

Отчёт по лабораторной работе 5

Архитектура компьютера

Довлетов Довлет

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	14

Список иллюстраций

2.1	Создание каталога	6
2.2	Программа lab05-1.asm	7
2.3	Просмотр файла lab05-1.asm	8
2.4	Запуск программы lab05-1.asm	9
2.5	Копирование файла	9
2.6	Программа lab05-2.asm	10
2.7	Запуск программы lab05-2.asm	10
2.8	Программа lab05-2.asm	11
2.9	Запуск программы lab05-2.asm	11
2.10	Программа lab05-3.asm	12
2.11	Запуск программы lab05-3.asm	12
2.12	Программа lab05-4.asm	13
2.13	Запуск программы lab05-4.asm	13

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

1. Открыл Midnight Commander и перешел в каталог ~/work/arch-pc. Создал новый каталог lab05 для работы.

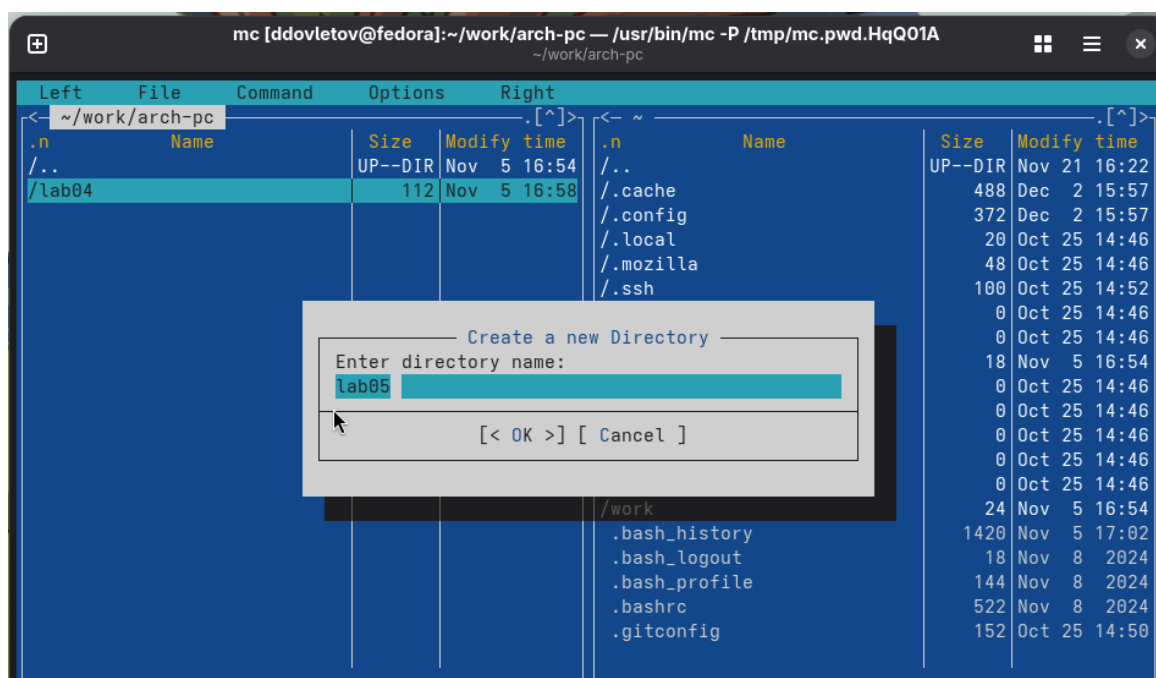


Рисунок 2.1: Создание каталога

2. Создал файл lab05-1.asm, открыл его для редактирования и написал начальный код программы.

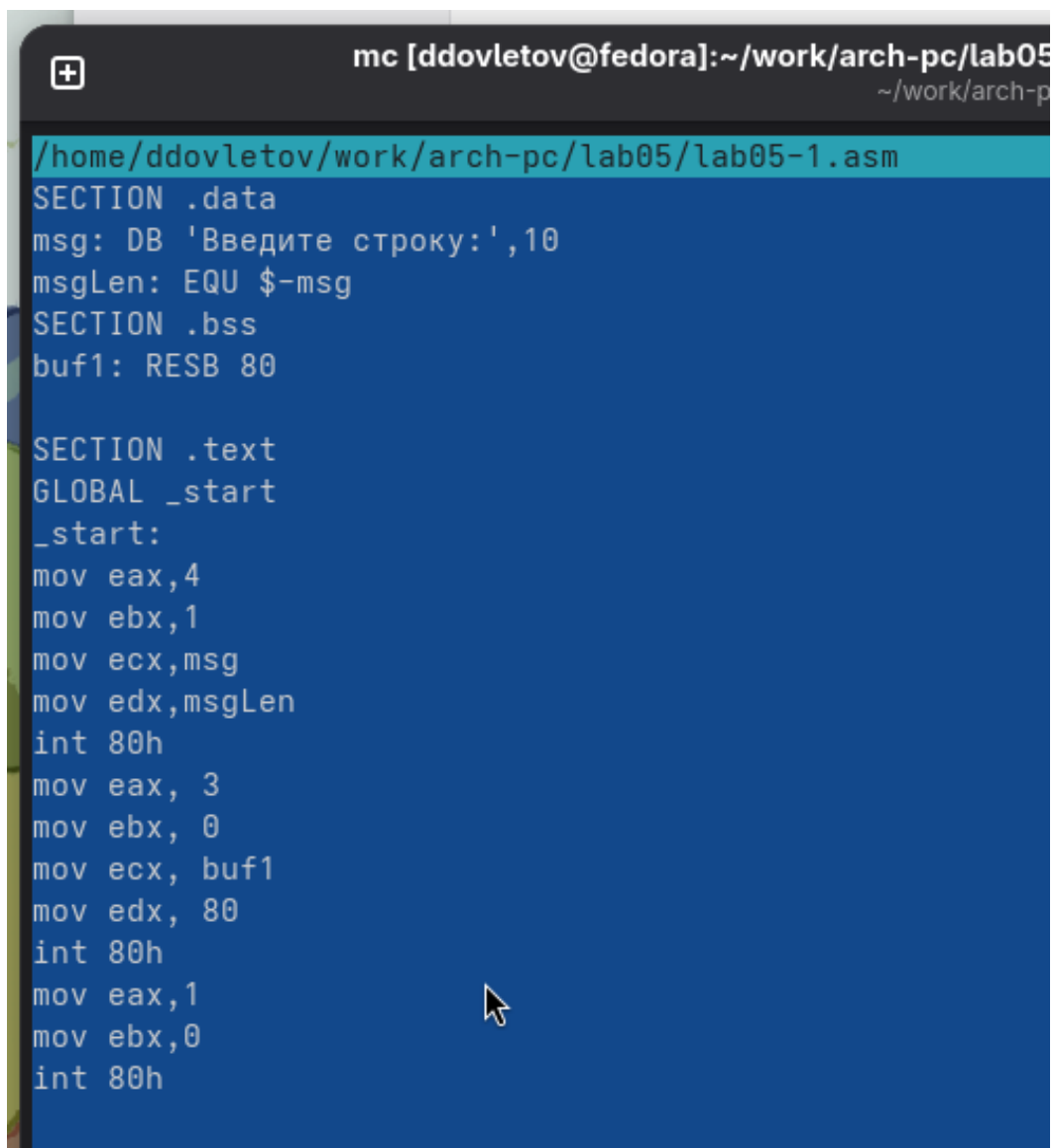


```
mc [ddovletov@fedora]:~/work
lab05-1.asm [----] 7 L:[ 1+18 19
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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 2.2: Программа lab05-1.asm

3. Открыл файл для просмотра и проверил корректность написанного кода.



```
mc [ddovletov@fedora]:~/work/arch-pc/lab05
~/work/arch-p
/home/ddovletov/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 2.3: Просмотр файла lab05-1.asm

4. Скомпилировал файл и запустил полученный исполняемый файл, проверив его работоспособность.


```
ddovletov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
ddovletov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
ddovletov@fedora:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Dovlet
ddovletov@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.4: Запуск программы lab05-1.asm

5. Скачал файл in_out.asm, добавил его в рабочий каталог. Скопировал файл lab05-1.asm и переименовал в lab05-2.asm.

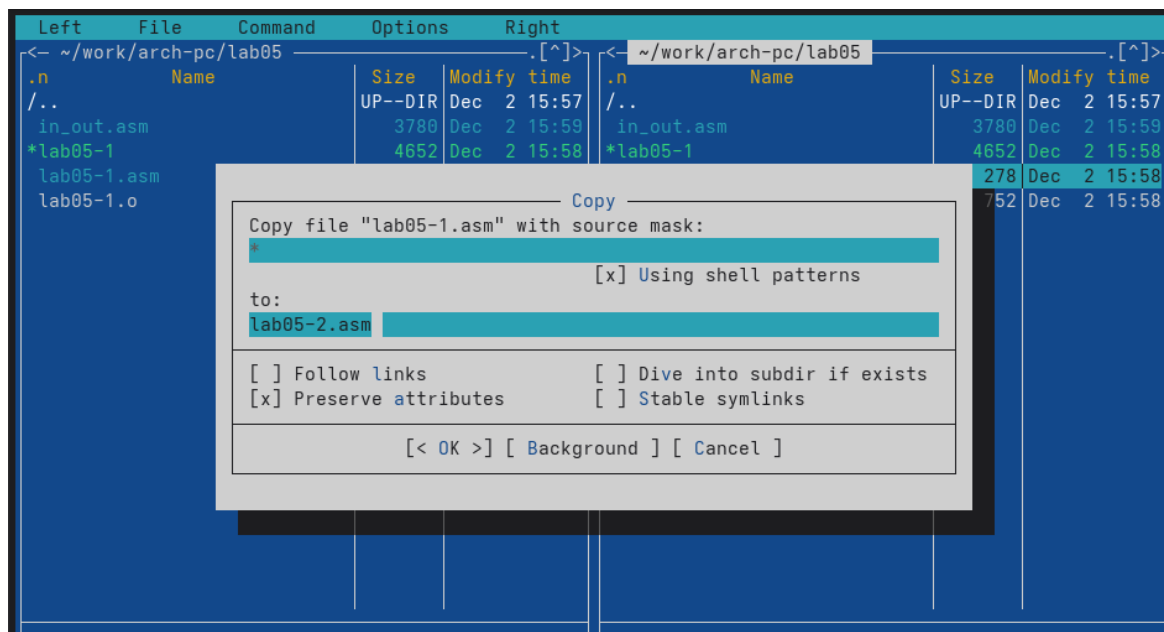
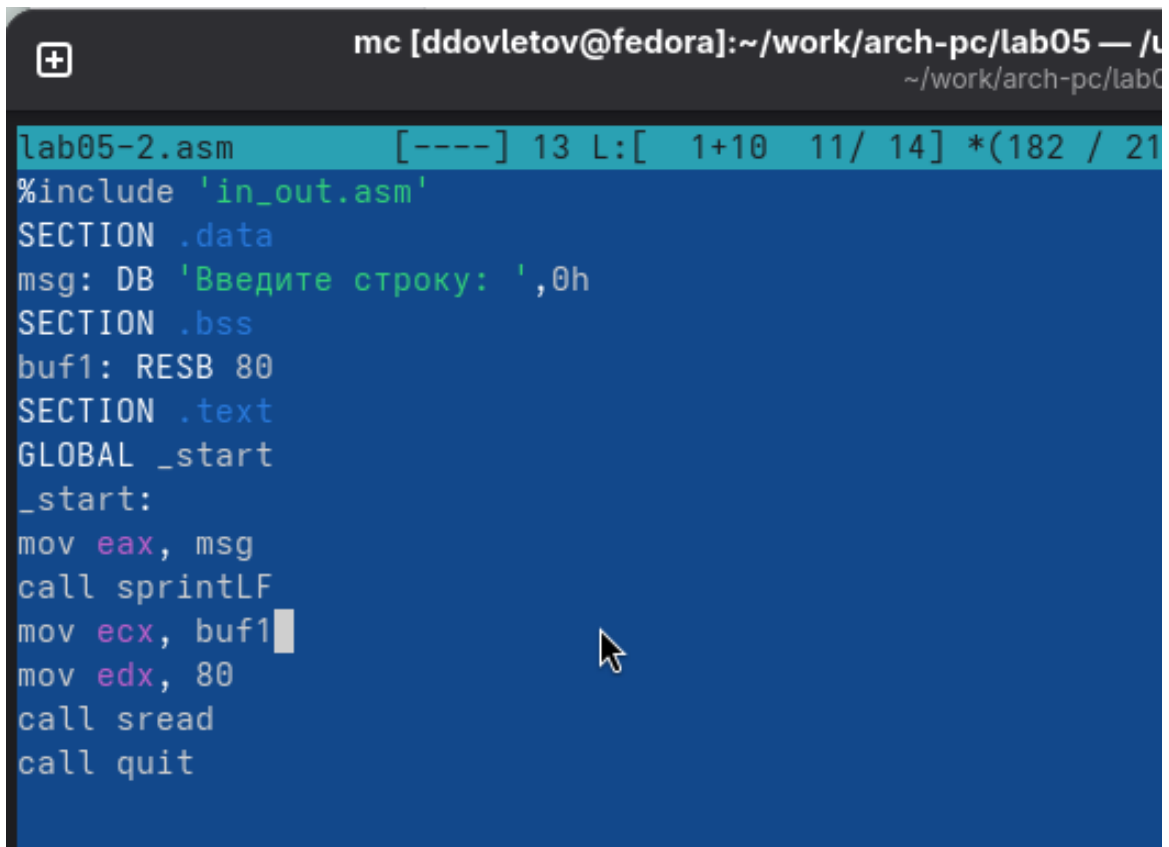


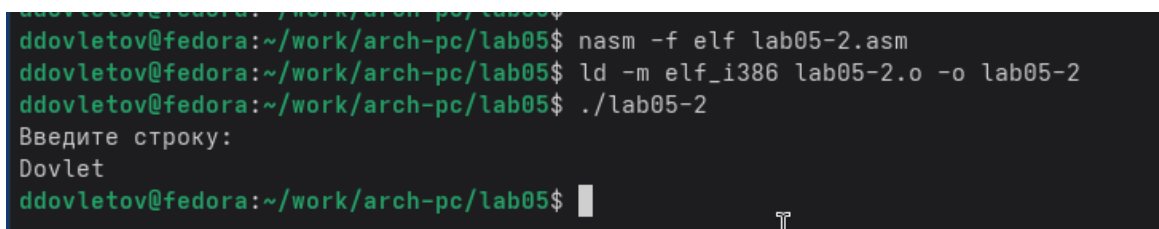
Рисунок 2.5: Копирование файла

6. Написал код для программы lab05-2.asm, скомпилировал её и проверил корректность работы.



```
lab05-2.asm [----] 13 L:[ 1+10 11/ 14] *(182 / 21
#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
```

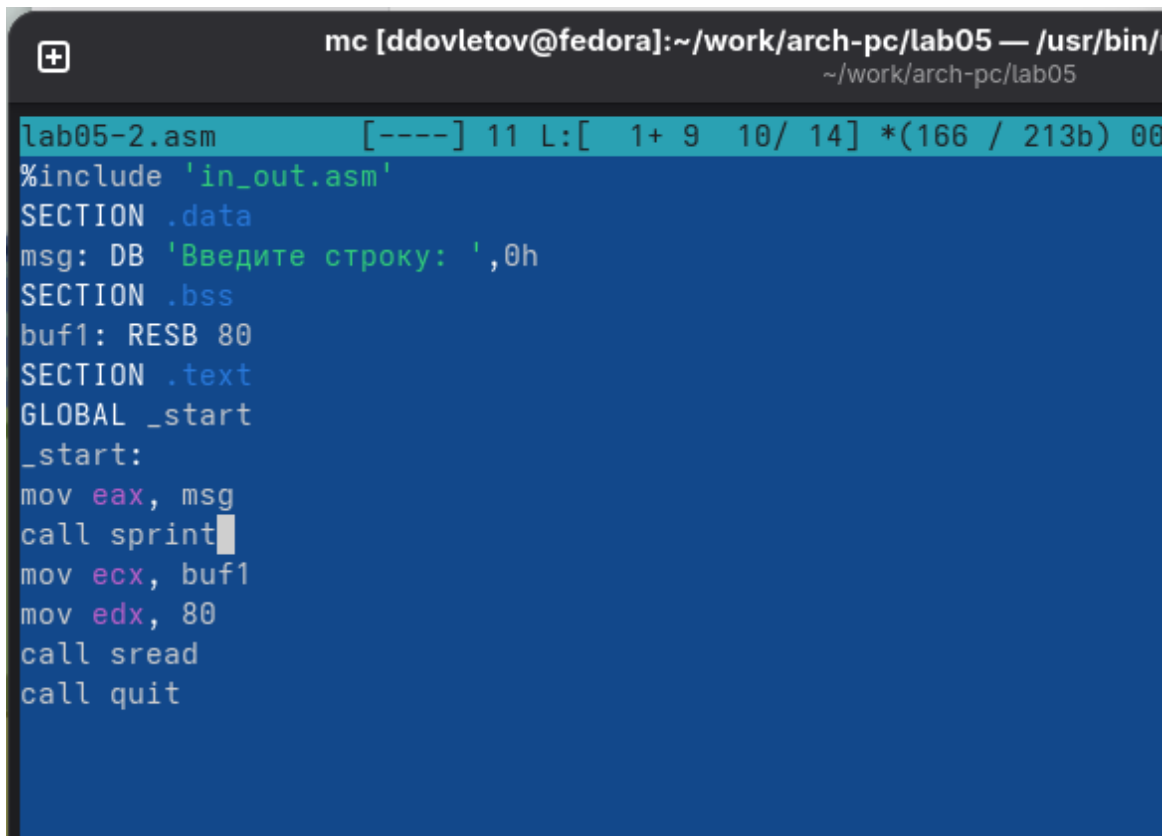
Рисунок 2.6: Программа lab05-2.asm



```
ddovletov@fedora: ~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
ddovletov@fedora: ~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
ddovletov@fedora: ~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Dovlet
ddovletov@fedora: ~/work/arch-pc/lab05$
```

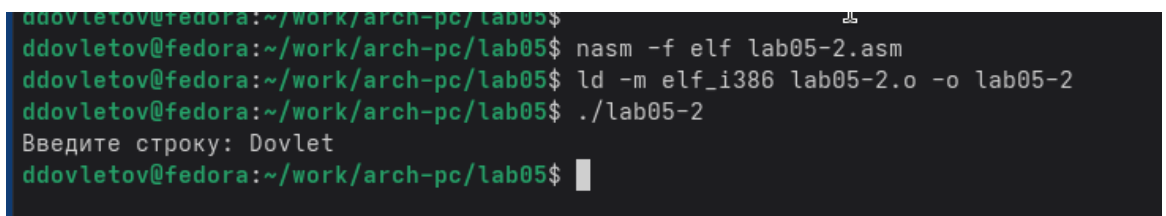
Рисунок 2.7: Запуск программы lab05-2.asm

7. В файле lab05-2.asm заменил подпрограмму sprintLF на sprint. Пересобрал исполняемый файл. Теперь строка выводится без перехода на новую строку.



```
mc [ddovletov@fedora]:~/work/arch-pc/lab05 — /usr/bin/
~/work/arch-pc/lab05
lab05-2.asm [----] 11 L:[ 1+ 9 10/ 14] *(166 / 213b) 00
#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 2.8: Программа lab05-2.asm



```
ddovletov@fedora:~/work/arch-pc/lab05$
ddovletov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
ddovletov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
ddovletov@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: Dovlet
ddovletov@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.9: Запуск программы lab05-2.asm

8. Скопировал программу lab05-1.asm и изменил код таким образом, чтобы:

- Вывести приглашение типа «Введите строку:».
- Прочитать введенную строку с клавиатуры.
- Вывести введенную строку на экран.

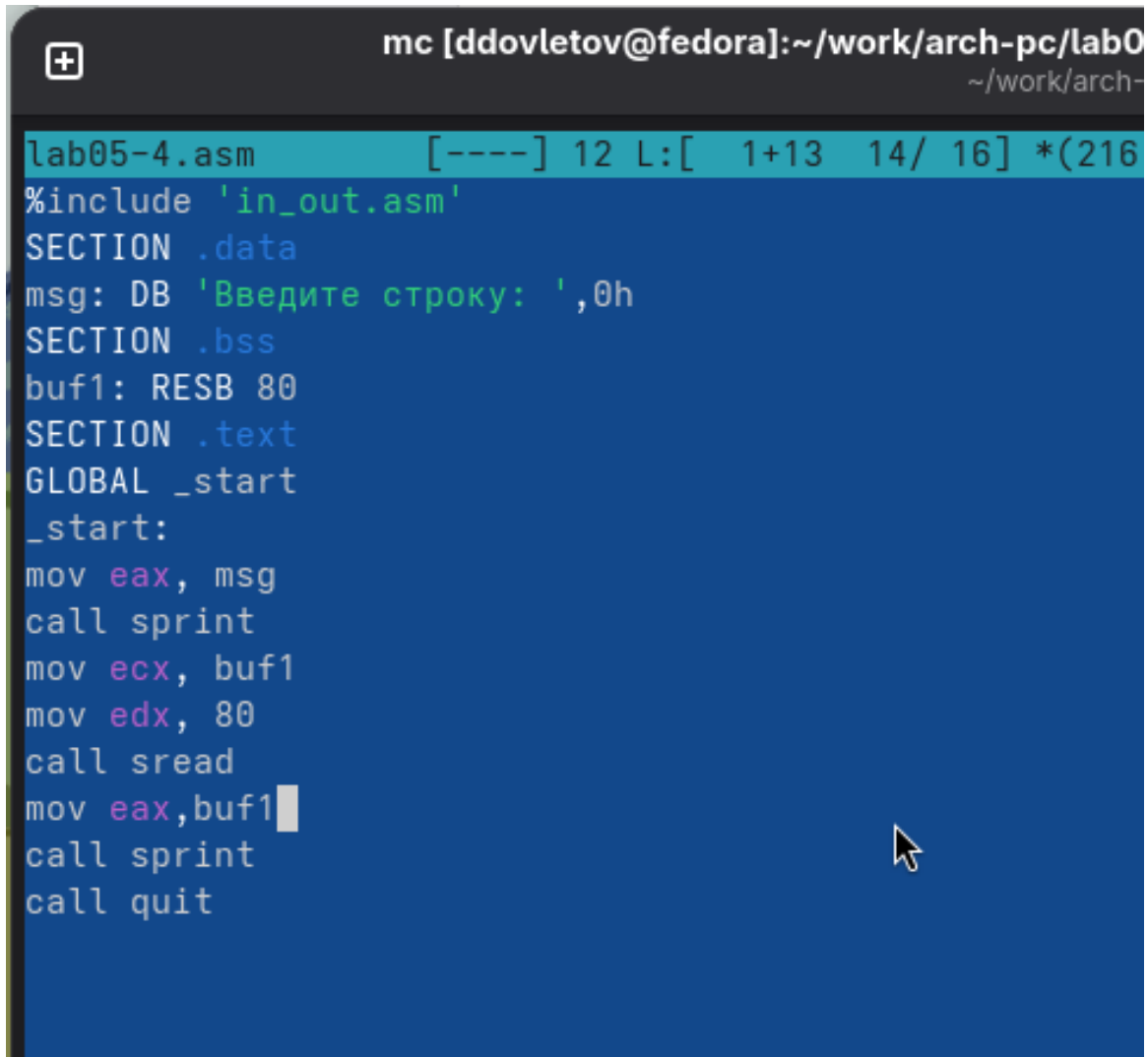
```
mc [ddovletov@fedora]:~/work/arch-pc/lab05$  
lab05-3.asm [----] 0 L: [ 1+27 28/ 28] *(331  
SECTION .data  
msg: DB 'Введите строку:',10  
msgLen: EQU $-msg  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,4  
mov ebx,1  
mov ecx,msg  
mov edx,msgLen  
int 80h  
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 2.10: Программа lab05-3.asm

```
ddovletov@fedora:~/work/arch-pc/lab05$  
ddovletov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm  
ddovletov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3  
ddovletov@fedora:~/work/arch-pc/lab05$ ./lab05-3  
Введите строку:  
Dovlet  
Dovlet  
ddovletov@fedora:~/work/arch-pc/lab05$
```

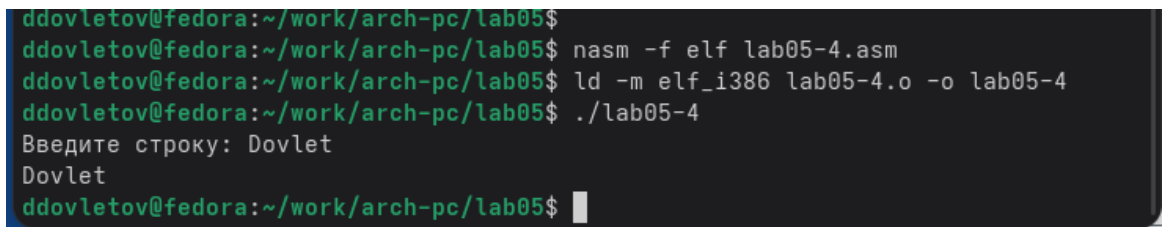
Рисунок 2.11: Запуск программы lab05-3.asm

9. Скопировал программу lab05-2.asm и адаптировал её по аналогии с предыдущим заданием, используя возможности из файла in_out.asm.



```
mc [ddovletov@fedora]:~/work/arch-pc/lab05
~/work/arch-
lab05-4.asm [----] 12 L:[ 1+13 14/ 16] *(216
%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 2.12: Программа lab05-4.asm



```
ddovletov@fedora:~/work/arch-pc/lab05$
ddovletov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
ddovletov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
ddovletov@fedora:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: Dovlet
Dovlet
ddovletov@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.13: Запуск программы lab05-4.asm

3 Выводы

В ходе выполнения лабораторной работы были приобретены навыки написания базовых ассемблерных программ. Также освоены основные ассемблерные инструкции, такие как `mov` и `int`.