

Отчёт по лабораторной работе 5

Архитектура компьютеров

Ел Вакил Марьям Махмоудовна НБИбд-03-23

Содержание

1	Цель работы	5
2	Задания	6
3	Выполнение лабораторной работы	7
3.1	Основы работы в Midnight Commander	7
3.2	Подключение внешнего файла in_out.asm	12
3.3	Задание для самостоятельной работы	17
4	Выводы	22

Список иллюстраций

3.1	Создание каталога	8
3.2	Создание файла lab05-1.asm	9
3.3	Код программы	10
3.4	Просмотр файла lab05-1.asm	11
3.5	Проверка программы lab05-1.asm	12
3.6	Копирование файла	13
3.7	Код программы	14
3.8	Проверка программы lab05-2.asm	15
3.9	Код программы	16
3.10	Проверка программы lab05-2.asm	17
3.11	Код программы	18
3.12	Проверка программы lab05-3.asm	19
3.13	Код программы	20
3.14	Проверка 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 Задания

1. Изучение Midnight Commander
2. Рассмотреть примеры программ ввода/вывода
3. Рассмотреть примеры программ ввода/вывода с использованием внешнего файла `in_out.asm`
4. Выполнение заданий для самостоятельной работы.

3 Выполнение лабораторной работы

3.1 Основы работы в Midnight Commander

Я запустила Midnight Commander и направились в каталог `~/work/arch-pc`, чтобы приступить к выполнению лабораторной работы. Внутри этого каталога создала новую папку с именем `lab05` для хранения файлов текущего задания.

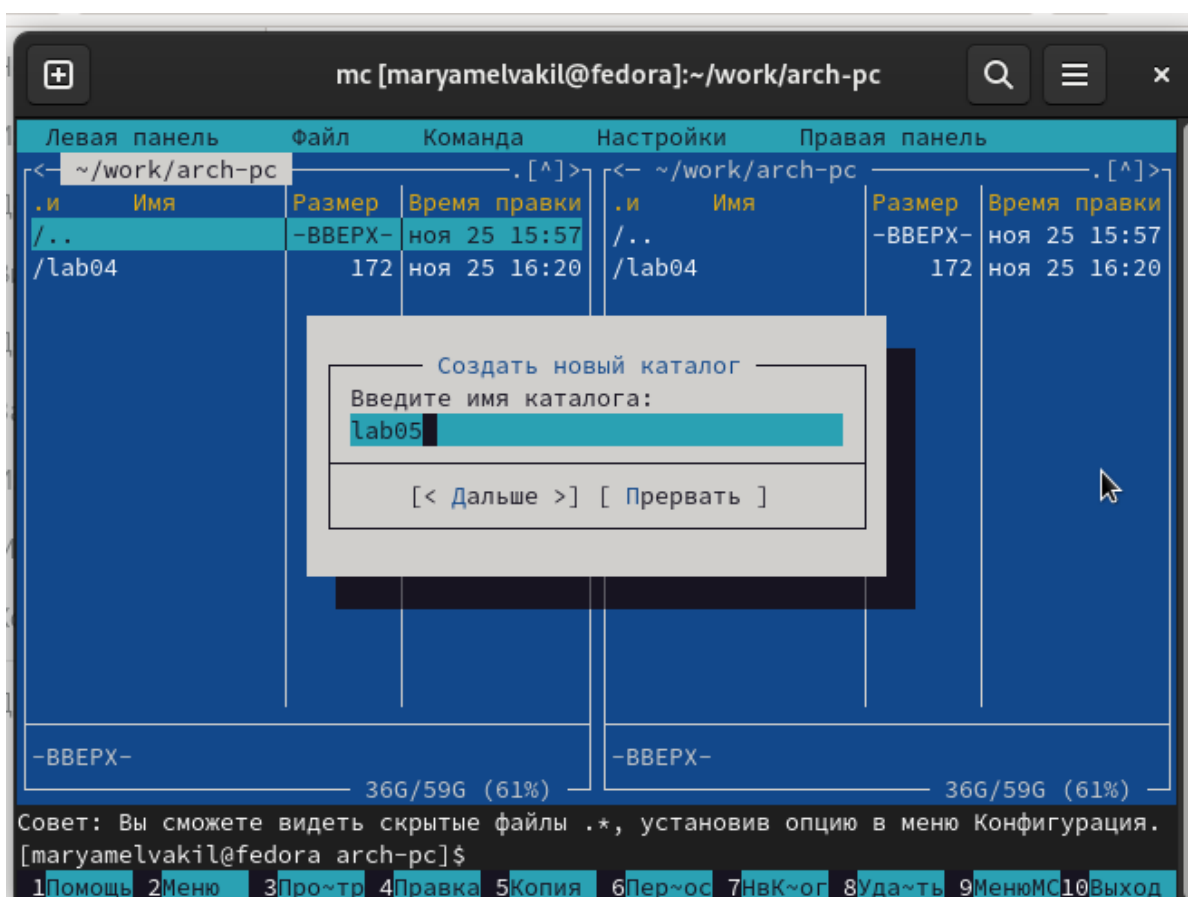


Рис. 3.1: Создание каталога

Далее внутри папки lab05 создаю файл под названием lab05-1.asm - это будет мой рабочий файл для написания кода программы в ассемблере.

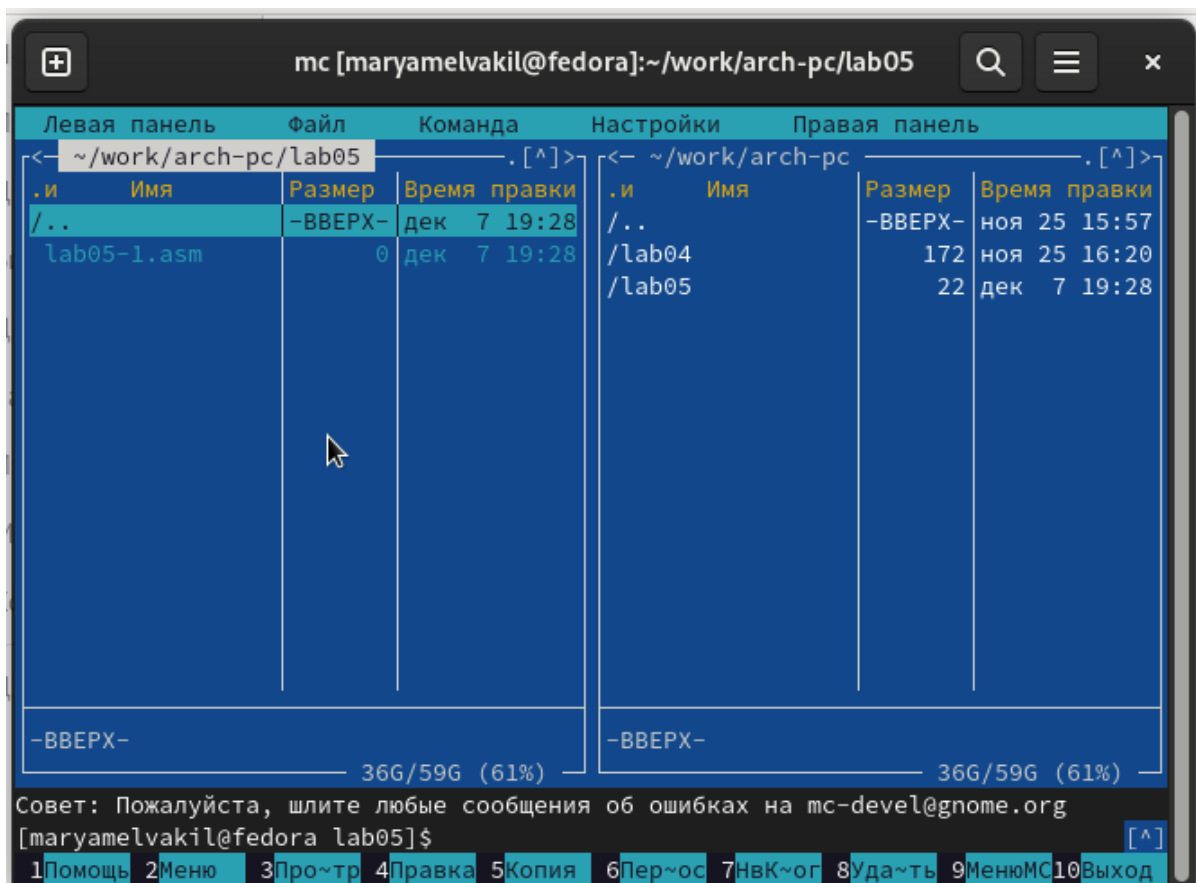
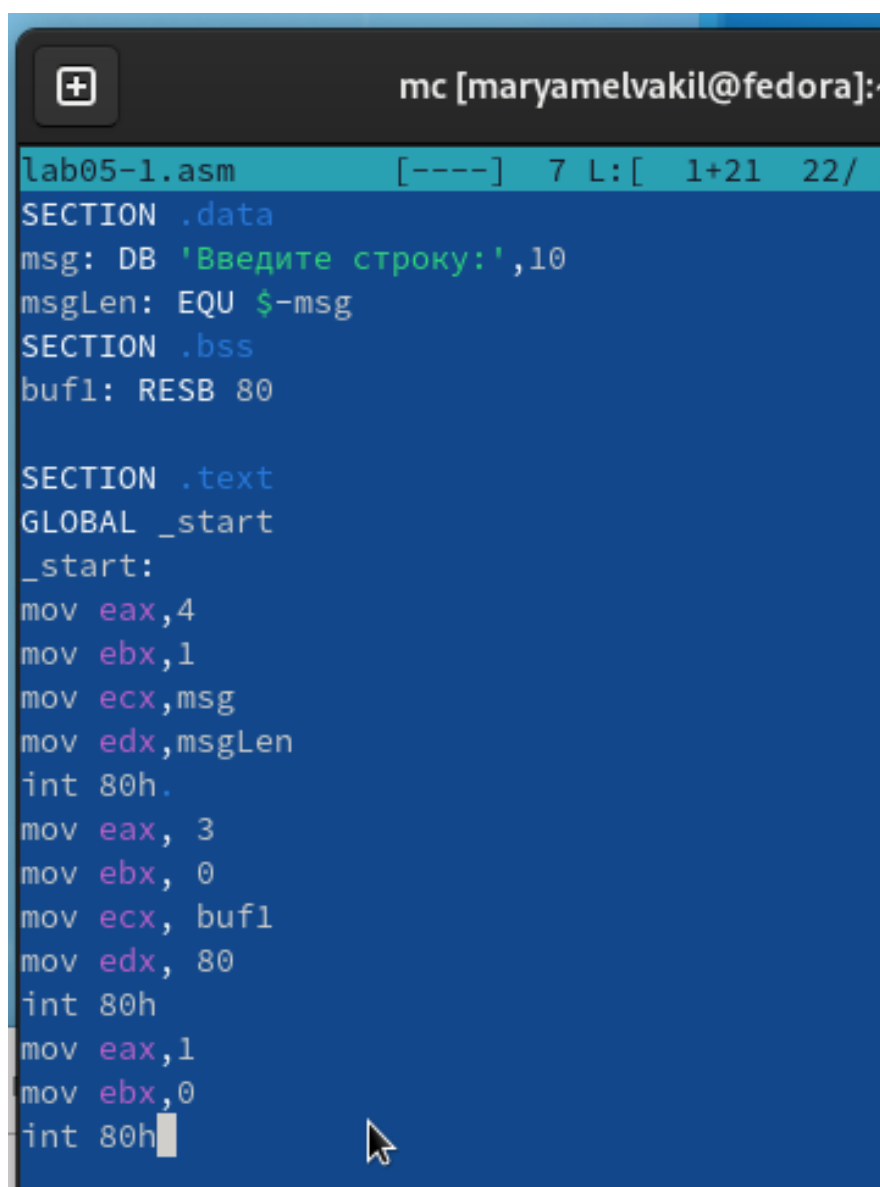


Рис. 3.2: Создание файла lab05-1.asm

Я открыла файл lab05-1.asm для редактирования и написала в нем код программы.

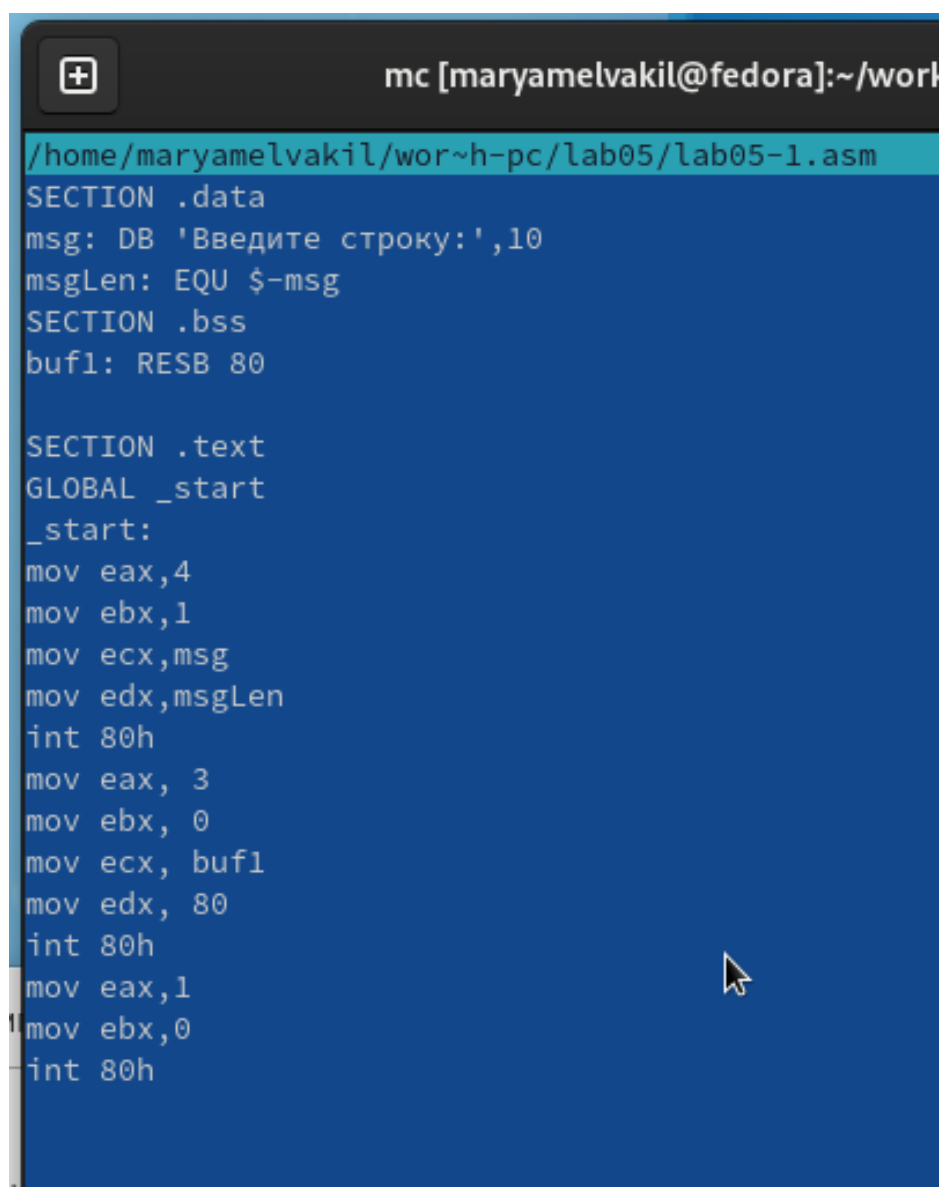


```
lab05-1.asm [----] 7 L:[ 1+21 22/
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

Рис. 3.3: Код программы

После этого открыла файл для просмотра и убедилась, что он содержит написанный мной код.



```
mc [maryamelvakil@fedora]:~/work
/home/maryamelvakil/work-h-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
```

Рис. 3.4: Просмотр файла lab05-1.asm

Затем скомпилировала программу и проверила ее работу, получив исполняемый файл.

```
[maryam@fedora lab05]$ nasm -f elf lab05-1.asm
[maryam@fedora lab05]$ ld -m elf_i386 lab05-1.o -o lab05-1
[maryam@fedora lab05]$ ./lab05-1
Введите строку:
Maryam
[maryam@fedora lab05]$
```

Рис. 3.5: Проверка программы lab05-1.asm

3.2 Подключение внешнего файла in_out.asm

Загрузила вспомогательный файл in_out.asm, содержащий полезные подпрограммы, и сохранила его в своей рабочей папке. Затем я скопировала файл lab05-1.asm и создала копию с именем lab05-2.asm.

Скопирую файл с программой

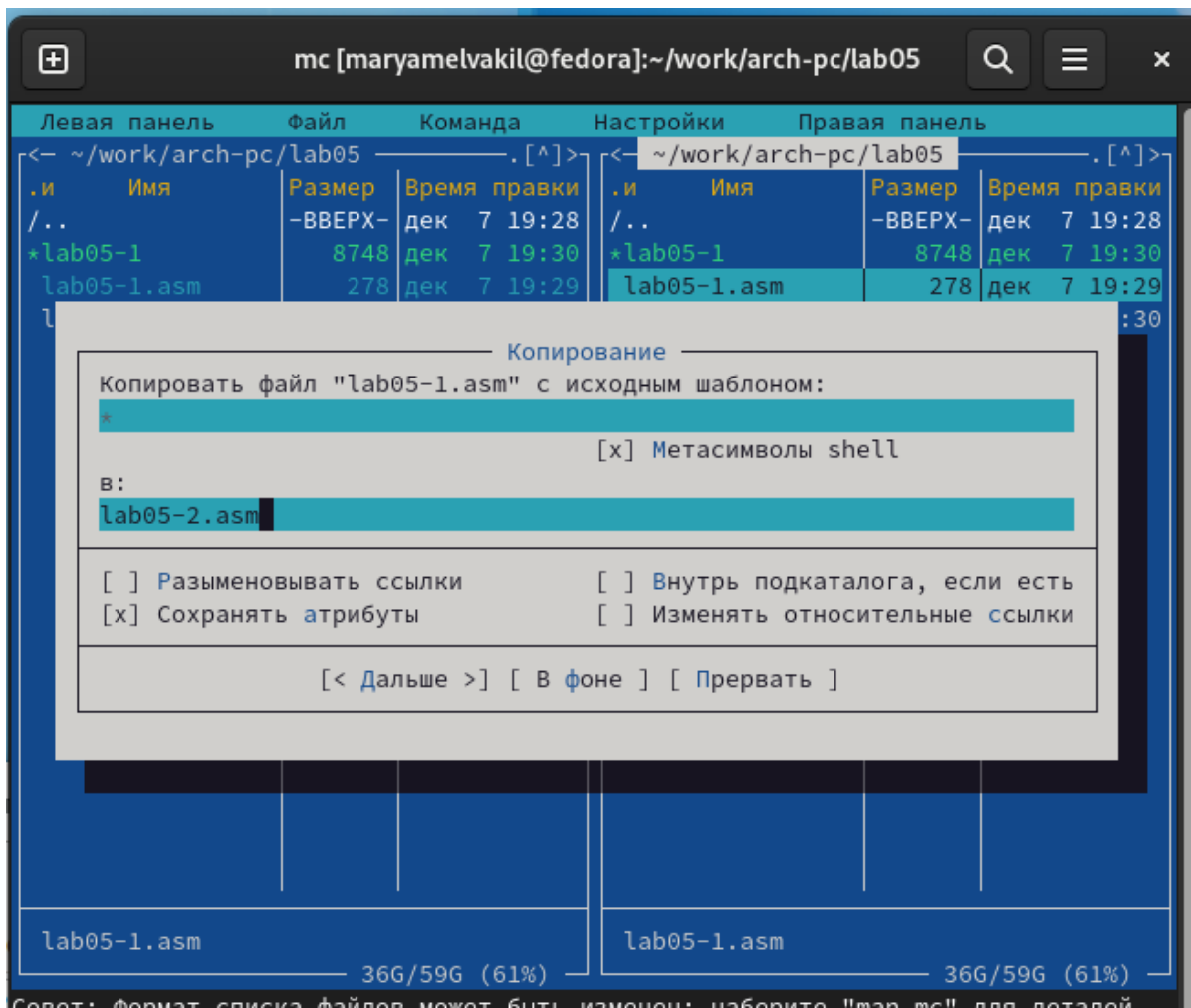
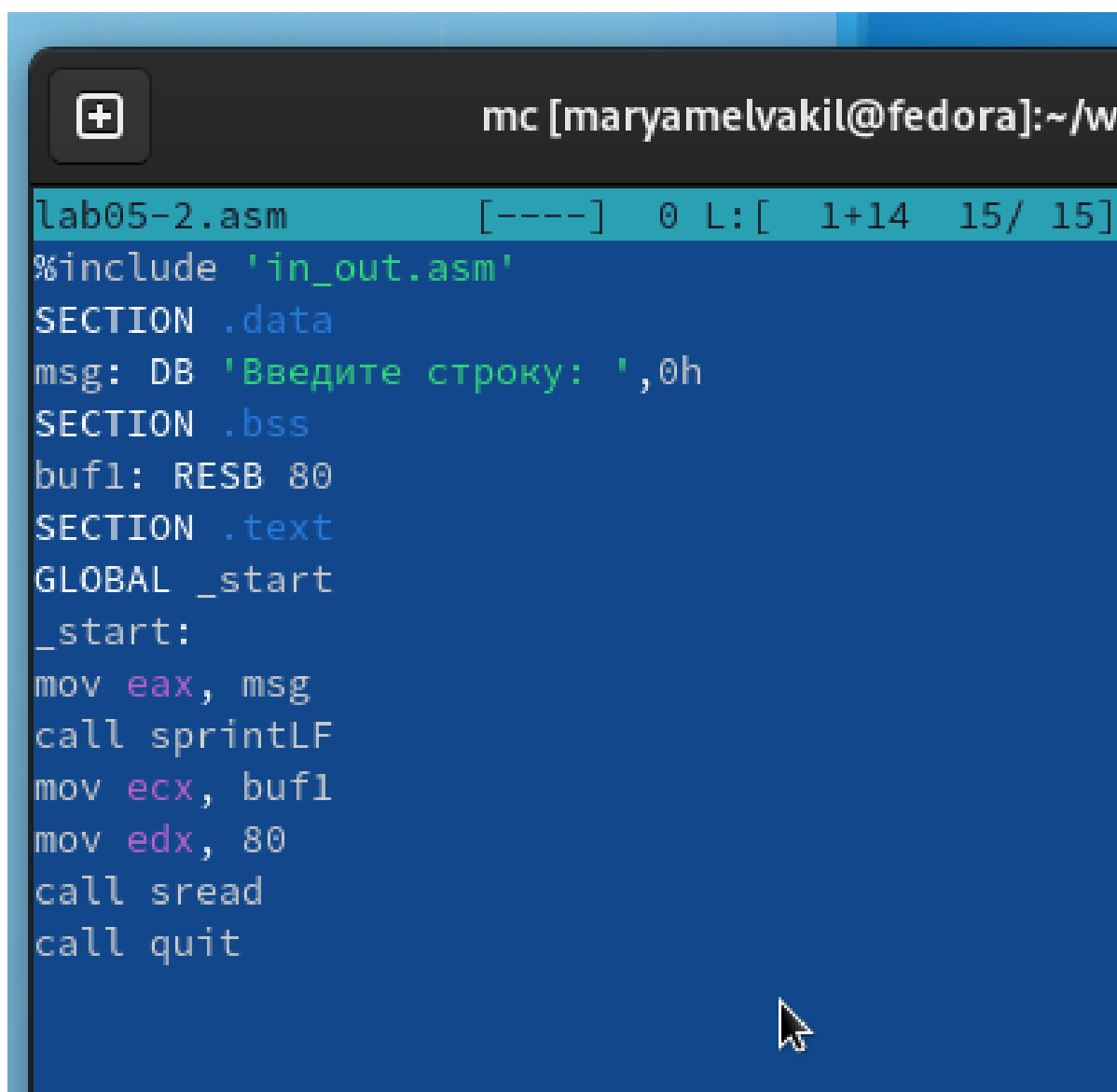


Рис. 3.6: Копирование файла

Написала код программы в файле lab05-2.asm.



The image shows a code editor window with a dark theme. The title bar at the top indicates the user is 'mc [maryamelvakil@fedora]:~/w'. The editor displays the file 'lab05-2.asm' with a status bar showing '[----] 0 L: [1+14 15/ 15]'. The code is as follows:

```
lab05-2.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
```

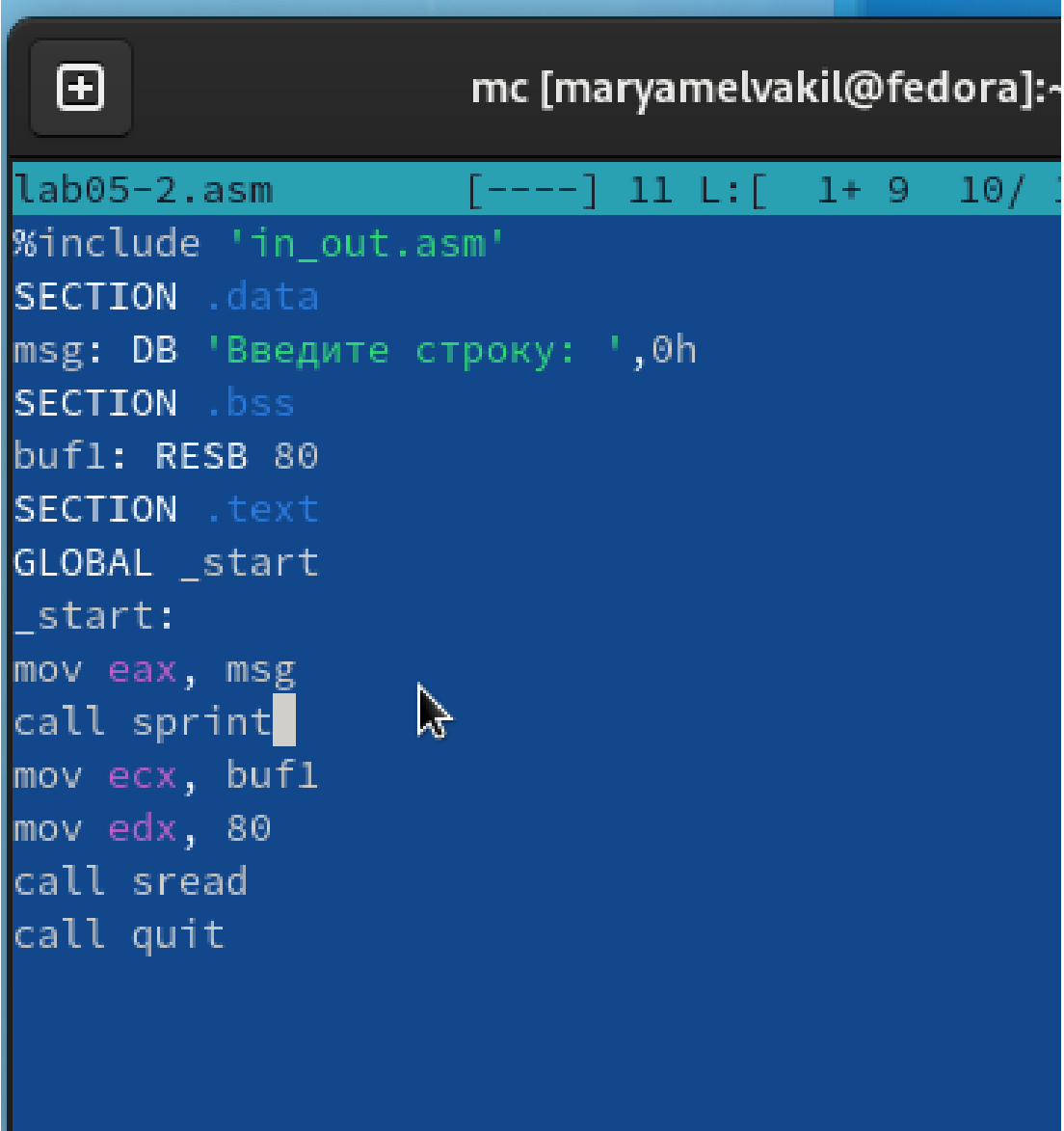
Рис. 3.7: Код программы

После этого я скомпилировала программу и проверила ее запуск.

```
[maryamelvakil@fedora lab05]$  
[maryamelvakil@fedora lab05]$ nasm -f elf lab05-2.asm  
[maryamelvakil@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2  
[maryamelvakil@fedora lab05]$ ./lab05-2  
Введите строку:  
Maryam  
[maryamelvakil@fedora lab05]$
```

Рис. 3.8: Проверка программы lab05-2.asm

В файле lab05-2.asm внесла изменения, заменив подпрограмму sprintLF на sprint.



```
mc [maryamelvakil@fedora]:~  
lab05-2.asm [-----] 11 L: [ 1+ 9 10/ :  
%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
```

Рис. 3.9: Код программы

Это позволило строке вывода не завершаться символом перехода на новую строку.


```
[maryamelvakil@fedora lab05]$ nasm -f elf lab05-2.asm
[maryamelvakil@fedora lab05]$ ld -m elf_i386 lab05-2.o -o lab05-2
[maryamelvakil@fedora lab05]$ ./lab05-2
Введите строку: Maryam
[maryamelvakil@fedora lab05]$
```

Рис. 3.10: Проверка программы lab05-2.asm

3.3 Задание для самостоятельной работы

Нужно скопировать lab05-1.asm и внести изменения в код, чтобы программа выводила приглашение “Введите строку:”, затем считывала строку с клавиатуры и выводила ее на экран.



```
mc [maryamelvakil@fedora]
lab05-3.asm [-----] 9 L: [ 1+25 26/
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

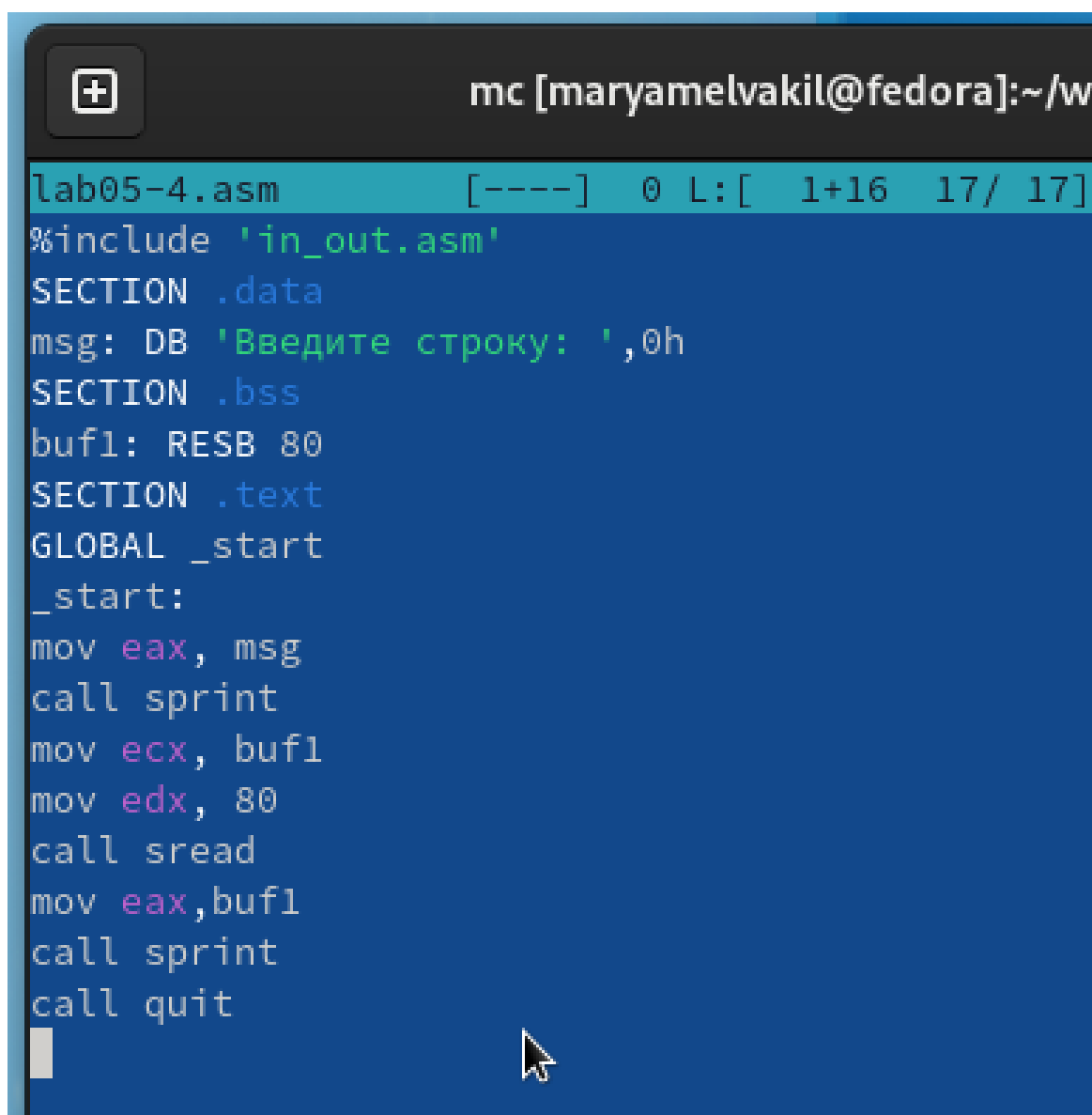
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 3.11: Код программы

```
[maryam@fedora lab05]$  
[maryam@fedora lab05]$ nasm -f elf lab05-3.asm  
[maryam@fedora lab05]$ ld -m elf_i386 lab05-3.o -o lab05-3  
[maryam@fedora lab05]$ ./lab05-3  
Введите строку:  
Maryam  
Maryam  
[maryam@fedora lab05]$  
[maryam@fedora lab05]$
```

Рис. 3.12: Проверка программы lab05-3.asm

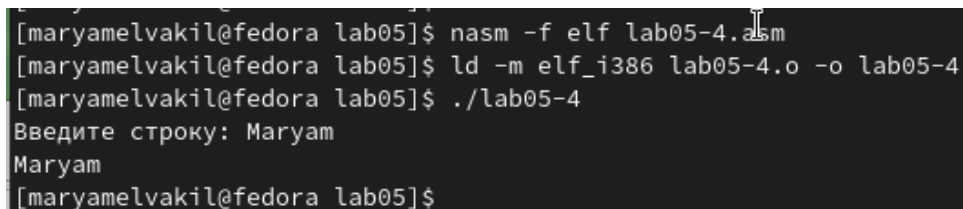
Также нужно скопировать программу lab05-2.asm и сделать аналогичные изменения.



The screenshot shows a code editor window titled 'mc [maryamelvakil@fedora]:~/w'. The file 'lab05-4.asm' is open, with a status bar indicating line 17 of 17. The code is as follows:

```
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
```

Рис. 3.13: Код программы



The screenshot shows a terminal window with the following commands and output:

```
[maryamelvakil@fedora lab05]$ nasm -f elf lab05-4.asm
[maryamelvakil@fedora lab05]$ ld -m elf_i386 lab05-4.o -o lab05-4
[maryamelvakil@fedora lab05]$ ./lab05-4
Введите строку: Maryam
Maryam
[maryamelvakil@fedora lab05]$
```

Рис. 3.14: Проверка программы lab05-4.asm

Отличие между этими двумя реализациями заключается в том, что файл `in_out.asm` уже содержит готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать нужную подпрограмму с помощью инструкции `call`. Это упрощает кодирование и обеспечивает более гибкую работу с вводом и выводом данных.

4 Выводы

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