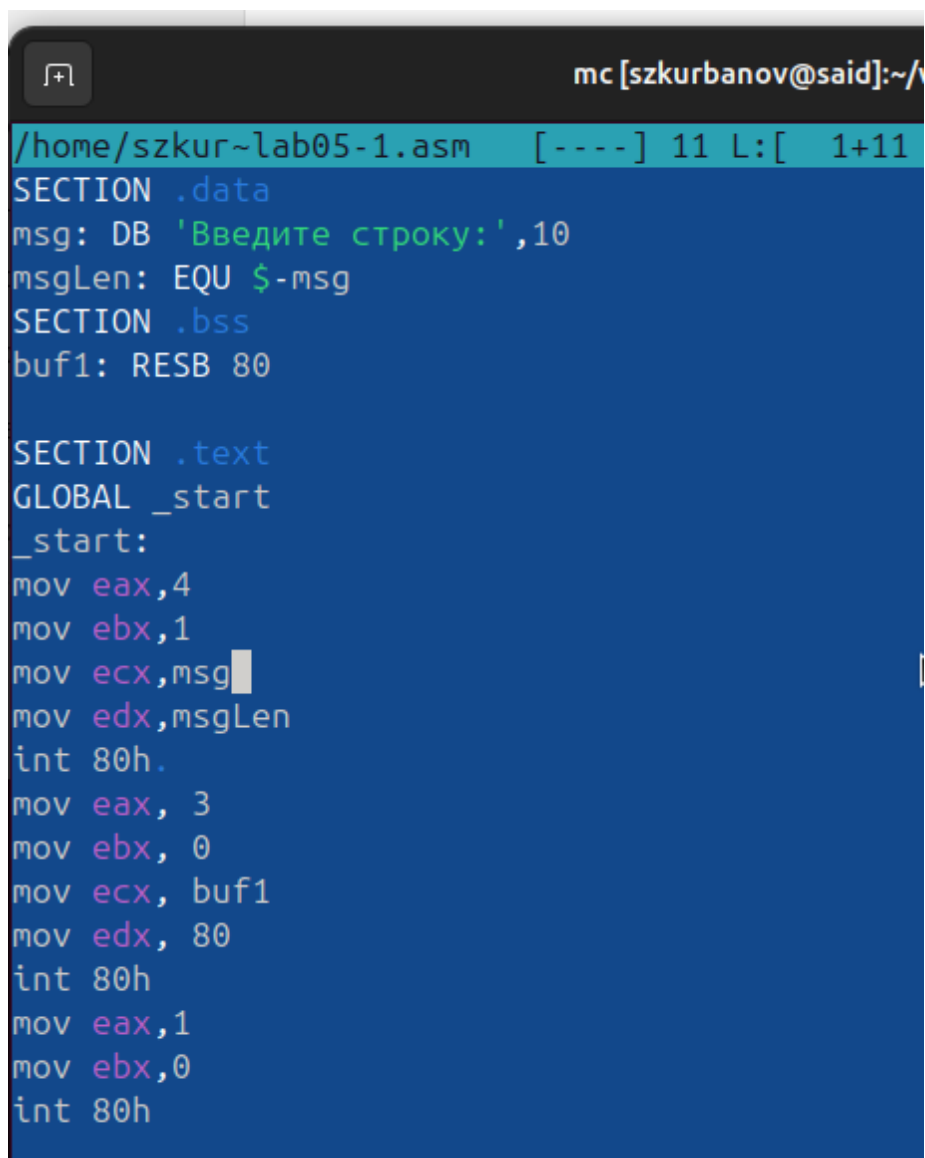


Рисунок 2.3: Создание файла lab05-1.asm

Открыл файл на редактирование клавишей F4, выбрал редактор mceditor, написал код программы из задания.

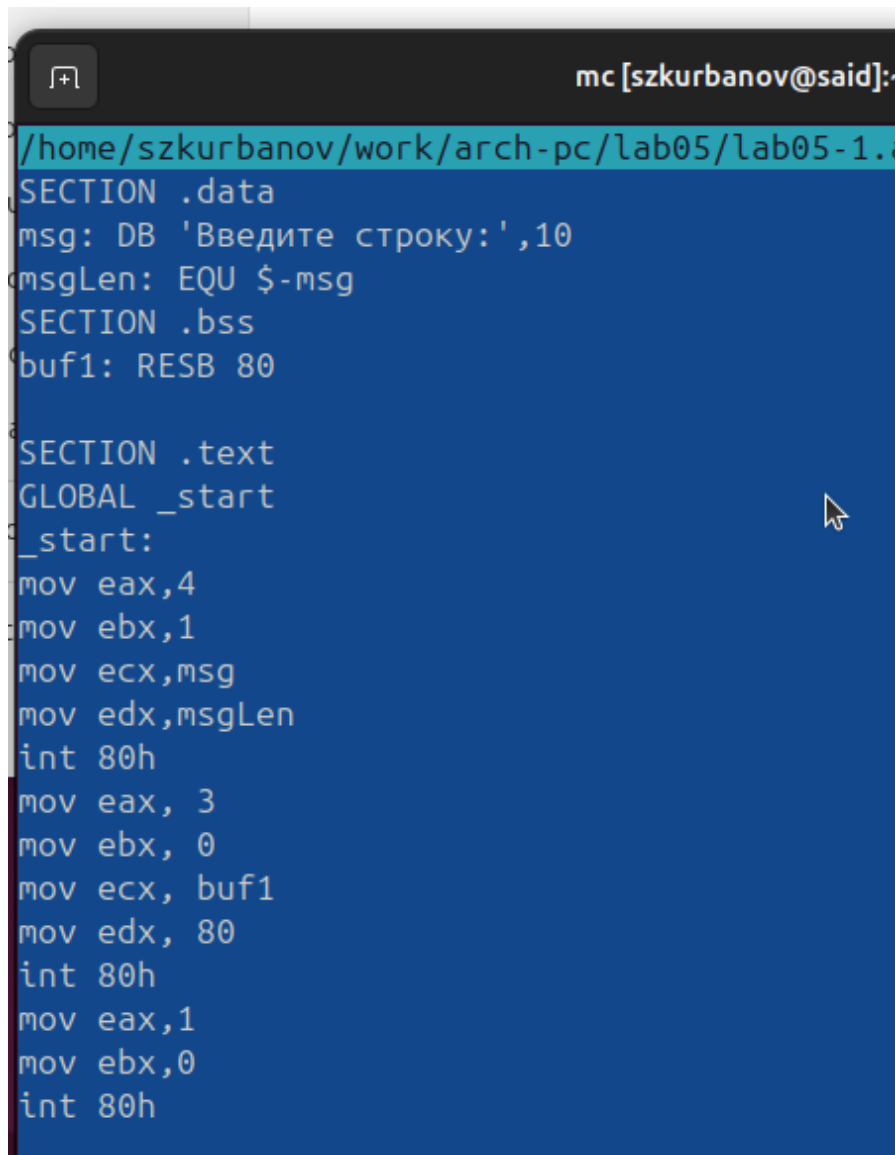




```
mc [szkurbanov@said]:~/ /  
/home/szkur~lab05-1.asm [----] 11 L:[ 1+11  
SECTION .data  
msg: DB 'Введите строку:',10  
msgLen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msgLen  
int 80h  
mov eax, 3  
mov ebx, 0  
mov ecx, buf1  
mov edx, 80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h
```

Рисунок 2.4: Программа в файле lab05-1.asm

Открыл файл на просмотр клавишей F3 и убелился, что он содержит набран-  
ный код.



```
mc [szkurbanov@said]:-
/home/szkurbanov/work/arch-pc/lab05/lab05-1.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,1
mov ebx,0
int 80h
```

Рисунок 2.5: Просмотр файла lab05-1.asm

Транслировал файл программы в объектный файл, выполнил компоновку объектного файла, получил исполняемый файл программы и проверил ее работу.

```
szkurbanov@said:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
szkurbanov@said:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
szkurbanov@said:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Said
szkurbanov@said:~/work/arch-pc/lab05$
```

Рисунок 2.6: Запуск программы lab05-1.asm

## 2.2 Подключение внешнего файла in\_out.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

Для выполнения лабораторных работ используется файл in\_out.asm, который содержит следующие подпрограммы:

- `slen` – вычисление длины строки (используется в подпрограммах печати сообщения для определения количества выводимых байтов);
- `sprint` – вывод сообщения на экран, перед вызовом `sprint` в регистр `eax` необходимо записать выводимое сообщение (`mov eax,;`);
- `sprintLF` – работает аналогично `sprint`, но при выводе на экран добавляет к сообщению символ перевода строки;
- `sread` – ввод сообщения с клавиатуры, перед вызовом `sread` в регистр `eax` необходимо записать адрес переменной в которую введенное сообщение будет записано (`mov eax,;`), в регистр `ebx` – длину вводимой строки (`mov ebx,;`);

- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax` необходимо записать выводимое число (`mov eax,;`);
- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добавляет к символ перевода строки;
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`, перед вызовом `atoi` в регистр `eax` необходимо записать число (`mov eax,;`);
- `quit` – завершение программы.

Скачал файл `in_out.asm` и разместил его в рабочем каталоге. Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

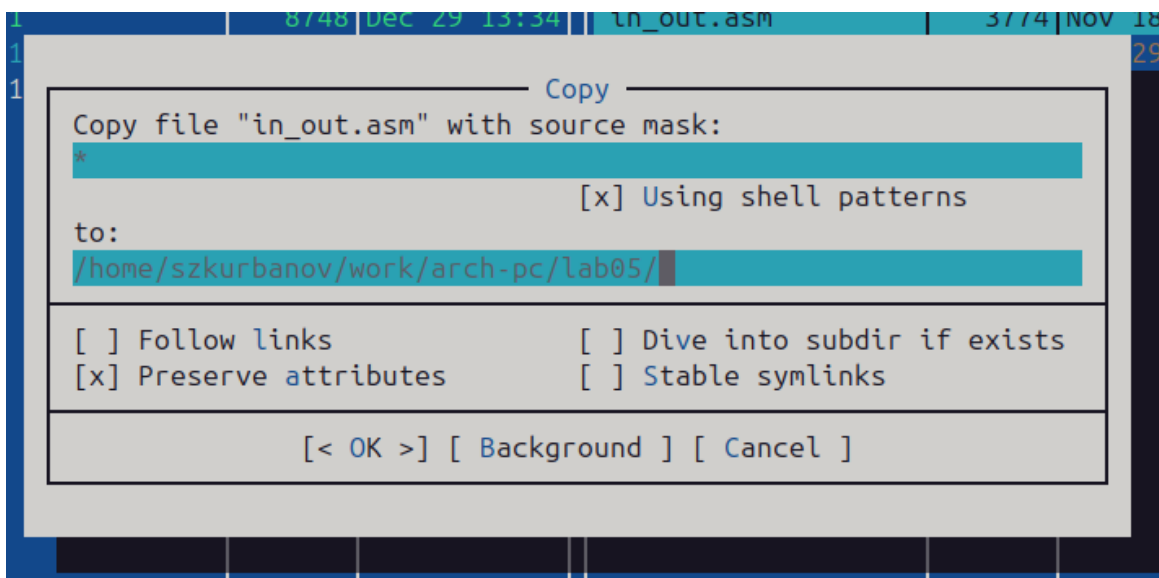


Рисунок 2.7: Копирование файла `in_out.asm`

Скопировал `lab05-1.asm` в `lab05-2.asm`.

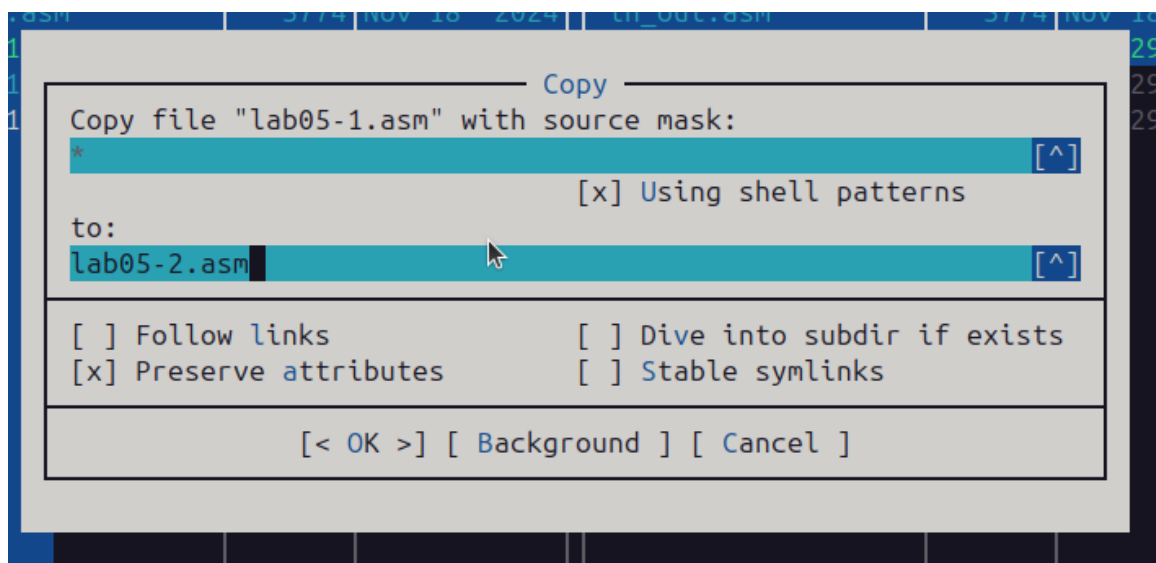
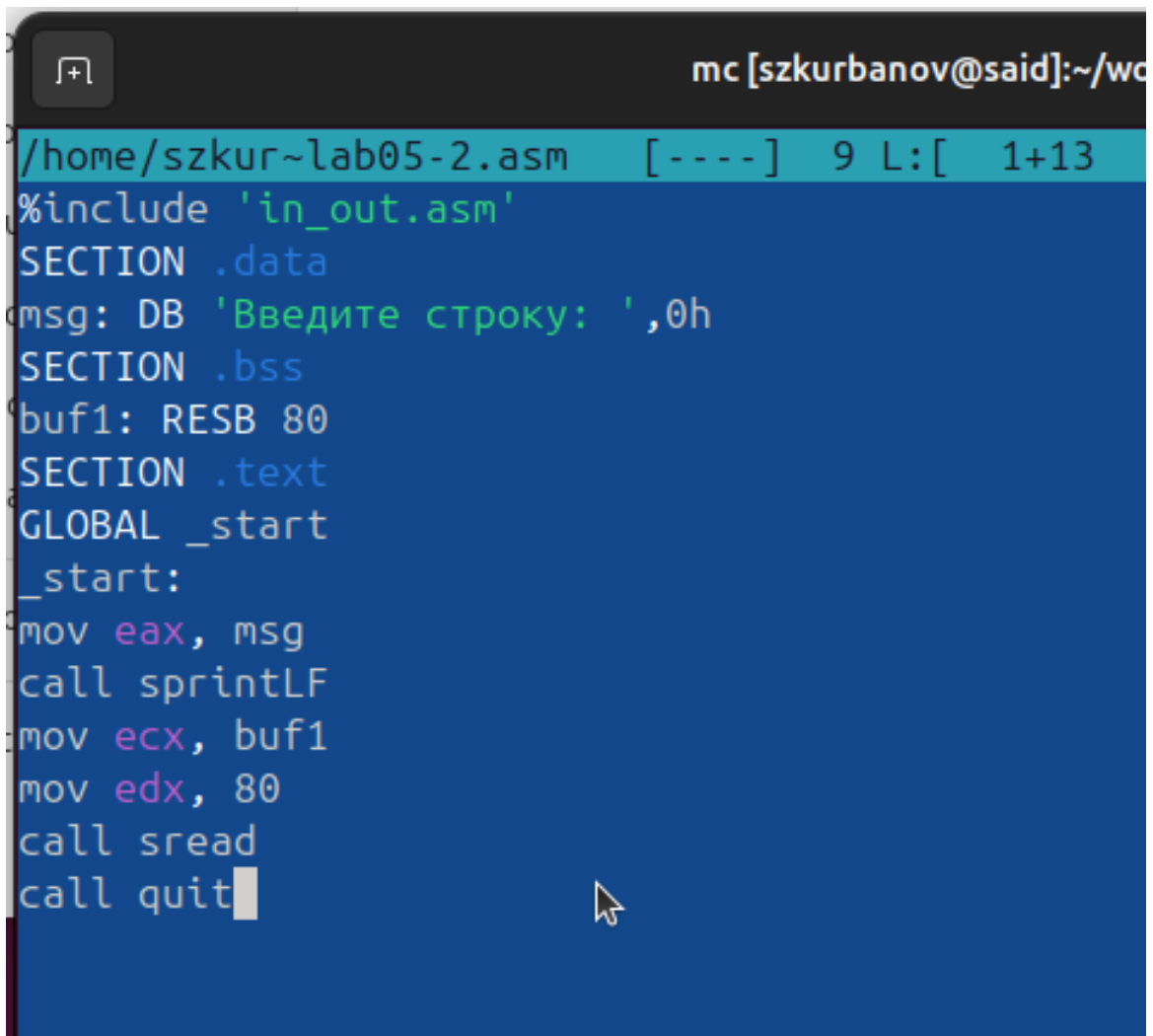


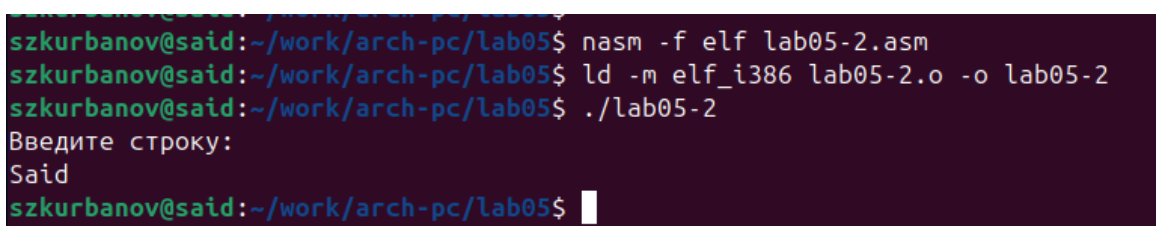
Рисунок 2.8: Копирование файла lab05-1.asm

Написал код программы lab05-2.asm с использованием подпрограмм из внешнего файла in\_out.asm . Скомпилировал программу и проверил запуск.



```
mc [szkurbanov@said]:~/wo
/home/szkur~lab05-2.asm [----] 9 L:[ 1+13
#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
```

Рисунок 2.9: Программа в файле lab05-2.asm

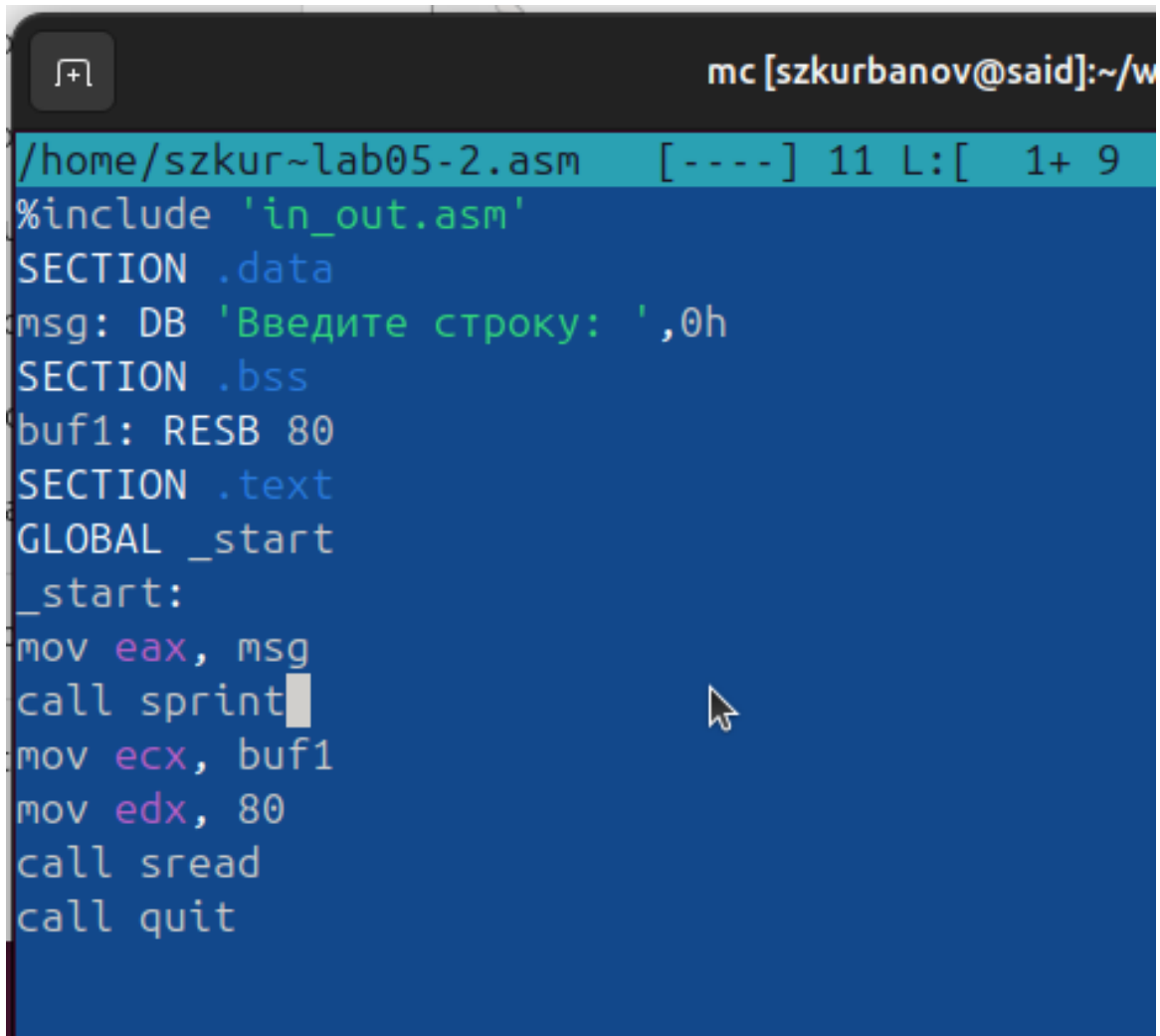


```
szkurbanov@said:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
szkurbanov@said:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
szkurbanov@said:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Said
szkurbanov@said:~/work/arch-pc/lab05$
```

Рисунок 2.10: Запуск программы lab05-2.asm

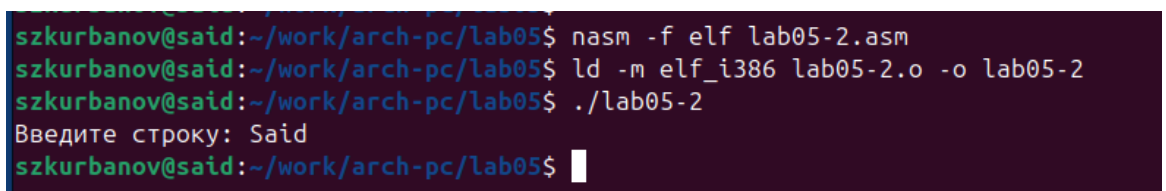
В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Заново собрал исполняемый файл. Теперь после вывода строки она не завершается символом

перехода на новую строку.



```
mc [szkurbanov@said]:~/w
/home/szkur~lab05-2.asm  [ - - - - ] 11 L:[ 1+ 9
%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 sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рисунок 2.11: Программа в файле lab05-2.asm



```
szkurbanov@said:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
szkurbanov@said:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
szkurbanov@said:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: Said
szkurbanov@said:~/work/arch-pc/lab05$
```

Рисунок 2.12: Запуск программы lab05-2.asm

## 2.3 Задание для самостоятельной работы

Скопировал программу lab05-1.asm и изменил код, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа «Введите строку:»;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.

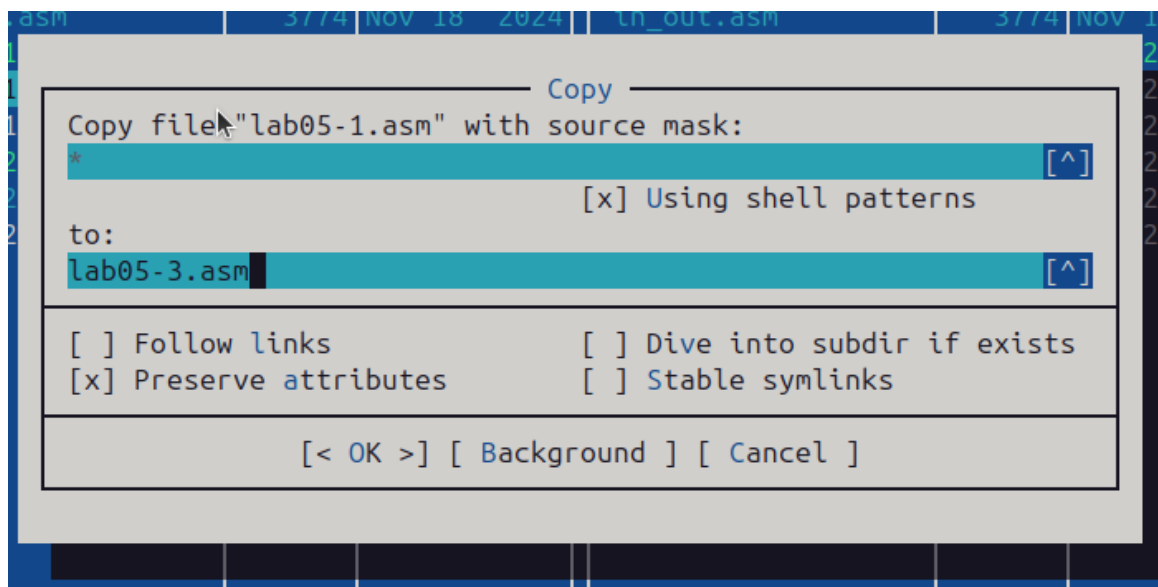


Рисунок 2.13: Копирование файла lab05-1.asm



```

mc [szkurbanov@said]:~/work/arch-pc/lab05$ cat lab05-3.asm
/home/szkur~lab05-3.asm  [ - - - ]  8 L: [  1+13  14/ 28 ] *(
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

```

Рисунок 2.14: Программа в файле lab05-3.asm

```

szkurbanov@said:~/work/arch-pc/lab05$
szkurbanov@said:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
szkurbanov@said:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
szkurbanov@said:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Said
Said
szkurbanov@said:~/work/arch-pc/lab05$

```

Рисунок 2.15: Запуск программы lab05-3.asm

Аналогично скопировал программу lab05-2.asm и изменил код, но теперь использовал подпрограммы из файла in\_out.asm.

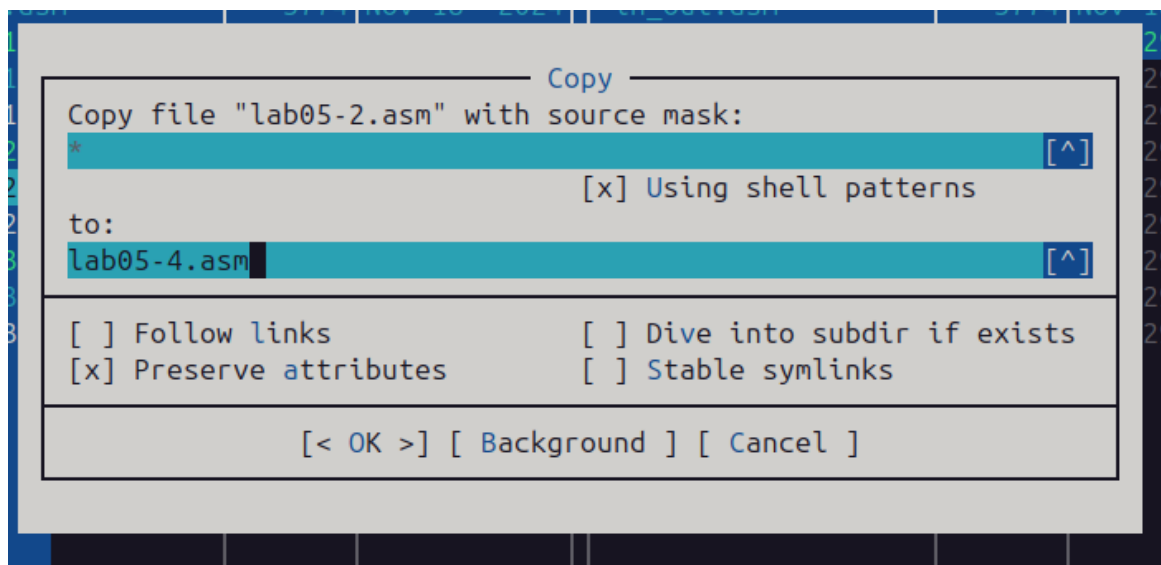
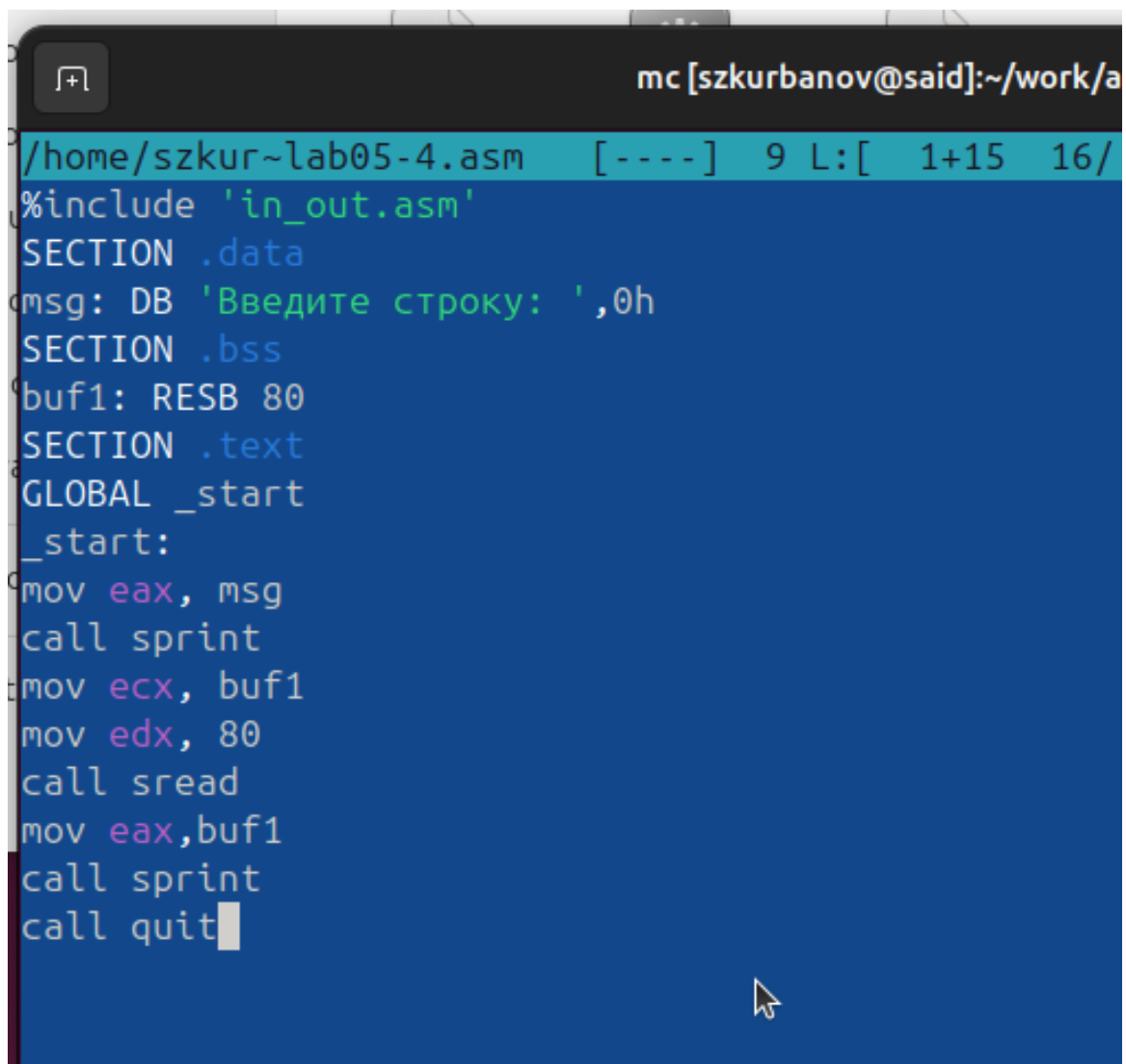
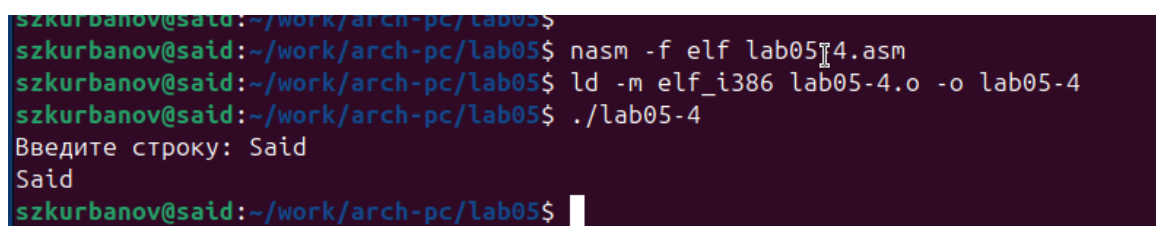


Рисунок 2.16: Копирование файла lab05-2.asm



```
mc [szkurbanov@said]:~/work/a
/home/szkur~lab05-4.asm [----] 9 L:[ 1+15 16/
#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 sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, buf1
call sprint
call quit
```

Рисунок 2.17: Программа в файле lab05-4.asm



```
szkurbanov@said:~/work/arch-pc/lab05$
szkurbanov@said:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
szkurbanov@said:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
szkurbanov@said:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: Said
Said
szkurbanov@said:~/work/arch-pc/lab05$
```

Рисунок 2.18: Запуск программы lab05-4.asm

## 3 Выводы

Научились писать базовые ассемблерные программы. Освоили ассемблерные инструкции `mov` и `int`.