

Отчёт по лабораторной работе 5

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

Бозорова Ануша

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Знакомство с Midnight Commander	6
2.2	Подключение внешнего файла in_out.asm	10
2.3	Задание для самостоятельной работы	14
3	Выводы	18

Список иллюстраций

2.1	Запуск Midnight Commander	6
2.2	Создание каталога lab05	7
2.3	Создание файла lab05-1.asm	7
2.4	Редактирование программы в файле lab05-1.asm	8
2.5	Просмотр содержимого файла lab05-1.asm	9
2.6	Запуск программы lab05-1.asm	10
2.7	Копирование файла in_out.asm в рабочий каталог	10
2.8	Копирование файла lab05-1.asm в lab05-2.asm	11
2.9	Программа в файле lab05-2.asm	12
2.10	Запуск программы lab05-2.asm	12
2.11	Обновленная программа в lab05-2.asm	13
2.12	Запуск обновленной программы lab05-2.asm	13
2.13	Копирование файла lab05-1.asm для нового задания	14
2.14	Код программы в файле lab05-3.asm	15
2.15	Запуск программы lab05-3.asm	15
2.16	Копирование файла lab05-2.asm для модификации	16
2.17	Код программы в файле lab05-4.asm	17
2.18	Запуск программы lab05-4.asm	17

Список таблиц

1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander, а также освоение инструкций языка ассемблера `mov` и `int`.

2 Выполнение лабораторной работы

2.1 Знакомство с Midnight Commander

Я запустила Midnight Commander и, используя стрелочные клавиши и клавишу Enter, перешла в каталог ~/work/arch-рс. Затем нажала F7 для создания нового каталога lab05.

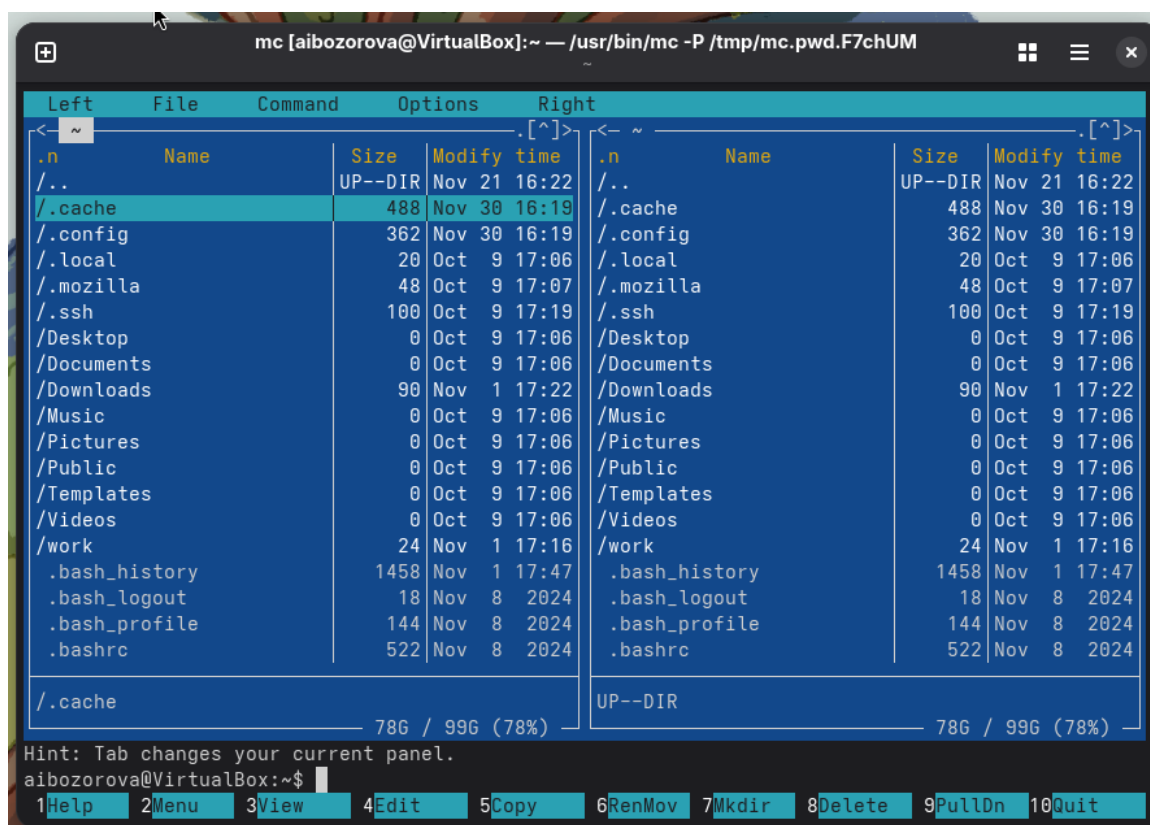


Рисунок 2.1: Запуск Midnight Commander

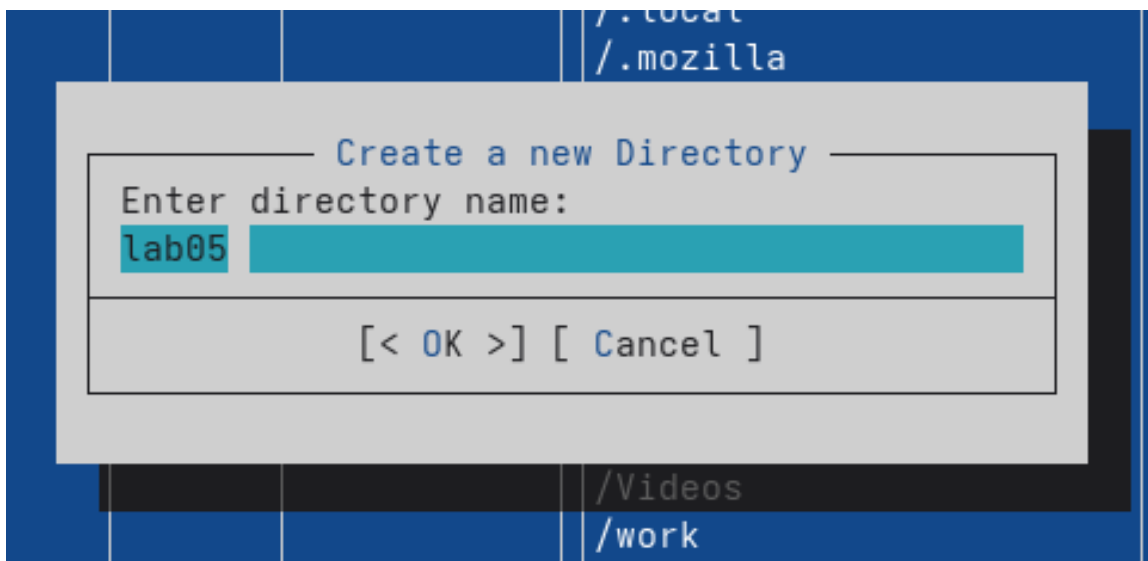


Рисунок 2.2: Создание каталога lab05

С помощью команды touch я создала файл lab05-1.asm.

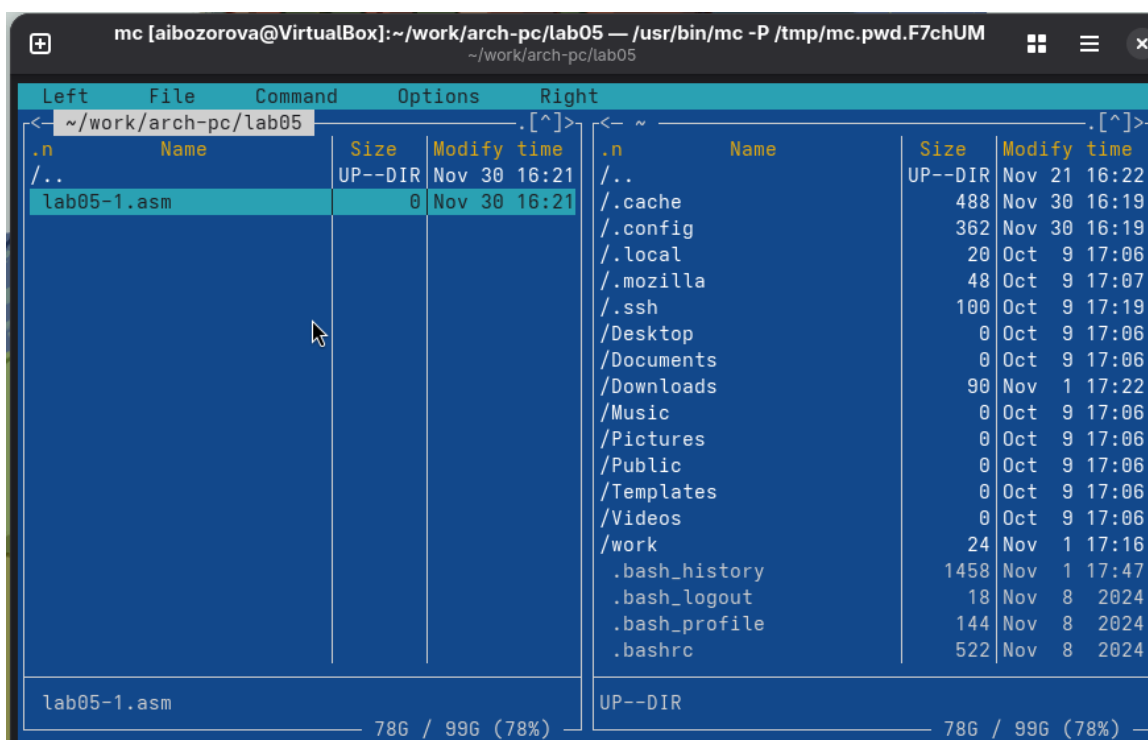


Рисунок 2.3: Создание файла lab05-1.asm

Я открыла файл для редактирования, нажав F4, выбрала редактор mceditor

и написала код программы в соответствии с заданием.

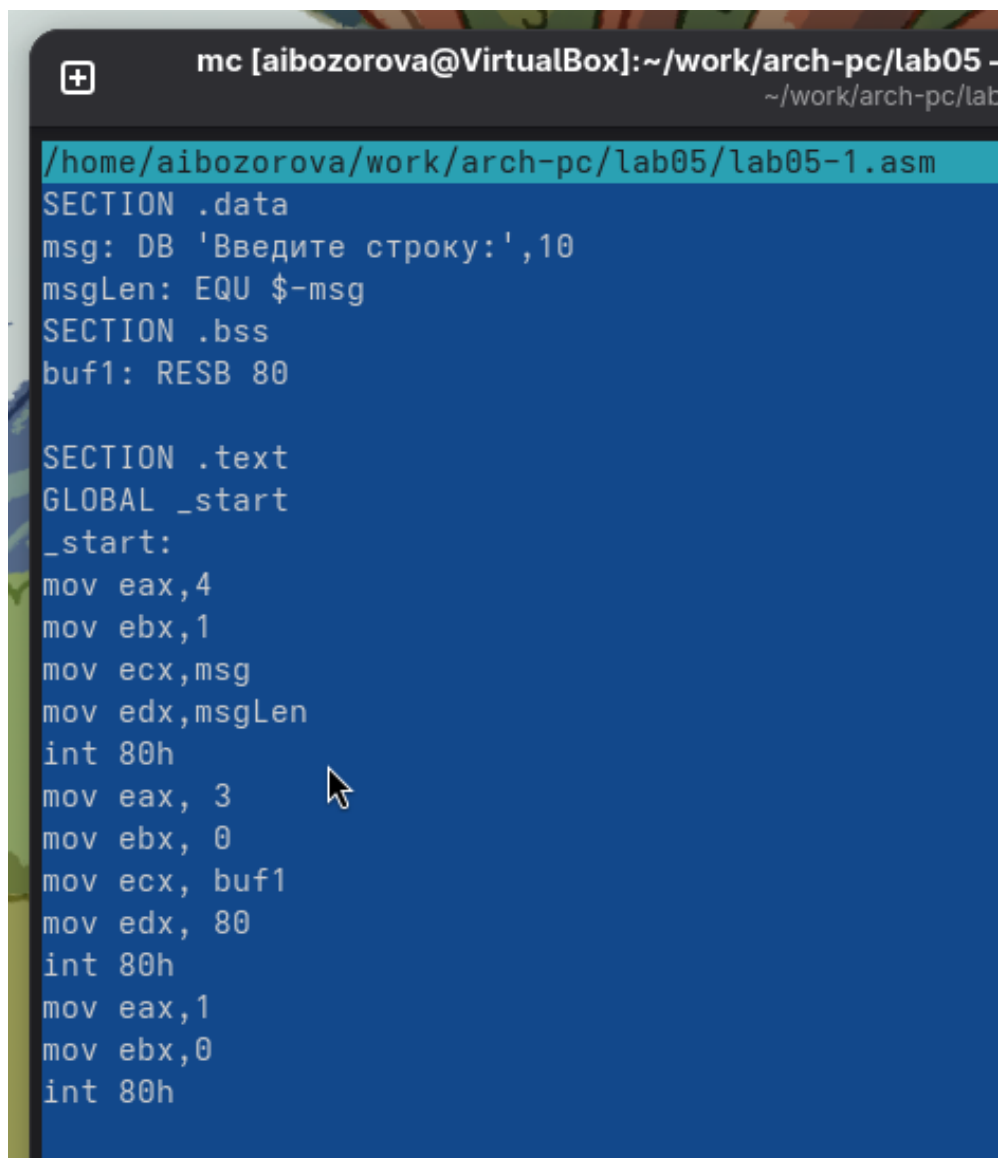


```
mc [aibozorova@VirtualBox]:~/work/arch
lab05-1.asm [----] 10 L:[ 1+14 15
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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: Редактирование программы в файле lab05-1.asm

Для проверки я открыла файл на просмотр, нажав F3, и убедилась, что он содержит правильный код.



```
mc [aibozorova@VirtualBox]:~/work/arch-pc/lab05 -  
~/work/arch-pc/lab  
  
/home/aibozorova/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 содержимого файла lab05-1.asm

Я скомпилировала программу, сгенерировала объектный файл, произвела компоновку и запустила исполняемый файл, чтобы убедиться в корректности работы.

```

aibozorova@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
hello
aibozorova@VirtualBox:~/work/arch-pc/lab05$

```

Рисунок 2.6: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

Я скачала файл in_out.asm и разместила его в рабочем каталоге. Для копирования я использовала клавишу F5, а для перемещения – клавишу F6.

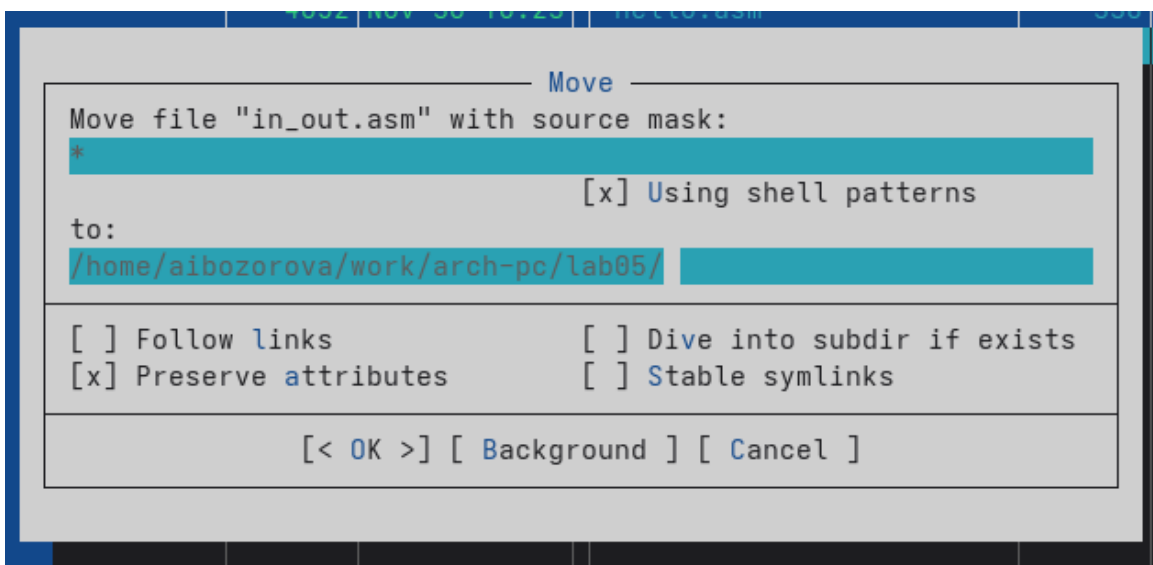


Рисунок 2.7: Копирование файла in_out.asm в рабочий каталог

Я также скопировала файл lab05-1.asm в новый файл lab05-2.asm.

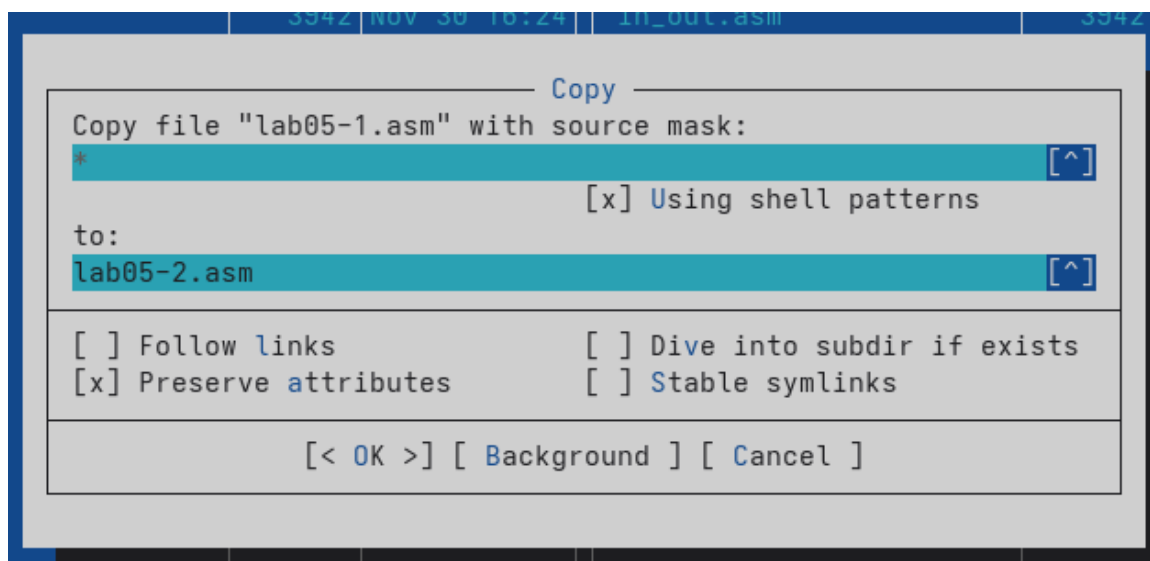
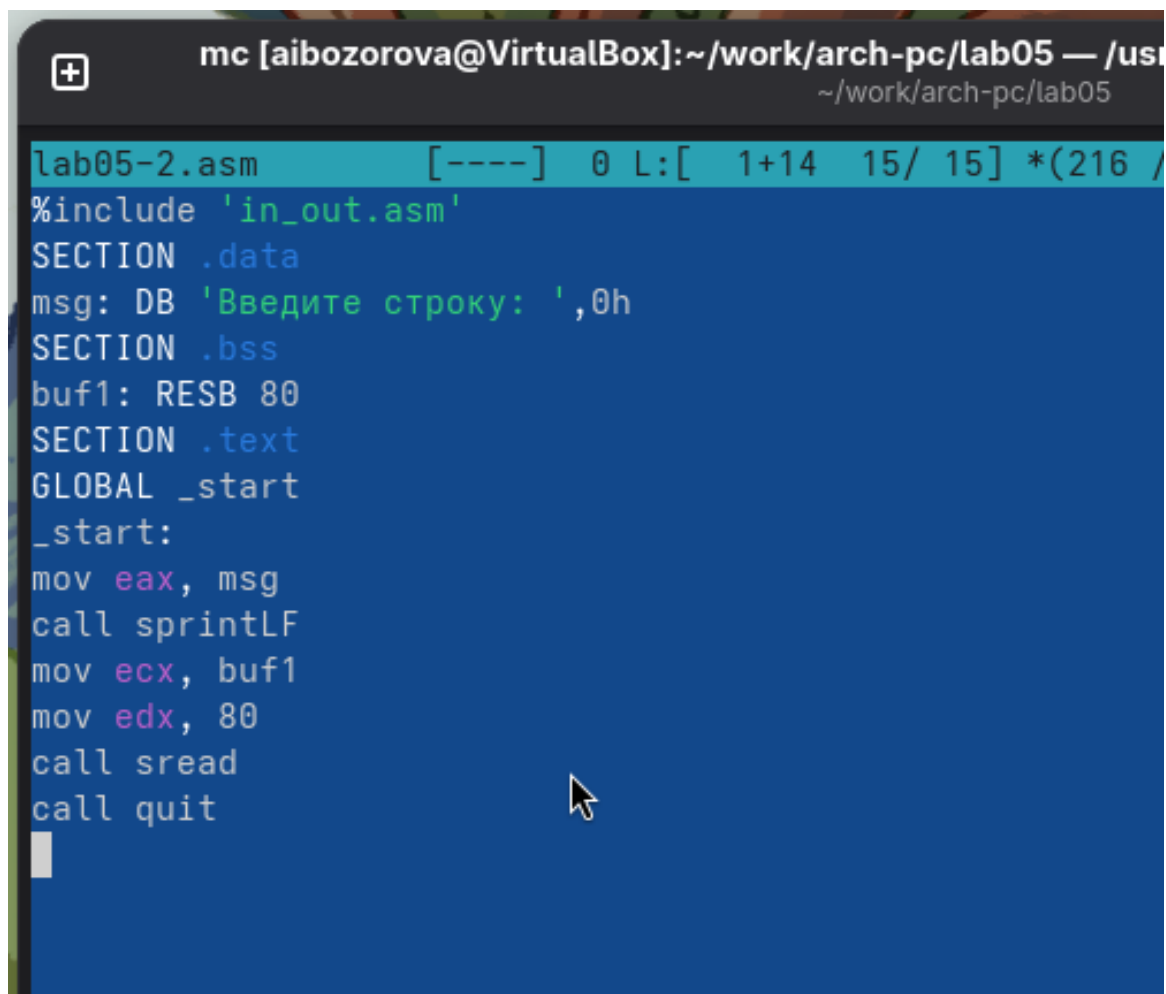


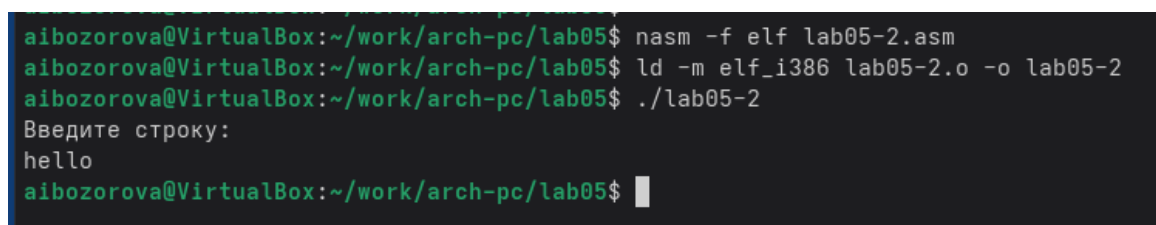
Рисунок 2.8: Копирование файла lab05-1.asm в lab05-2.asm

В файле lab05-2.asm я написала код программы, используя подпрограммы из внешнего файла in_out.asm. Программа была успешно скомпилирована и протестирована.



```
lab05-2.asm [----] 0 L:[ 1+14 15/ 15] *(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 printf
mov ecx, buf1
mov edx, 80
call gets
call _exit
```

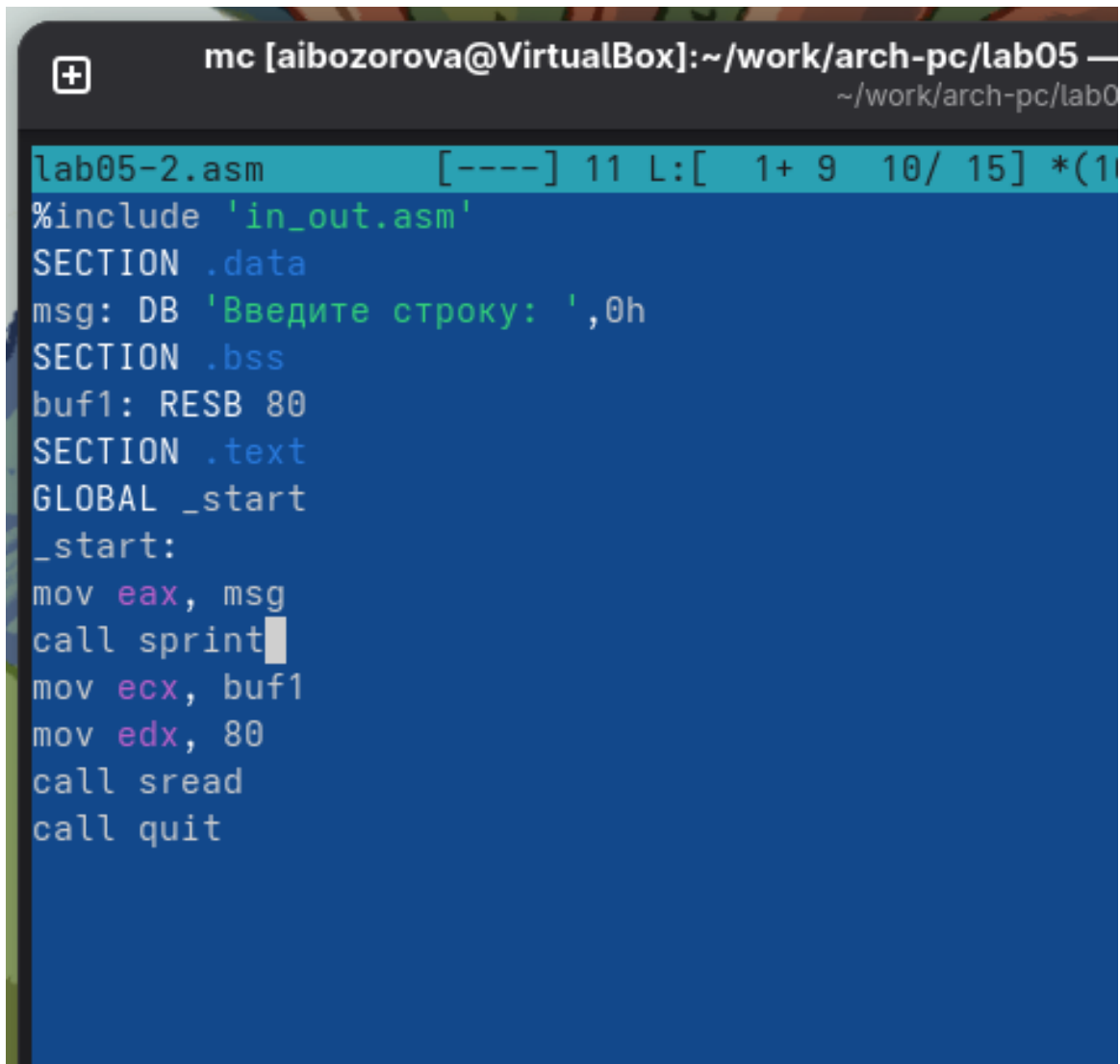
Рисунок 2.9: Программа в файле lab05-2.asm



```
aibozorova@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
hello
aibozorova@VirtualBox:~/work/arch-pc/lab05$
```

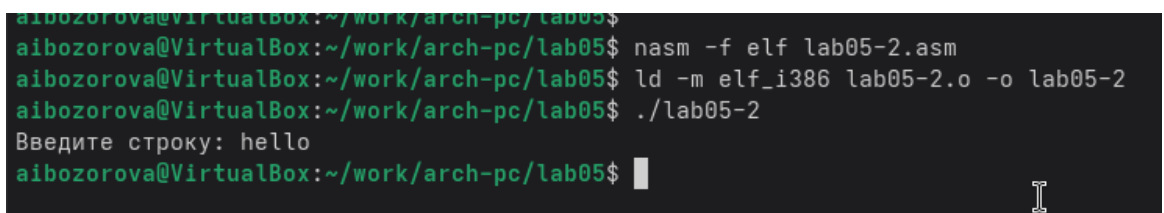
Рисунок 2.10: Запуск программы lab05-2.asm

В файле lab05-2.asm я заменила подпрограмму printf на sprintf. После пересборки исполняемого файла, теперь вывод строки не завершается символом новой строки.



```
mc [aibozorova@VirtualBox]:~/work/arch-pc/lab05 —  
~/work/arch-pc/lab05  
lab05-2.asm [----] 11 L:[ 1+ 9 10/ 15] *(1  
%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: Обновленная программа в lab05-2.asm



```
aibozorova@VirtualBox:~/work/arch-pc/lab05$  
aibozorova@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm  
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ./lab05-2  
Введите строку: hello  
aibozorova@VirtualBox:~/work/arch-pc/lab05$
```

Рисунок 2.12: Запуск обновленной программы lab05-2.asm

2.3 Задание для самостоятельной работы

Я скопировала программу lab05-1.asm и изменила код так, чтобы он работал по следующему алгоритму:

- вывести приглашение с текстом «Введите строку:»;
- считать строку с клавиатуры;
- вывести введённую строку на экран.

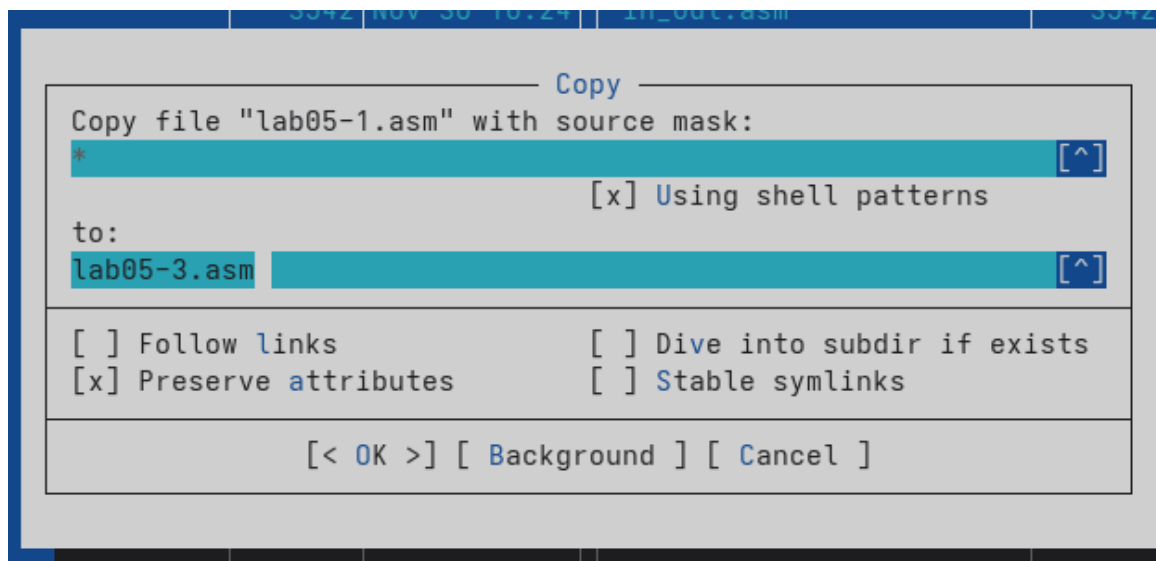


Рисунок 2.13: Копирование файла lab05-1.asm для нового задания

```
mc [aibozorova@VirtualBox]:~/work/arch-pc/lab05 — /u
~/work/arch-pc/lab05
lab05-3.asm [----] 0 L:[ 1+28 29/ 29] *(3
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 2.14: Код программы в файле lab05-3.asm

```
aibozorova@VirtualBox:~/work/arch-pc/lab05$
aibozorova@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
hello
hello
aibozorova@VirtualBox:~/work/arch-pc/lab05$
```

Рисунок 2.15: Запуск программы lab05-3.asm

Аналогично, я скопировала программу lab05-2.asm и внесла изменения в код, теперь используя подпрограммы из файла in_out.asm.

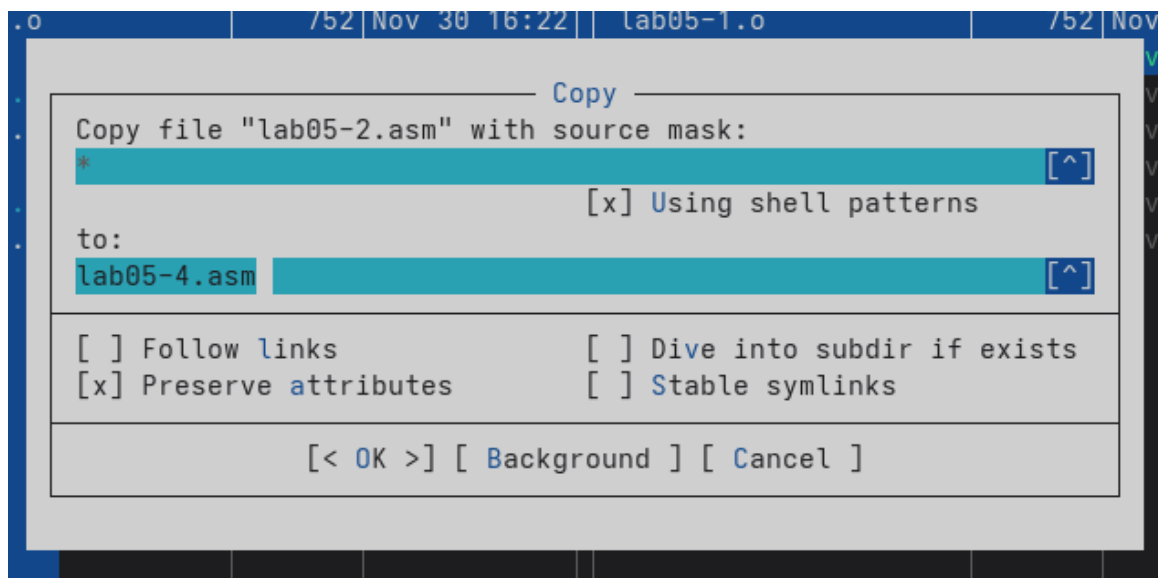
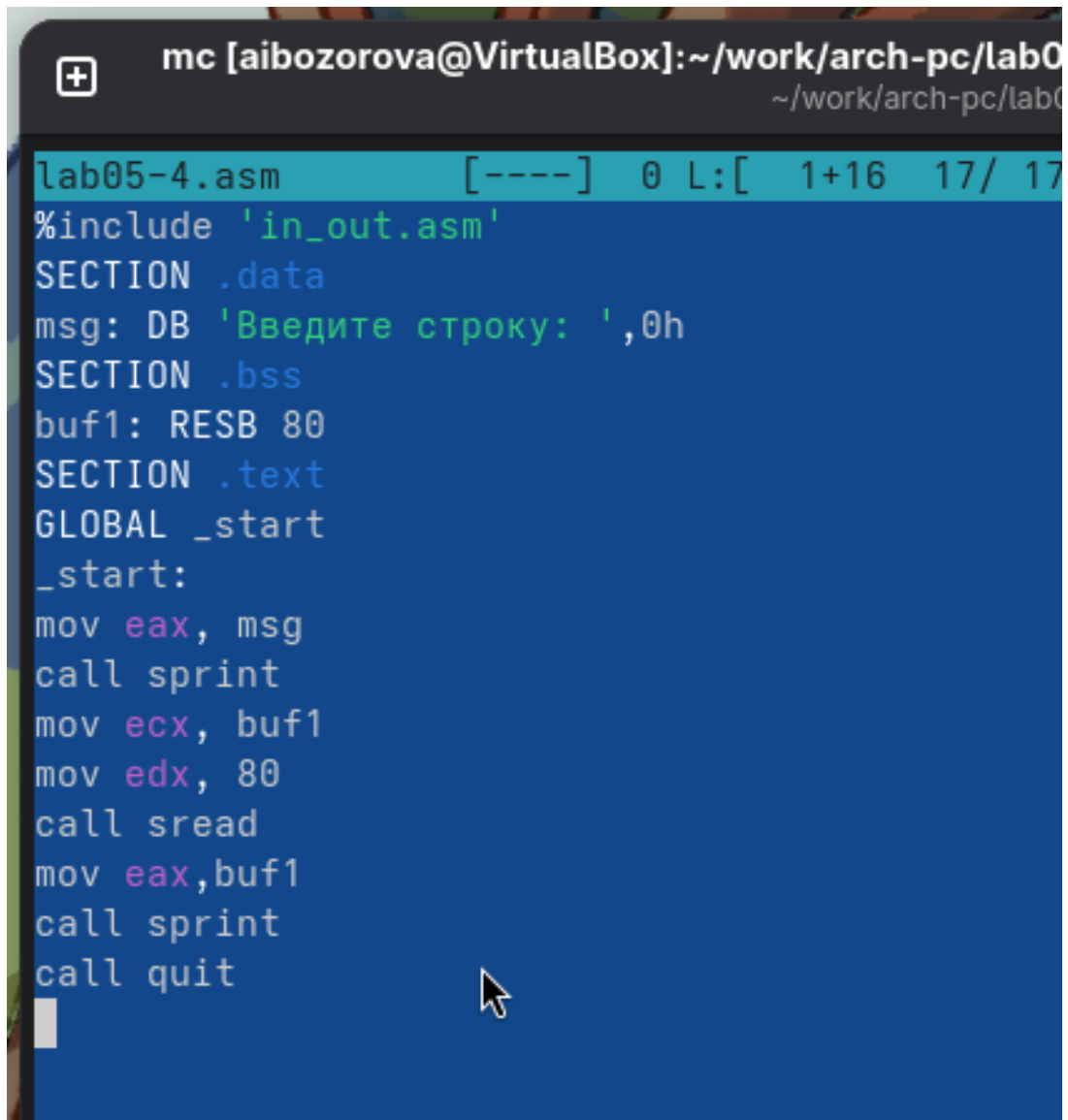
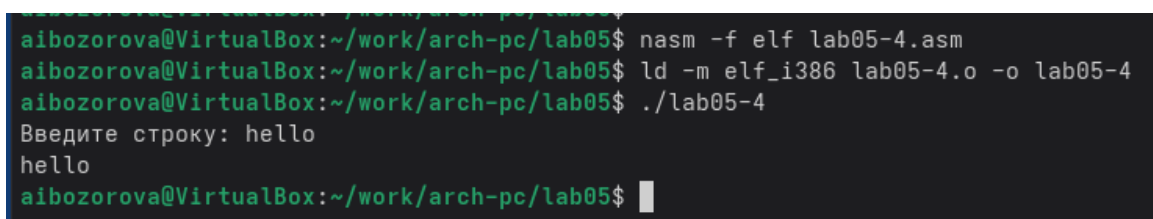


Рисунок 2.16: Копирование файла lab05-2.asm для модификации



```
mc [aibozorova@VirtualBox]:~/work/arch-pc/lab0
~/work/arch-pc/lab0
lab05-4.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 2.17: Код программы в файле lab05-4.asm



```
aibozorova@VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
aibozorova@VirtualBox:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: hello
hello
aibozorova@VirtualBox:~/work/arch-pc/lab05$
```

Рисунок 2.18: Запуск программы lab05-4.asm

3 Выводы

Я научилась писать базовые ассемблерные программы и освоила ассемблерные инструкции `mov` и `int`.