

Отчёт по лабораторной работе 5

Архитектура ЭВМ

Шеенкова Мария

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Знакомство с Midnight Commander	7
3.2	Подключение внешнего файла in_out.asm	12
3.3	Задание для самостоятельной работы	17
4	Выводы	21
	Список литературы	22

Список иллюстраций

3.1	Запуск Midnight Commander	7
3.2	Создание каталога	8
3.3	Создание файла lab05-1.asm	9
3.4	Программа lab05-1.asm	10
3.5	Просмотр файла lab05-1.asm	11
3.6	Запуск программы lab05-1.asm	12
3.7	Копирование файла in_out.asm	13
3.8	Копирование файла lab05-1.asm	14
3.9	Программа lab05-2.asm	15
3.10	Запуск программы lab05-2.asm	15
3.11	Программа в файле lab05-2.asm	16
3.12	Запуск программы lab05-2.asm	16
3.13	Программа lab05-3.asm	18
3.14	Запуск программы lab05-3.asm	18
3.15	Программа lab05-4.asm	19
3.16	Запуск программы lab05-4.asm	20

Список таблиц

1 Цель работы

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

2 Задание

1. Установить Midnight Commander
2. Изучить структуру программ
3. Изучить файл in_out.asm
4. Дополнить программы по заданию.

3 Выполнение лабораторной работы

3.1 Знакомство с Midnight Commander

Открываю Midnight Commander (рис. 3.1), с помощью клавиш со стрелками и Enter перехожу в каталог `~/work/arch-pc`. Далее нажимаю F7 и создаю каталог `lab05` (рис. 3.2).

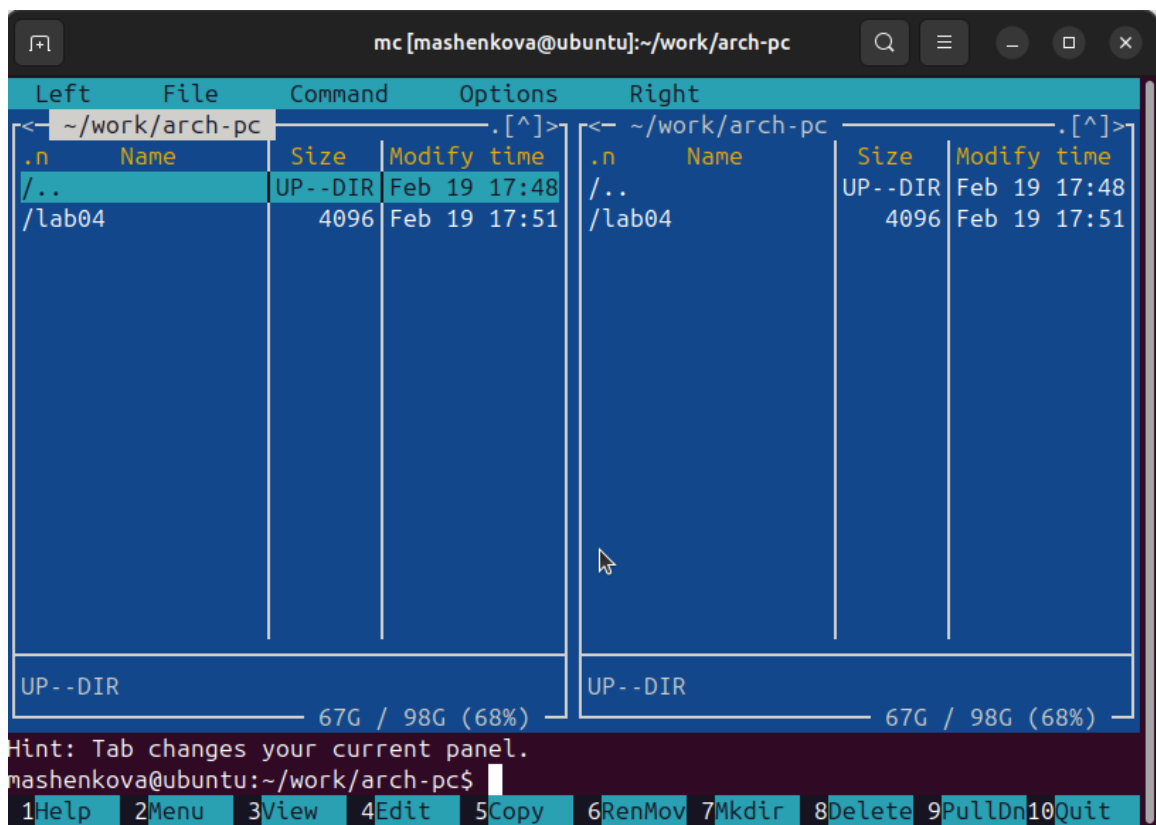


Рисунок 3.1: Запуск Midnight Commander

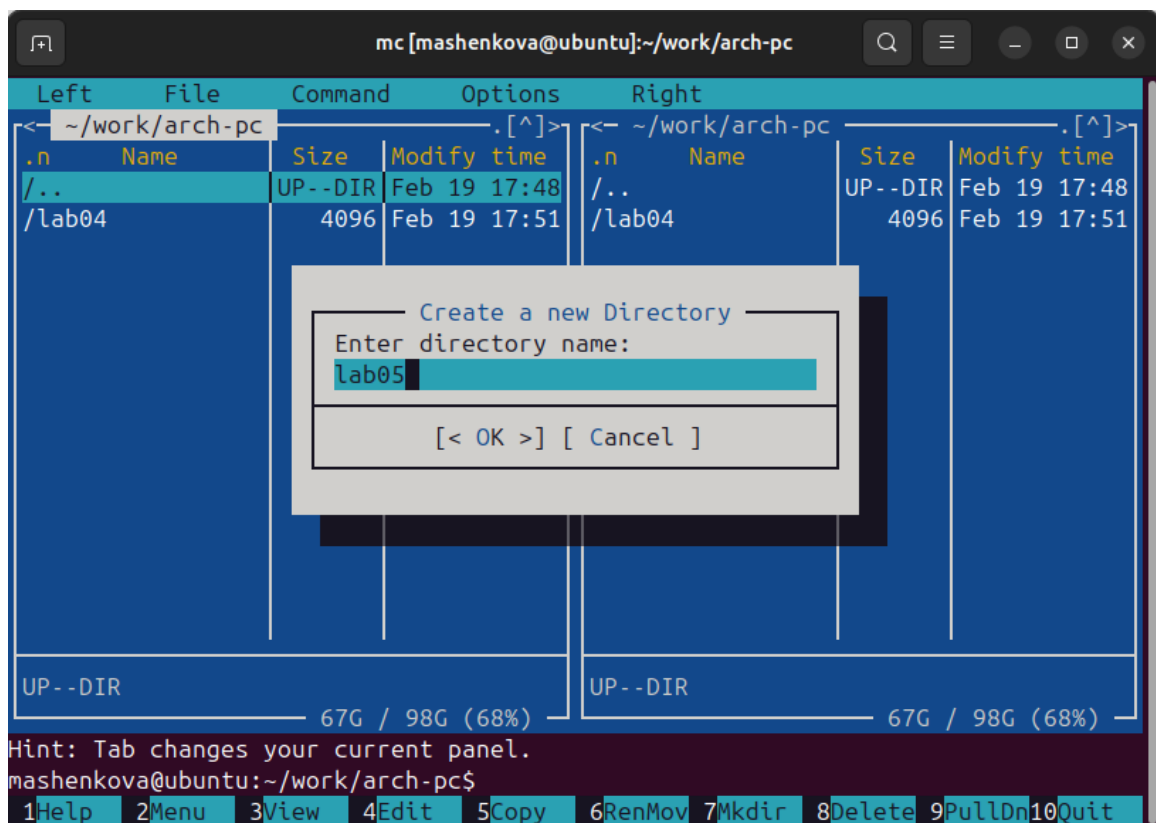


Рисунок 3.2: Создание каталога

При помощи `touch` создаю файл `lab05-1.asm` (рис. 3.3)

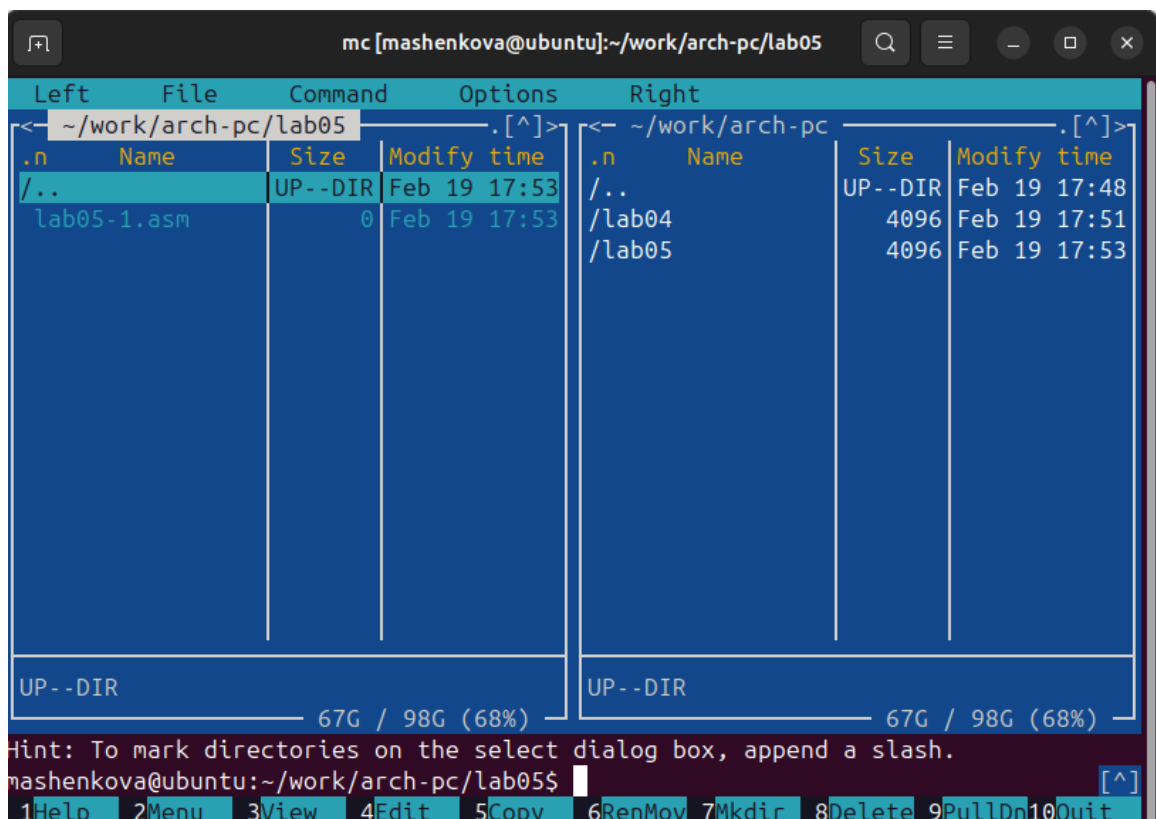
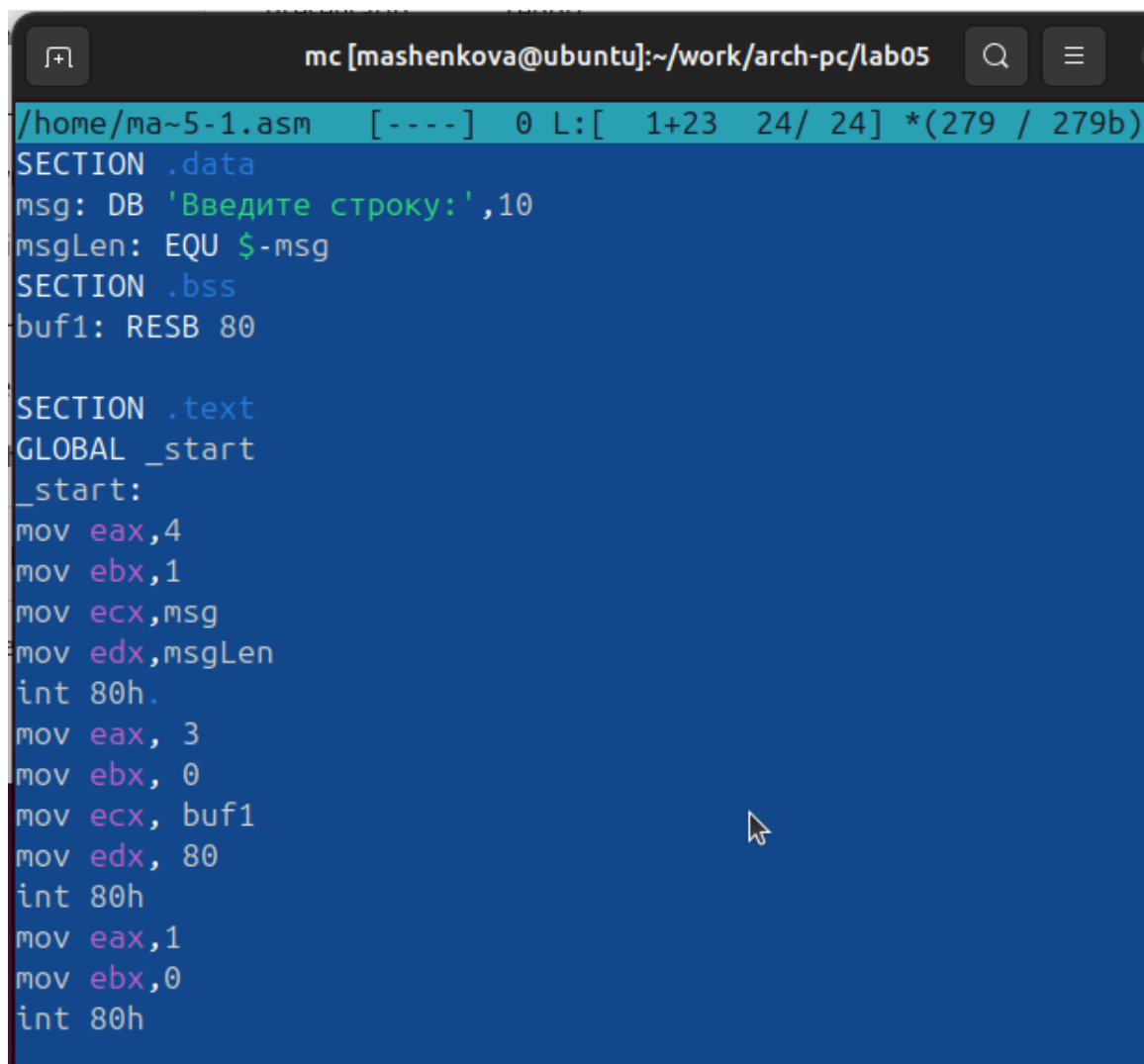


Рисунок 3.3: Создание файла lab05-1.asm

Открываю файл на редактирование клавишей F4, выбираю редактор mcedit, пишу код программы из задания. (рис. 3.4)

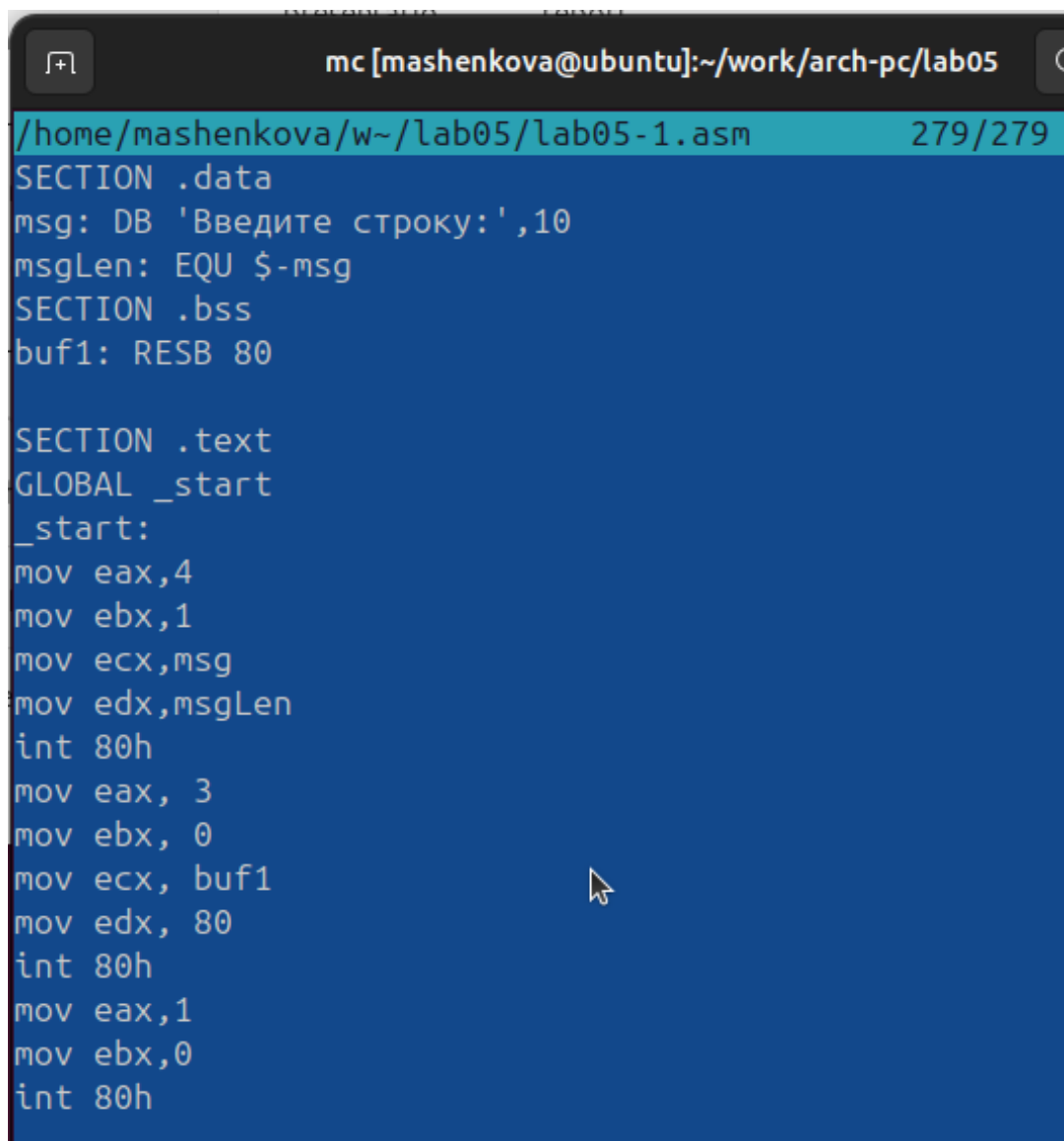


```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/ma~5-1.asm [----] 0 L:[ 1+23 24/ 24] *(279 / 279b)
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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 3.4: Программа lab05-1.asm

Открываю файл на просмотр клавишей F3 и проверяю, что он содержит набранный код. (рис. 3.5)



```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/mashenkova/w~/lab05/lab05-1.asm 279/279
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 3.5: Просмотр файла lab05-1.asm

Транслирую файл программы в объектный файл, выполняю компоновку объектного файла, получился исполняемый файл программы. (рис. 3.6)

```
mashenkova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
mashenkova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
mashenkova@ubuntu:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
linux
mashenkova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 3.6: Запуск программы lab05-1.asm

3.2 Подключение внешнего файла in_out.asm

Скачиваю файл in_out.asm и размещаю его в рабочем каталоге. (рис. 3.7) Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

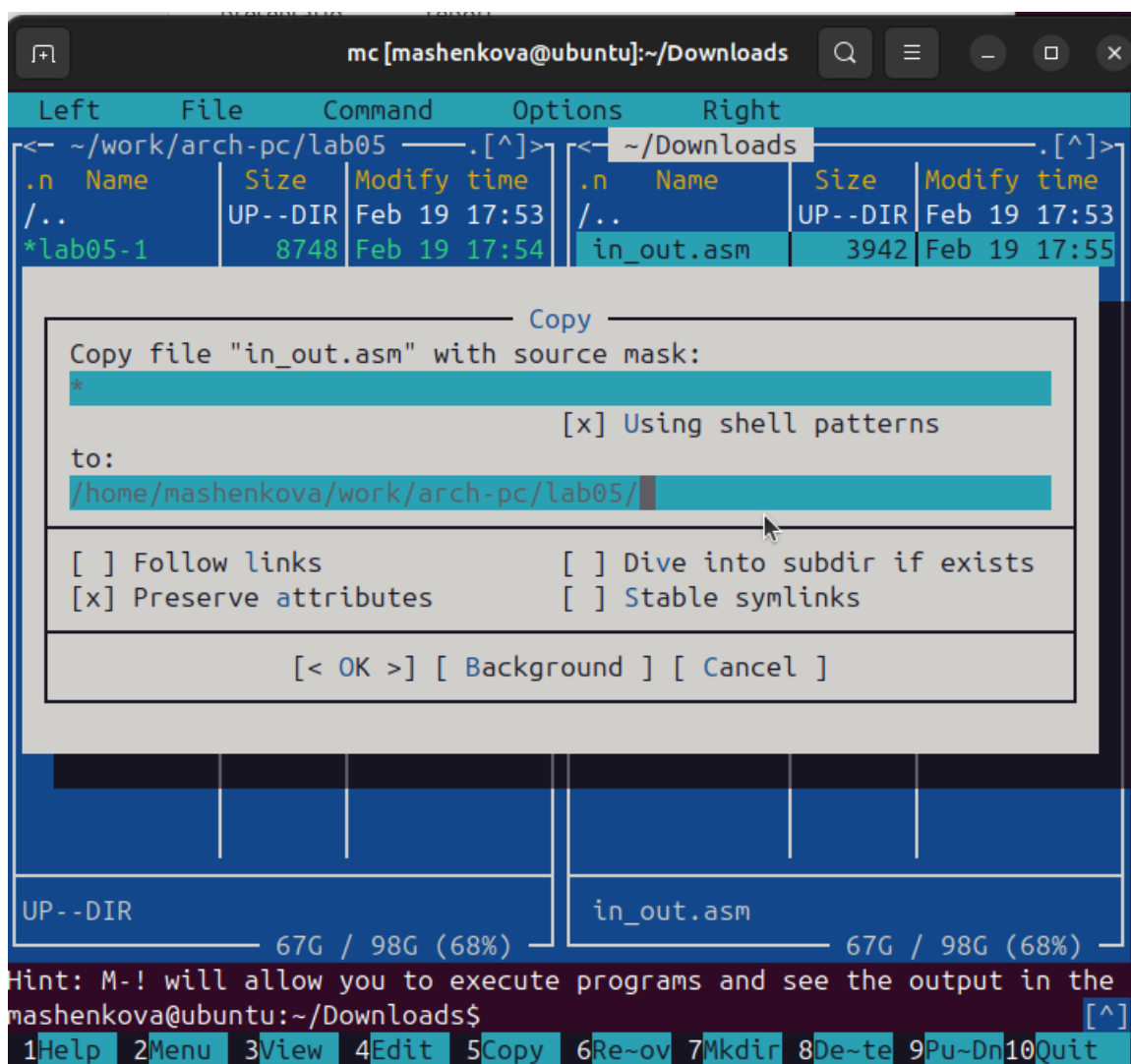


Рисунок 3.7: Копирование файла in_out.asm

Скопировала lab05-1.asm в lab05-2.asm. (рис. 3.8)

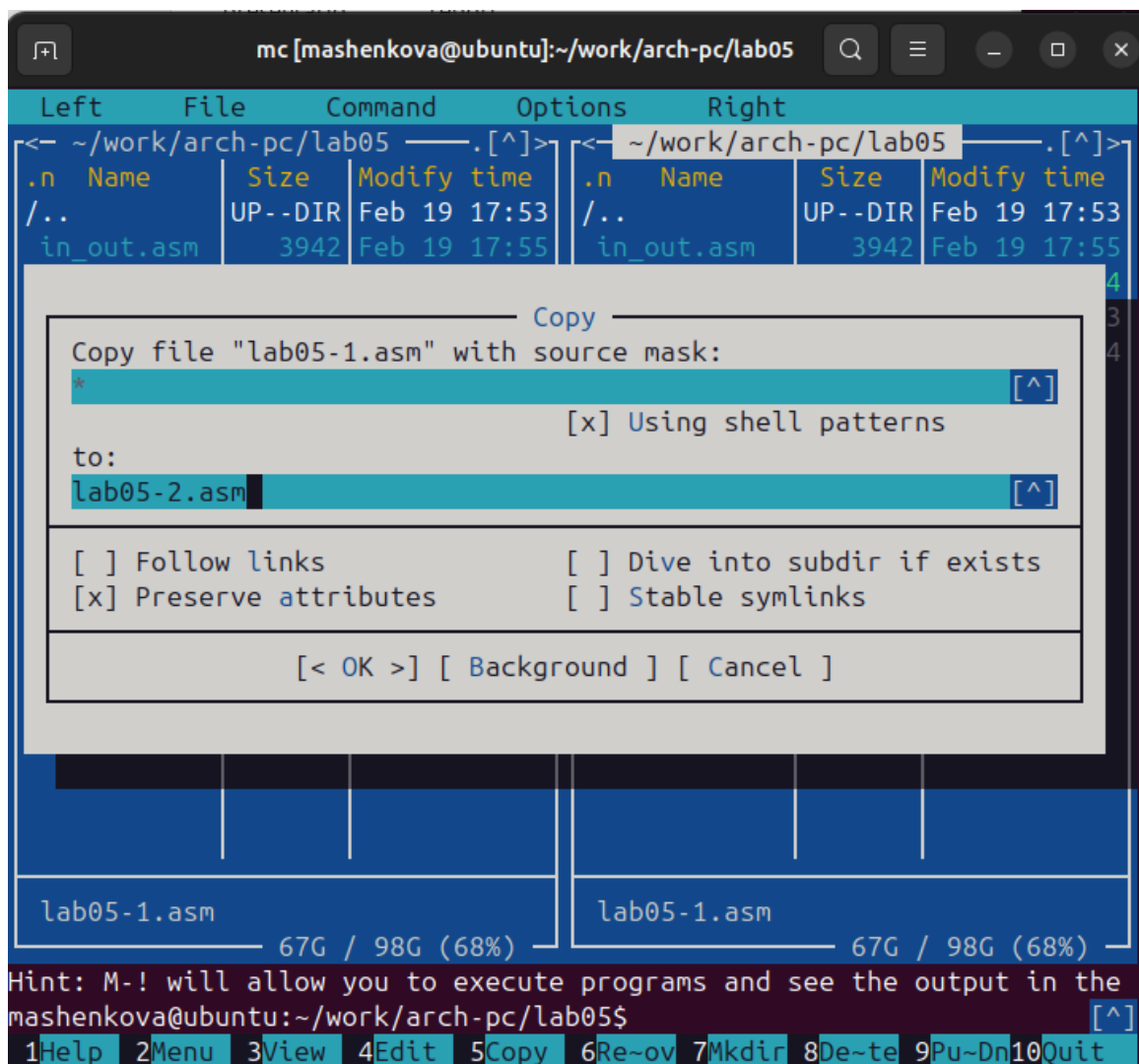
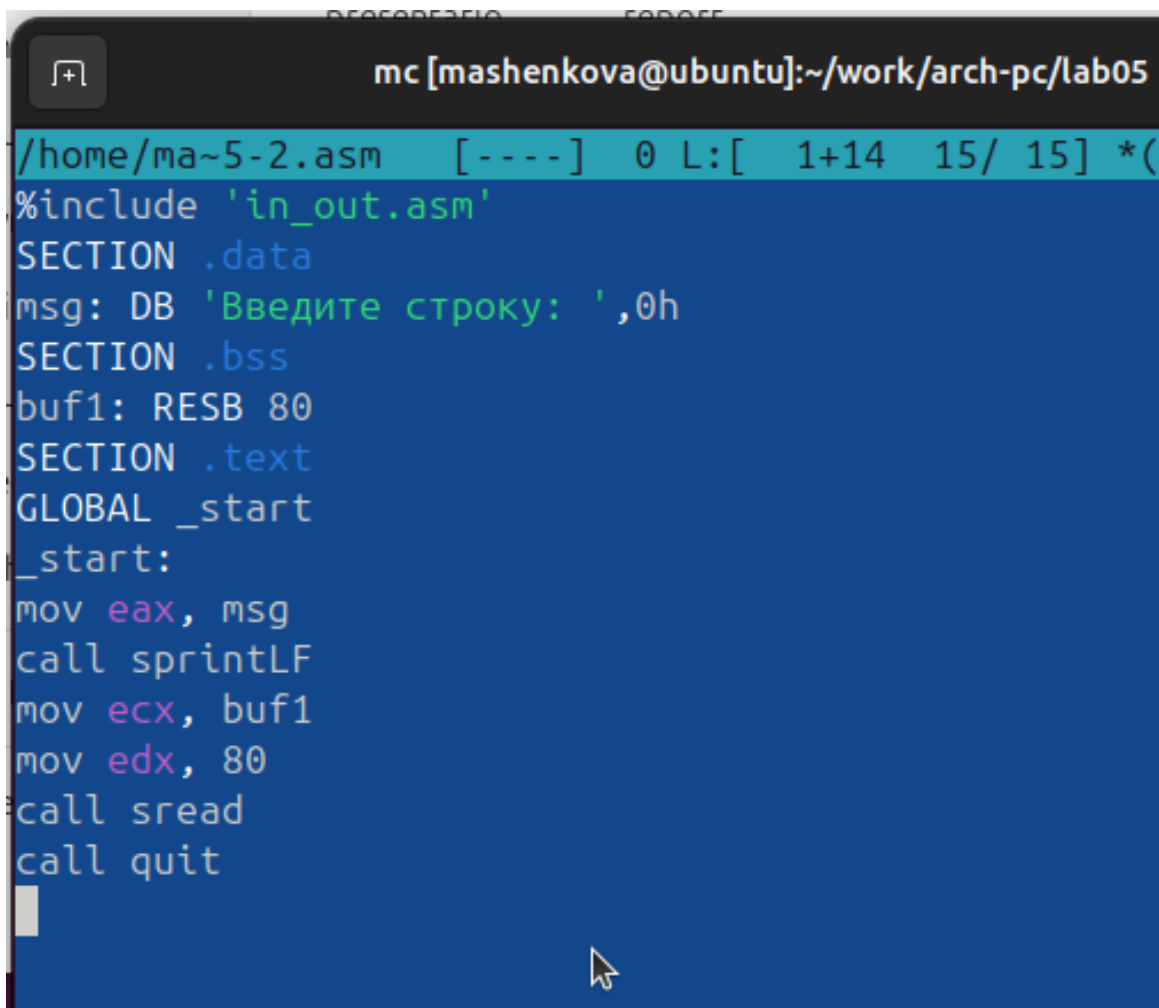


Рисунок 3.8: Копирование файла lab05-1.asm

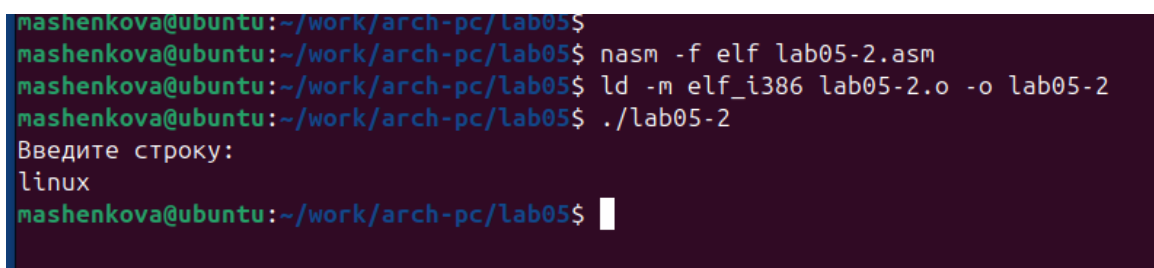
Пишу код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm. (рис. 3.9)



```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/ma~5-2.asm [----] 0 L:[ 1+14 15/ 15] *(
#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 3.9: Программа lab05-2.asm

Скомпилирую программу и проверю запуск. (рис. 3.10)

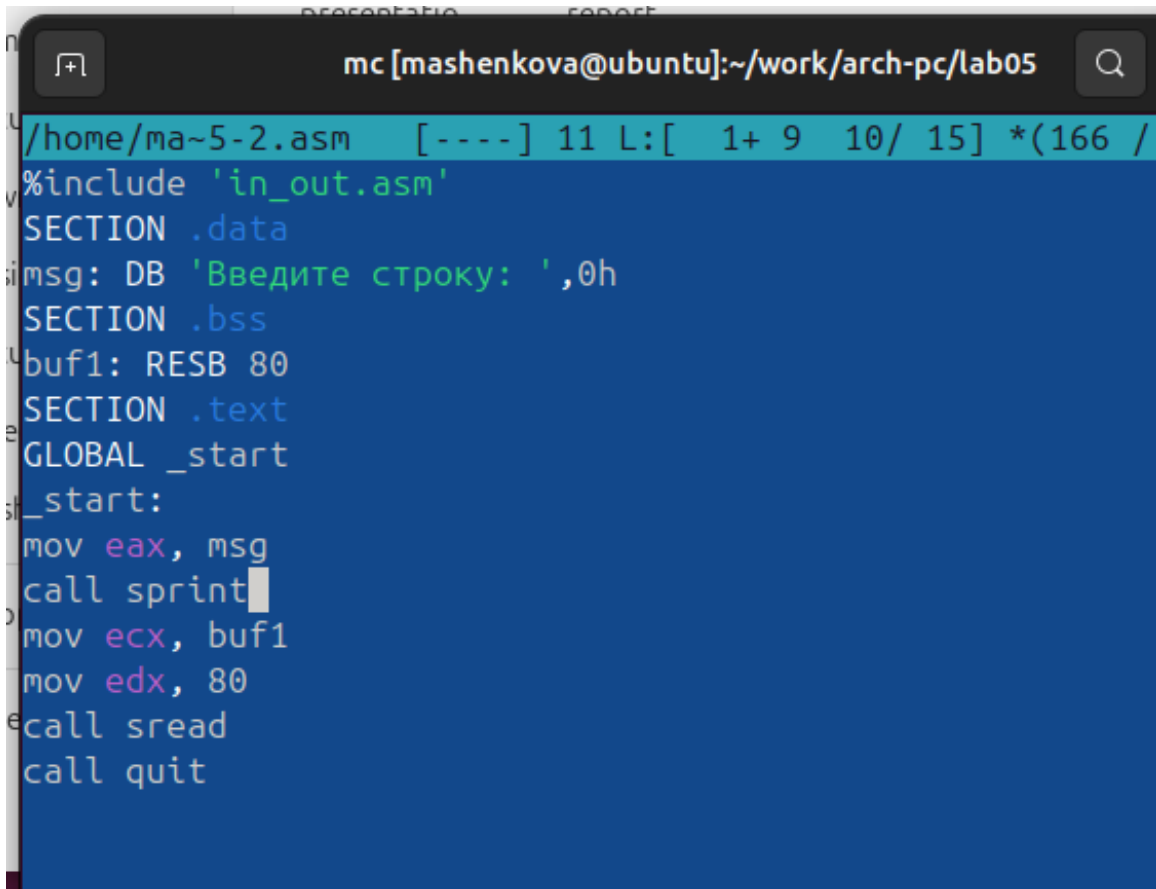


```
mashenkova@ubuntu:~/work/arch-pc/lab05$
mashenkova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
mashenkova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
mashenkova@ubuntu:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
linux
mashenkova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 3.10: Запуск программы lab05-2.asm

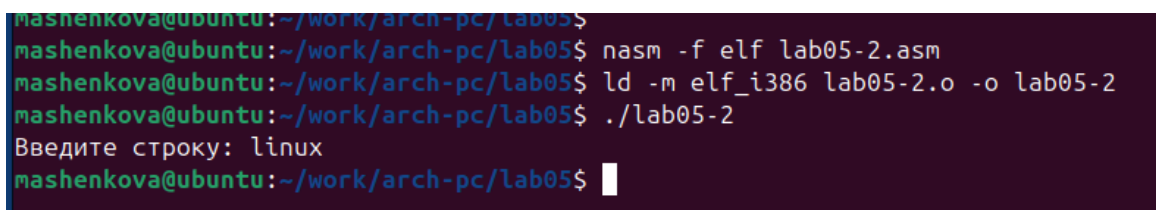
В файле lab5-2.asm заменила подпрограмму sprintLF на sprint. Заново со-

брала исполняемый файл. (рис. 3.11) (рис. 3.12)



```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/ma~5-2.asm [ - - - ] 11 L: [ 1+ 9 10/ 15] *(166 /
%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 3.11: Программа в файле lab05-2.asm



```
mashenkova@ubuntu:~/work/arch-pc/lab05$
mashenkova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
mashenkova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
mashenkova@ubuntu:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: linux
mashenkova@ubuntu:~/work/arch-pc/lab05$
```

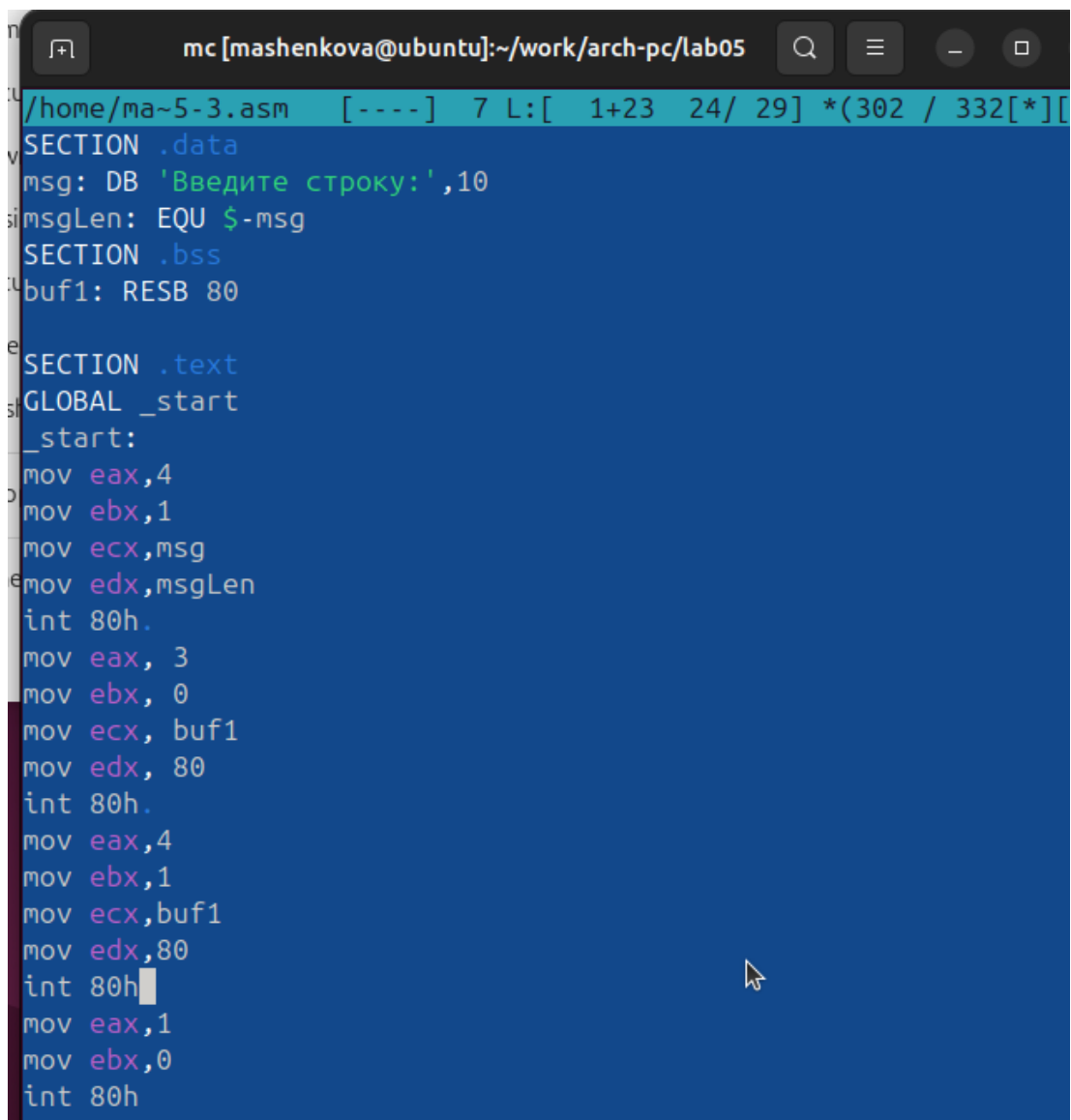
Рисунок 3.12: Запуск программы lab05-2.asm

Теперь после вывода строки она не завершается символом перехода на новую строку.

3.3 Задание для самостоятельной работы

Скопировала программу lab05-1.asm и изменила код, так чтобы она работала по следующему алгоритму: (рис. 3.13) (рис. 3.14)

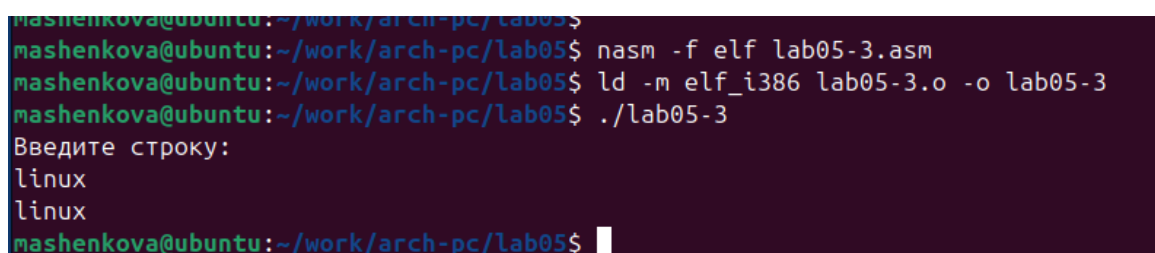
- вывести приглашение типа «Введите строку:»;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.



```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/ma~5-3.asm [----] 7 L:[ 1+23 24/ 29] *(302 / 332[*])[
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
```

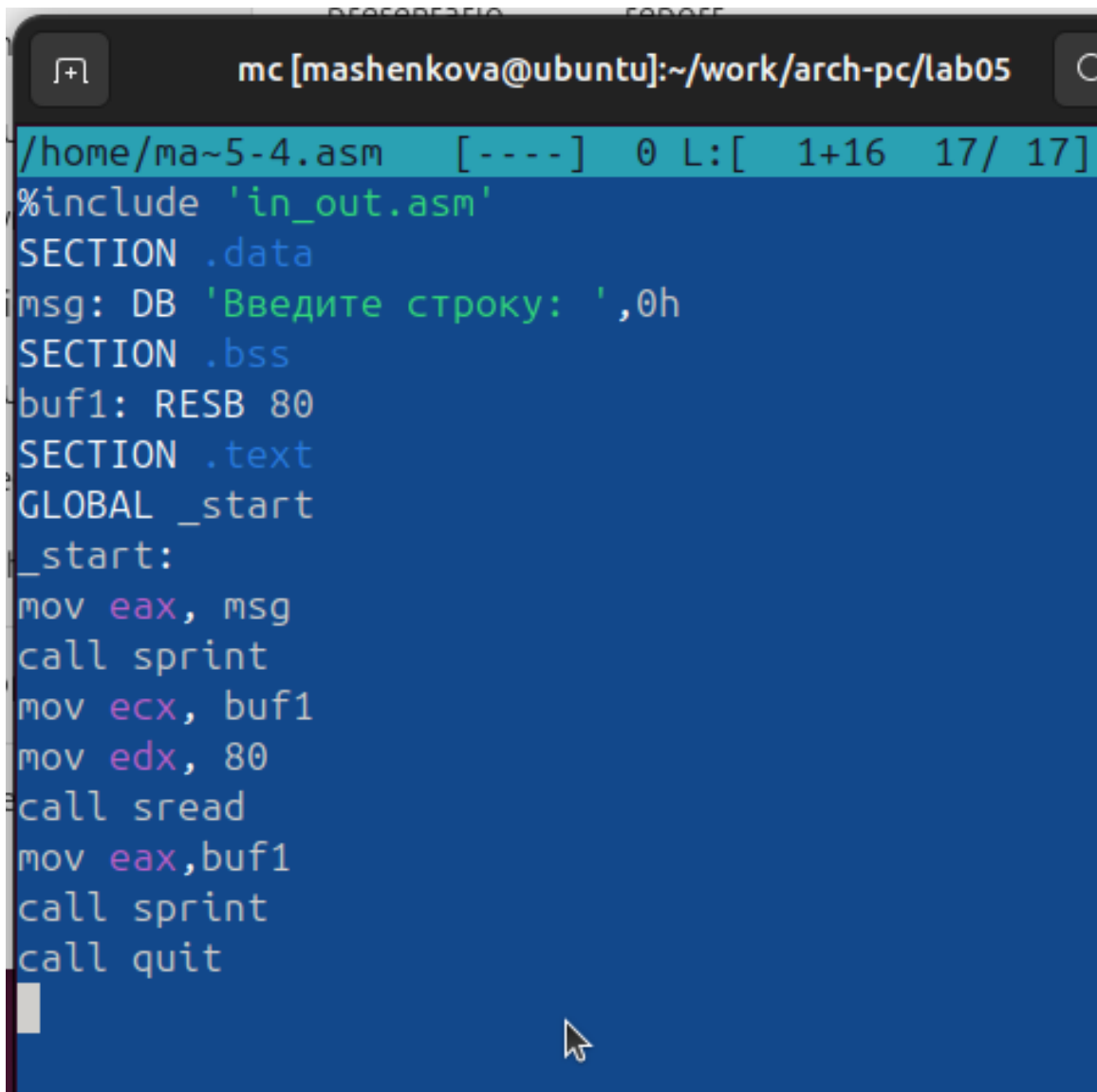
Рисунок 3.13: Программа lab05-3.asm



```
mashenkova@ubuntu:~/work/arch-pc/lab05$
mashenkova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
mashenkova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
mashenkova@ubuntu:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
linux
linux
mashenkova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 3.14: Запуск программы lab05-3.asm

Аналогично скопировала программу lab05-2.asm и изменила код, но теперь использовал подпрограммы из файла in_out.asm. (рис. 3.15) (рис. 3.16)



```
mc [mashenkova@ubuntu]:~/work/arch-pc/lab05
/home/ma~5-4.asm  [ - - - - ]  0  L:[ 1+16 17/ 17]
%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 3.15: Программа lab05-4.asm

```
mashenkova@ubuntu:~/work/arch-pc/lab05$  
mashenkova@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm  
mashenkova@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4  
mashenkova@ubuntu:~/work/arch-pc/lab05$ ./lab05-4  
Введите строку: linux  
linux  
mashenkova@ubuntu:~/work/arch-pc/lab05$
```

Рисунок 3.16: Запуск программы lab05-4.asm

4 Выводы

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

Список литературы

1. Архитектура ЭВМ
2. NASM Assembly Language Tutorials. — URL: <https://asmtutor.com/>.