

Отчёт по лабораторной работе 5

Архитектура компьютеров

Богдан Гаряев НБИбд-01-2025

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	12
2.3	Задание для самостоятельной работы	16
3	Выводы	19

Список иллюстраций

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

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

Запускаю Midnight Commander (см. рис. 2.1), используя клавиши со стрелками и Enter, перехожу в каталог ~/work/arch-pc. Затем нажимаю F7 для создания нового каталога под названием lab05 (см. рис. 2.2).

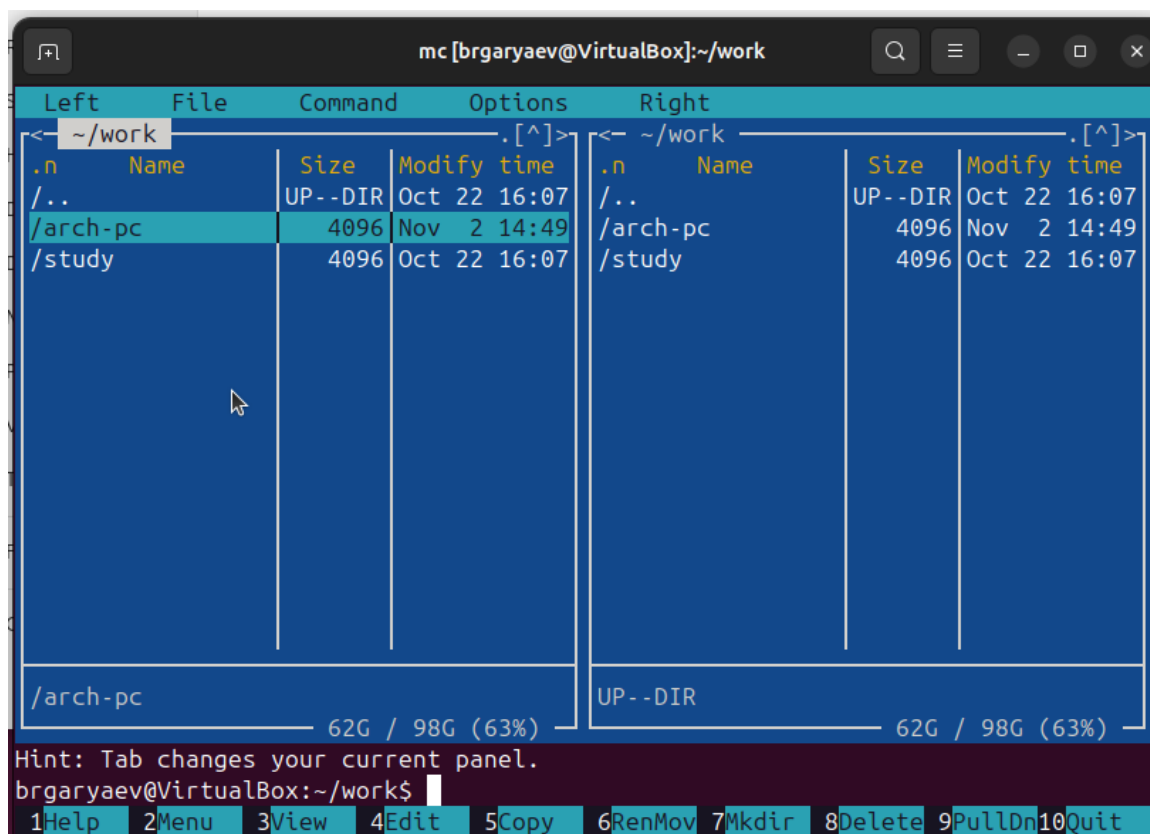


Рисунок 2.1: Запуск Midnight Commander

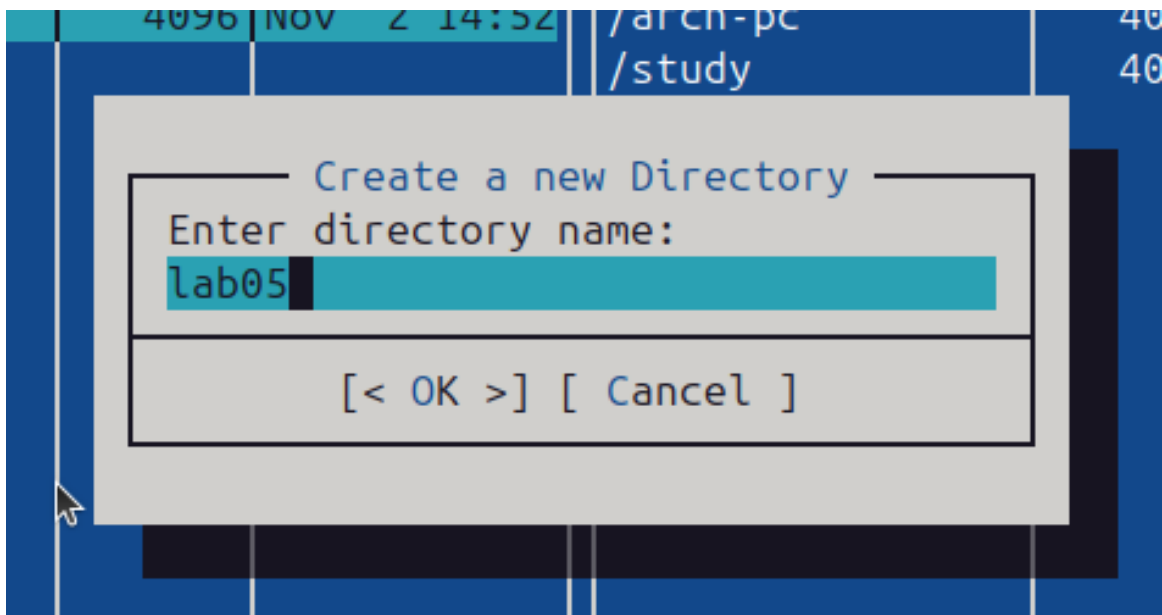


Рисунок 2.2: Создание каталога

С помощью команды `touch` создаю файл `lab05-1.asm` (см. рис. 2.3).

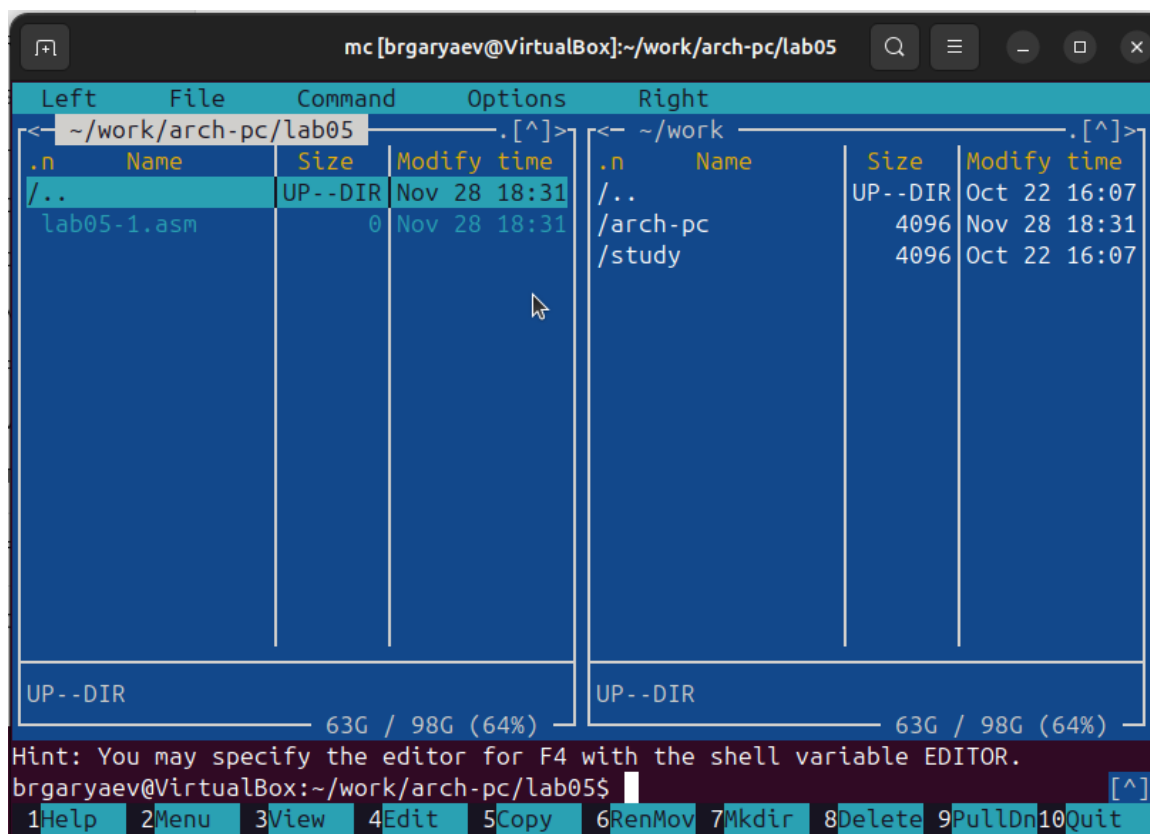
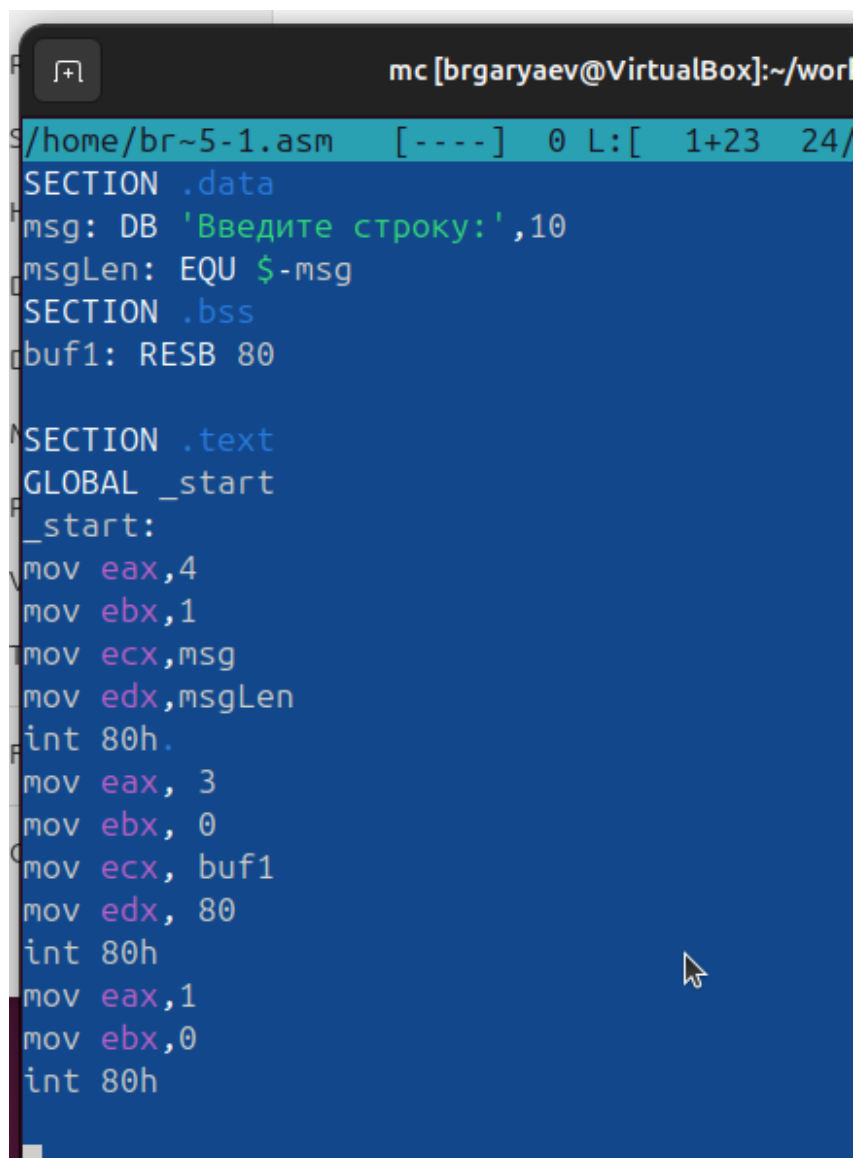


Рисунок 2.3: Создание файла lab05-1.asm

Открываю файл для редактирования, нажав F4, и выбираю редактор mcedit, после чего пишу код программы согласно заданию (см. рис. 2.5).


```
brgaryaev@VirtualBox:~/work$ mc
brgaryaev@VirtualBox:~/work/arch-pc/lab05$ touch lab05-1.asm
Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/mcedit
 3. /usr/bin/vim.tiny
 4. /bin/ed
Choose 1-4 [1]: 2
```

Рисунок 2.4: выбираю редактор

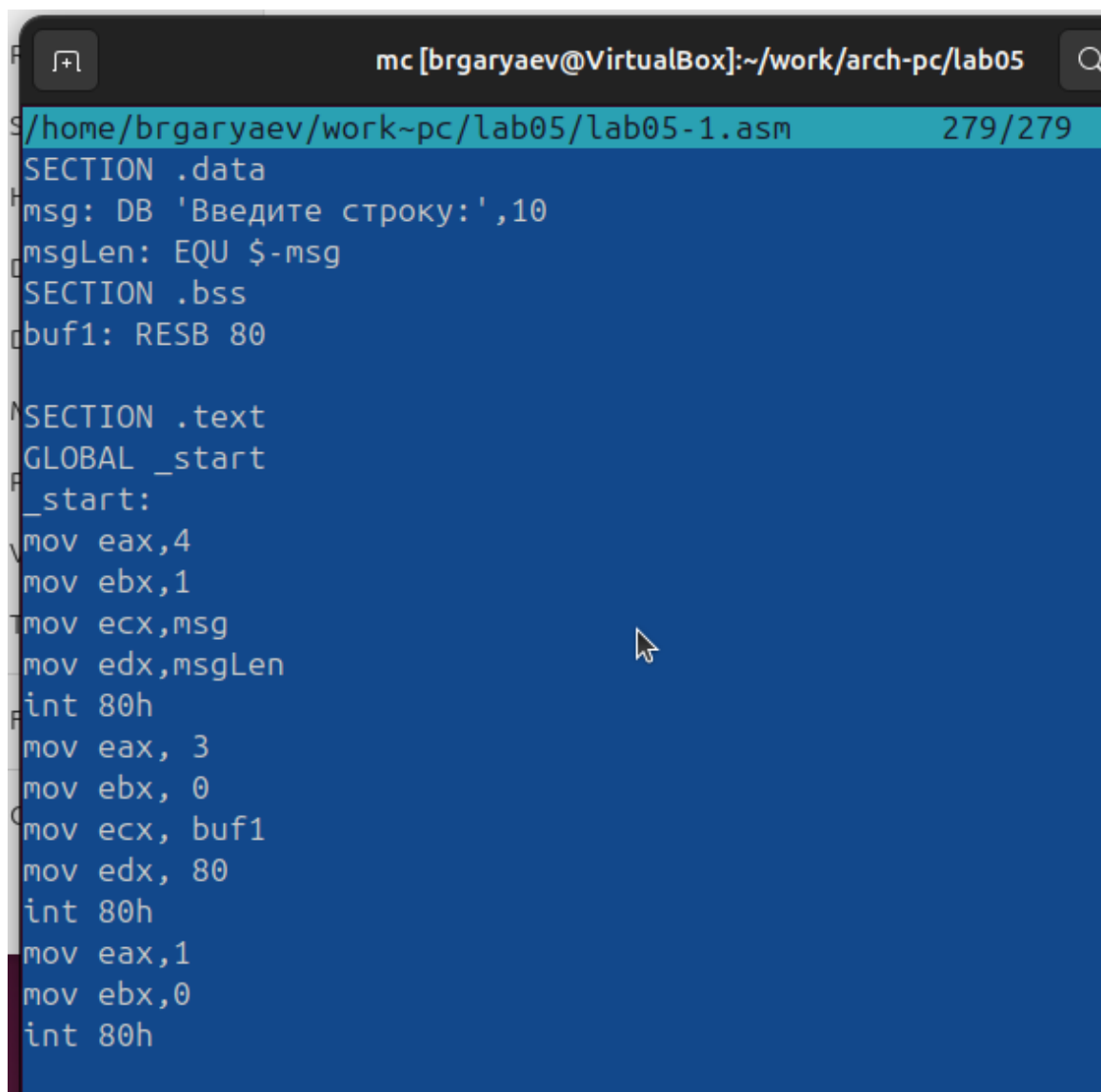


```
mc [brgaryaev@VirtualBox]:~/world
/home/br~5-1.asm [----] 0 L: [ 1+23 24/
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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: Программа lab05-1.asm

Для проверки содержимого файла открываю его на просмотр, нажав F3, и убеждаюсь, что код написан верно (см. рис. 2.6).



```
mc [brgaryaev@VirtualBox]:~/work/arch-pc/lab05
/home/brgaryaev/work~pc/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 2.6: Просмотр файла lab05-1.asm

Транслирую файл программы в объектный файл, а затем выполняю компоновку, в результате чего получаю исполняемый файл программы (см. рис. 2.7).

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

Рисунок 2.7: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

Скачиваю файл in_out.asm и размещаю его в рабочем каталоге (см. рис. 2.8). Для копирования файла использую клавишу F5, а для перемещения — клавишу F6.

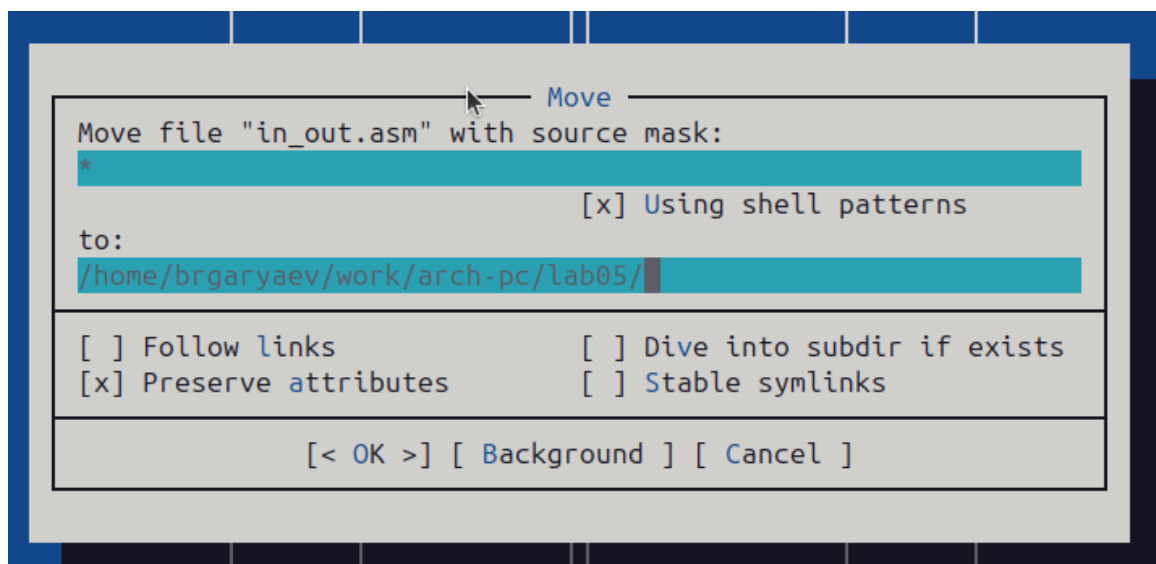


Рисунок 2.8: Копирование файла in_out.asm

Копирую файл lab05-1.asm, создавая его копию под именем lab05-2.asm (см. рис. 2.9).

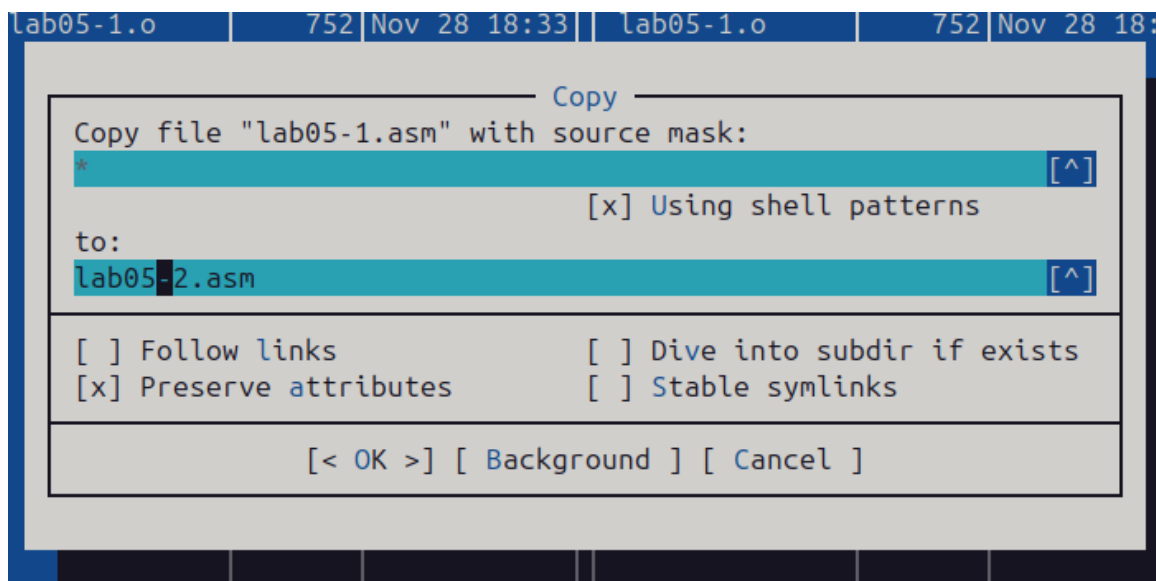
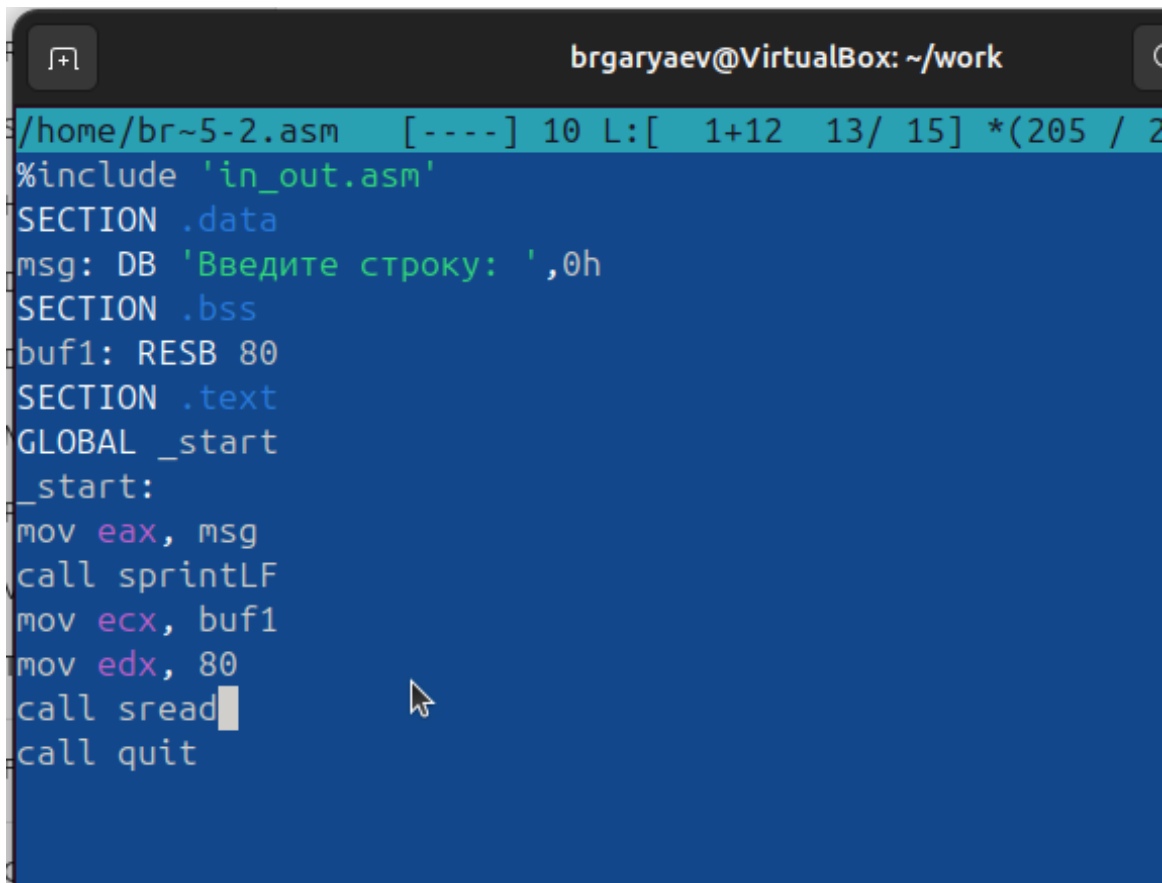


Рисунок 2.9: Копирование файла lab05-1.asm

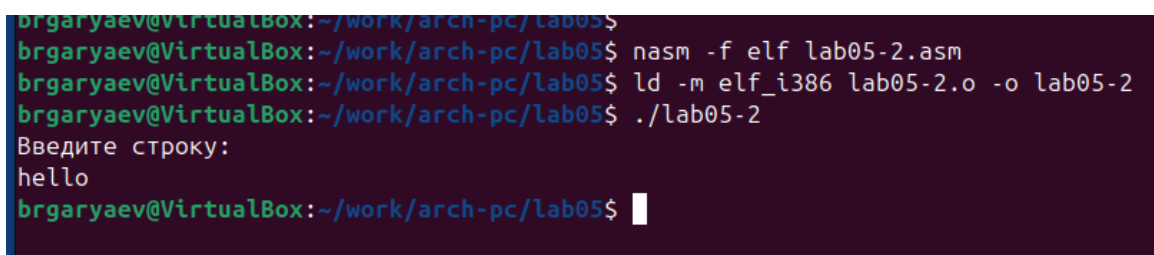
Пишу код для программы lab05-2.asm, используя подпрограммы из внешнего файла in_out.asm (см. рис. 2.10).



```
brgaryaev@VirtualBox: ~/work
/home/br~5-2.asm [----] 10 L:[ 1+12 13/ 15] *(205 / 2
#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 2.10: Программа lab05-2.asm

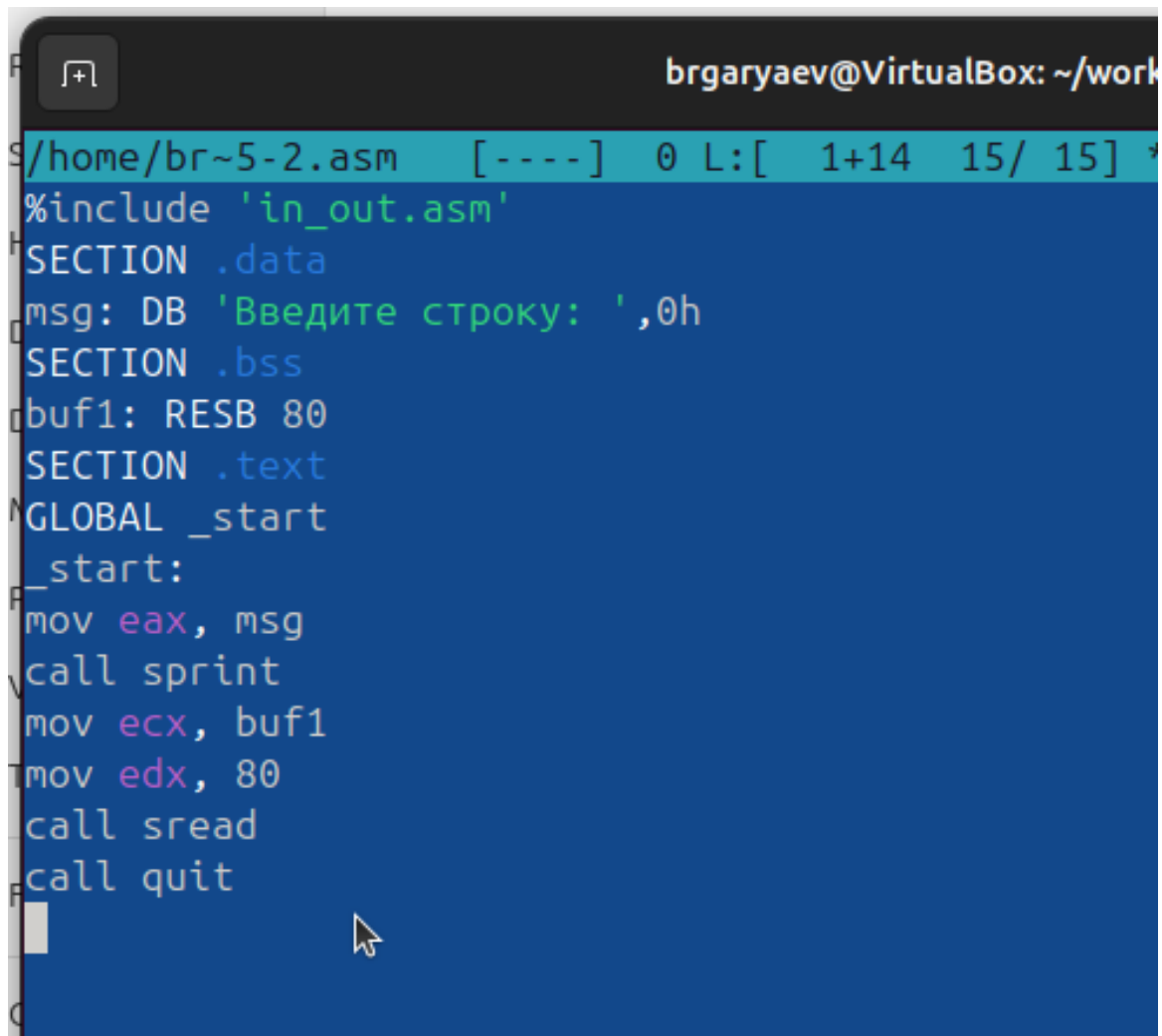
Компилирую программу и проверяю её запуск (см. рис. 2.11).



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

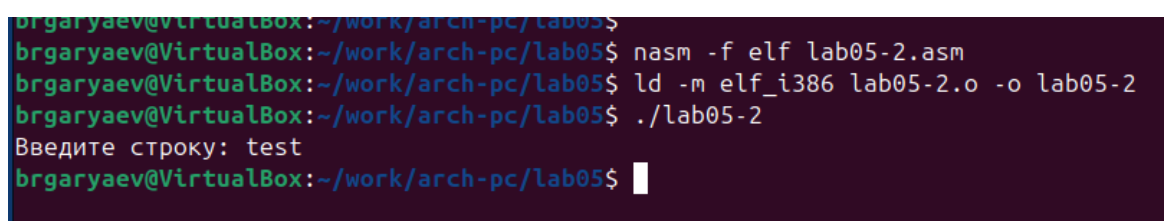
Рисунок 2.11: Запуск программы lab05-2.asm

В файле lab05-2.asm заменяю подпрограмму sprintLF на sprint. После этого заново собираю исполняемый файл (см. рис. 2.12 и 2.13).



```
brgaryaev@VirtualBox: ~/work
/home/br~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 sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рисунок 2.12: Программа в файле lab05-2.asm



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

Рисунок 2.13: Запуск программы lab05-2.asm

Теперь программа выводит строку без перехода на новую строку в конце.

2.3 Задание для самостоятельной работы

Копирую программу lab05-1.asm и модифицирую код, чтобы она работала по следующему алгоритму (см. рис. 2.14 и 2.15): - выводит приглашение «Введите строку:»; - принимает строку с клавиатуры; - отображает введенную строку на экране.



```
mc [brgaryaev@VirtualBox]:~/wor
/home/br~5-3.asm [ - - - ] 0 L: [ 1+12 13/
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

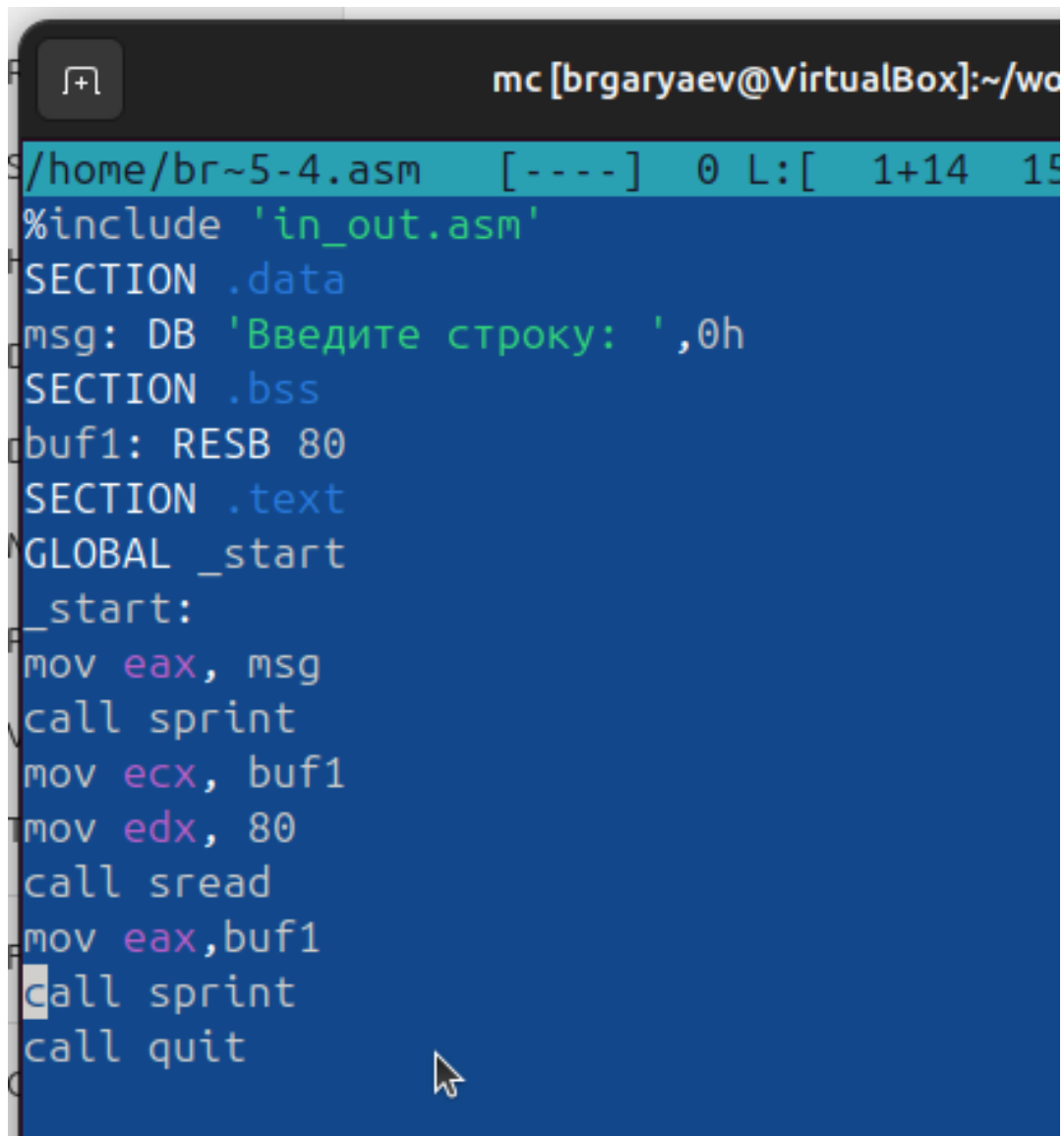
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рисунок 2.14: Программа lab05-3.asm


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

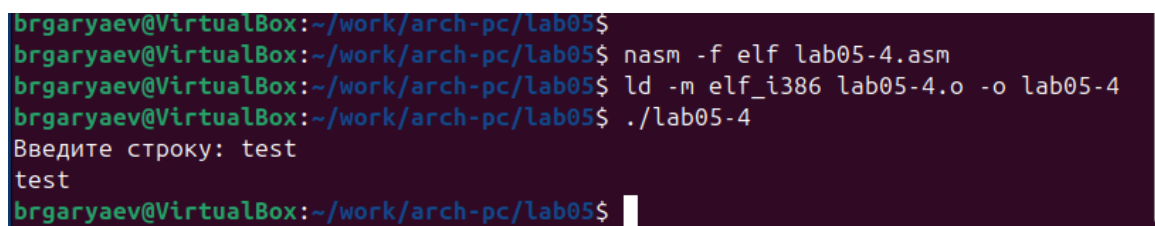
Рисунок 2.15: Запуск программы lab05-3.asm

Аналогично, копирую программу lab05-2.asm и изменяю код, теперь используя подпрограммы из файла in_out.asm (см. рис. 2.16 и 2.17).



```
mc [brgaryaev@VirtualBox]:~/wo
/home/br~5-4.asm [ - - - - ] 0 L:[ 1+14 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 sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax,buf1
call sprint
call quit
```

Рисунок 2.16: Программа lab05-4.asm



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

Рисунок 2.17: Запуск программы lab05-4.asm

3 Выводы

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