

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ЛАБОРАТОРНАЯ РАБОТА №5

**на тему « ОСНОВЫ РАБОТЫ С MIDNIGHT COMMANDER
(МС). СТРУКТУРА ПРОГРАММЫ НА ЯЗЫКЕ АССЕМБЛЕРА
NASM. СИСТЕМНЫЕ ВЫЗОВЫ В ОС GNU LINUX»**

дисциплина: Архитектура компьютера

Студент: Трусова А.А.

Группа: НКАбд-05-24

№ ст. билета: 1132246715

МОСКВА

2024г.

Содержание

Цель работы

Теоретическое введение

Выполнение лабораторной работы

4.1. Работа с тс и программа на языке ассемблера NASM

4.2. Подключение внешнего файла

4.3. Задание для самостоятельной работы

Вывод

Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициализированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициализированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти: - DB (define byte) — определяет переменную размером в 1 байт; - DW (define word) — определяет переменную размером в 2 байта (слово); - DD (define double word) — определяет переменную размером в 4 байта (двойное слово); - DQ (define quad word) — определяет переменную размером в 8 байт (четырёх-байтное слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

mov dst,src

Здесь операнд dst — приёмник, а src — источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

int n

Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

Выполнение лабораторной работы

4.1. Работа с mc и программа на языке ассемблера NASM

Открыла Midnight commander с помощью mc (рис.1).

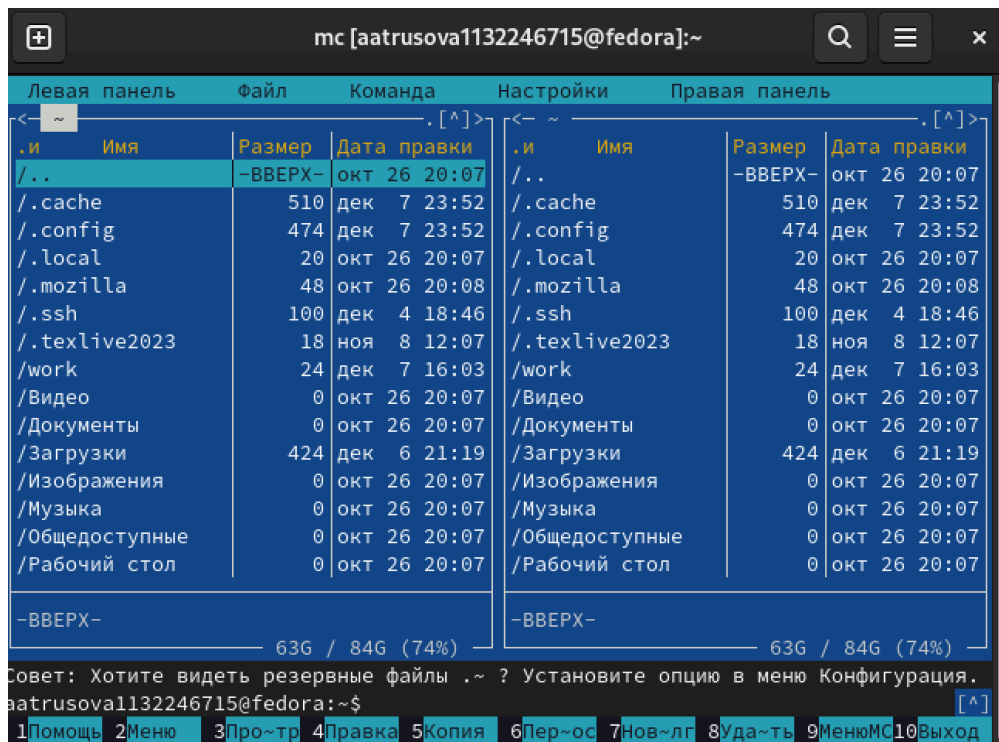


Рис.1

С помощью стрелочек и enter перешла в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 (рис. 2).

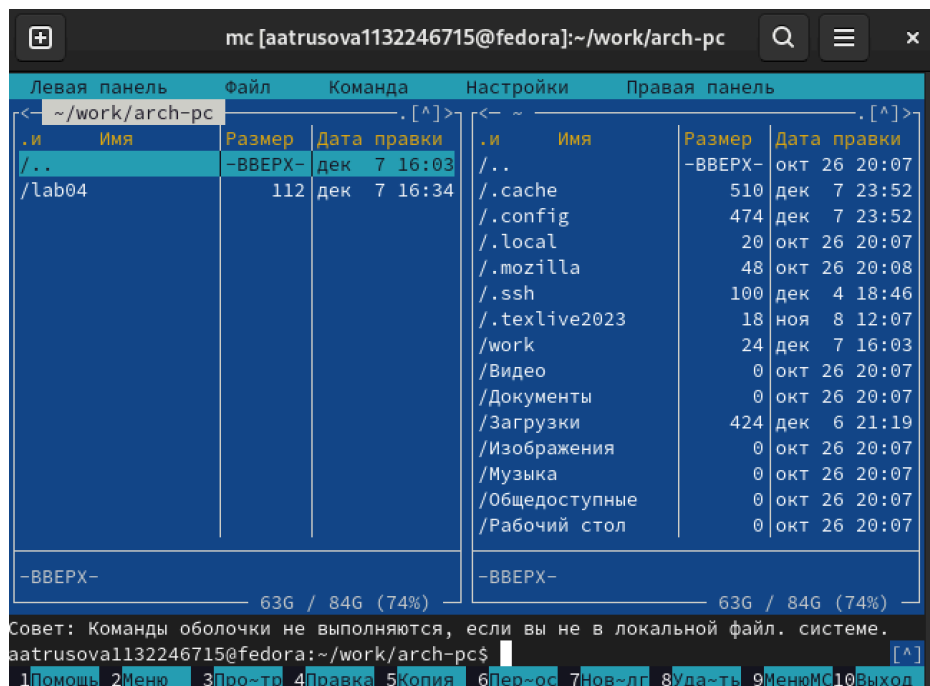


Рис.2

С помощью функциональной клавиши F7 создала папку lab05 и перешла в него (рис.3).

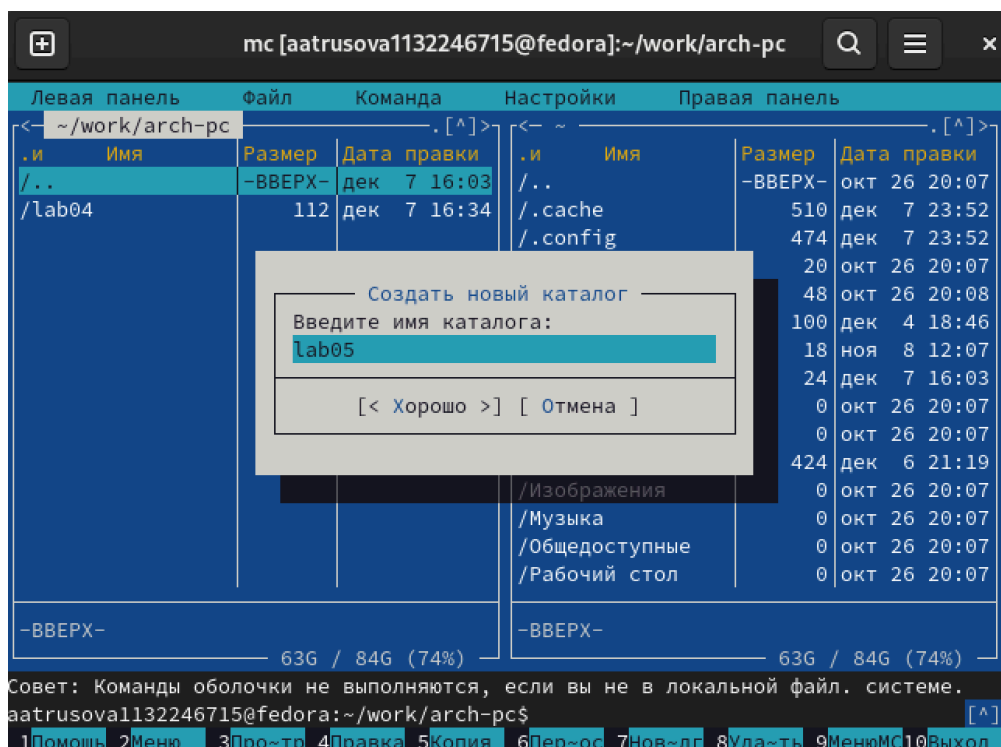


Рис.3

Пользуясь строкой ввода и командой touch, создала файл lab5-1.asm (рис.4).

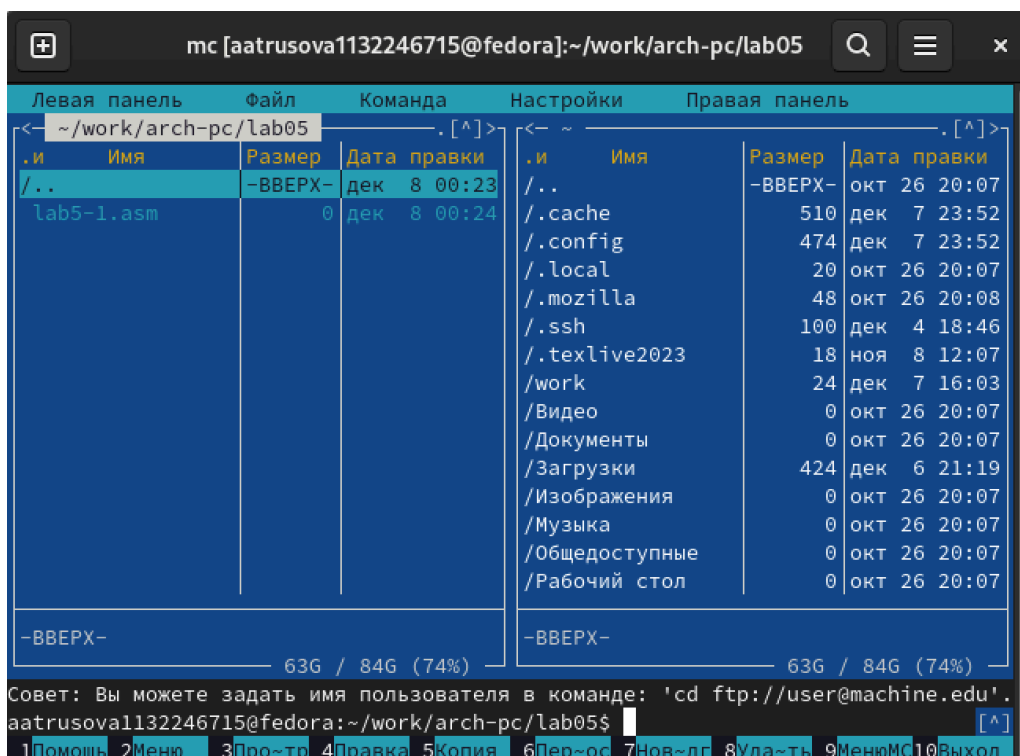
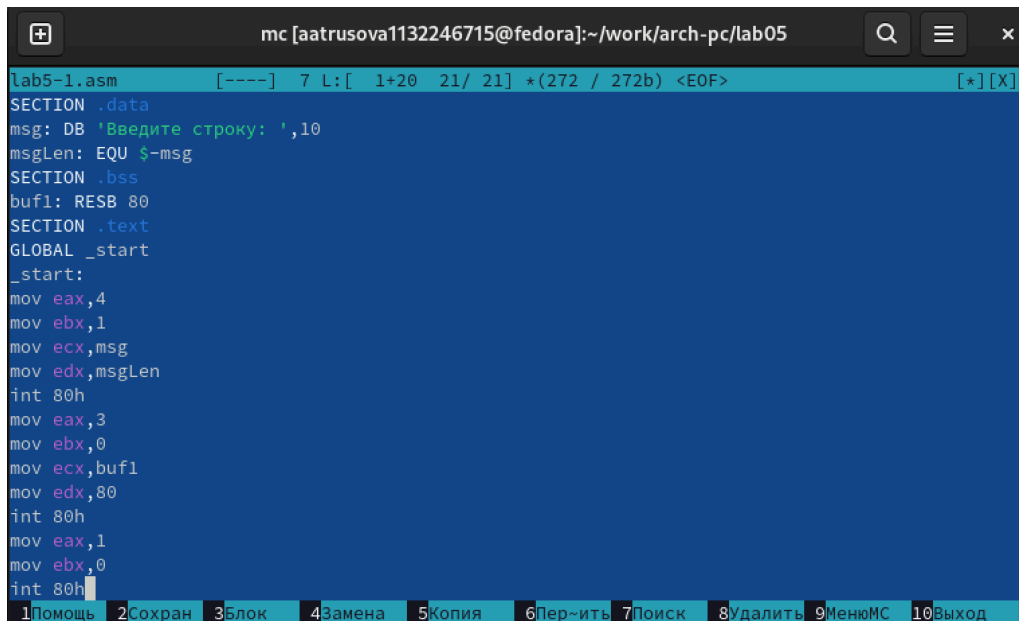


Рис.4

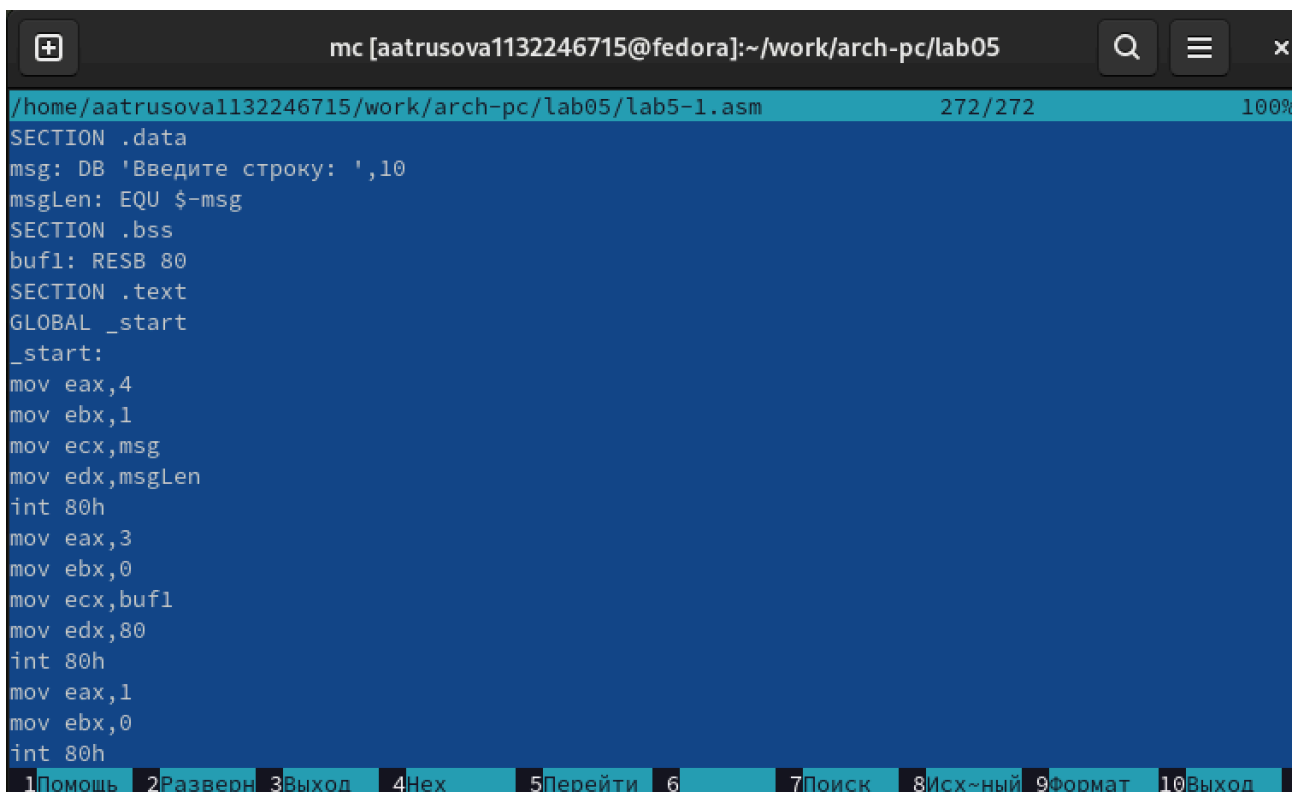
С помощью функциональной клавиши F4 открыла файл lab5-1.asm для редактирования во встроенном редакторе mcedit, ввела текст программы, сохранила изменения и закрыла файл (рис.5).



```
mc [aatrusova1132246715@fedora]:~/work/arch-pc/lab05
lab5-1.asm [----] 7 L: [ 1+20 21/ 21] *(272 / 272b) <EOF> [*] [X]
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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис.5

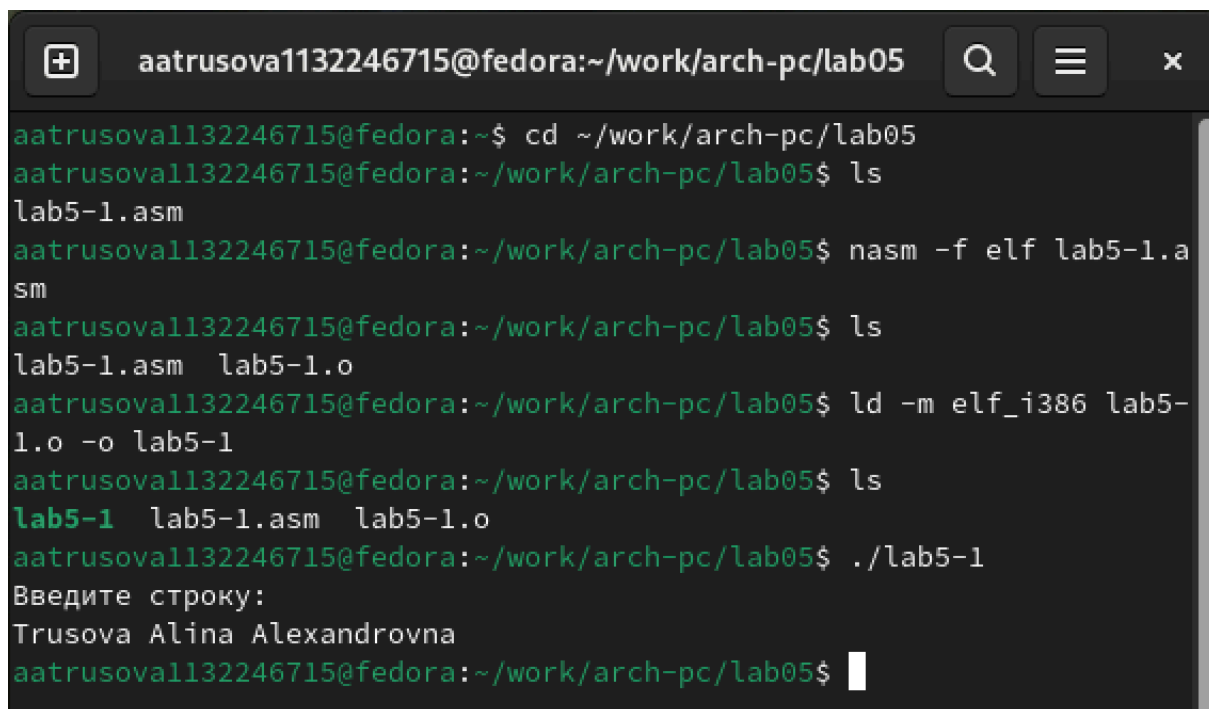
С помощью функциональной клавиши F3 открыла файл lab5-1.asm для просмотра. Файл содержит текст программы (рис.6).



```
mc [aatrusova1132246715@fedora]:~/work/arch-pc/lab05
/home/aatrusova1132246715/work/arch-pc/lab05/lab5-1.asm 272/272 100%
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
1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исх~ный 9Формат 10Выход
```

Рис.6

Оттранслировала текст программы lab5-1.asm в объектный файл, выполнила компоновку объектного файла и запустила получившийся исполняемый файл . На запрос ввела своё ФИО (рис.7).

A terminal window with a dark background and light green text. The window title is 'aatrusova1132246715@fedora:~/work/arch-pc/lab05'. The terminal shows the following commands and output:

```
aatrusova1132246715@fedora:~$ cd ~/work/arch-pc/lab05
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ls
lab5-1.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ls
lab5-1.asm  lab5-1.o
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1.o -o lab5-1
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ls
lab5-1  lab5-1.asm  lab5-1.o
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Trusova Alina Alexandrovna
aatrusova1132246715@fedora:~/work/arch-pc/lab05$
```

Рис.7

4.2. Подключение внешнего файла

Скачала файл in_out.asm со страницы курса в ТУИС. Он переместился в каталог Загрузки. В одной из панелей mc открыла каталог с файлом lab5-1.asm, в другой — каталог со скаченным файлом in_out.asm (рис. 8).

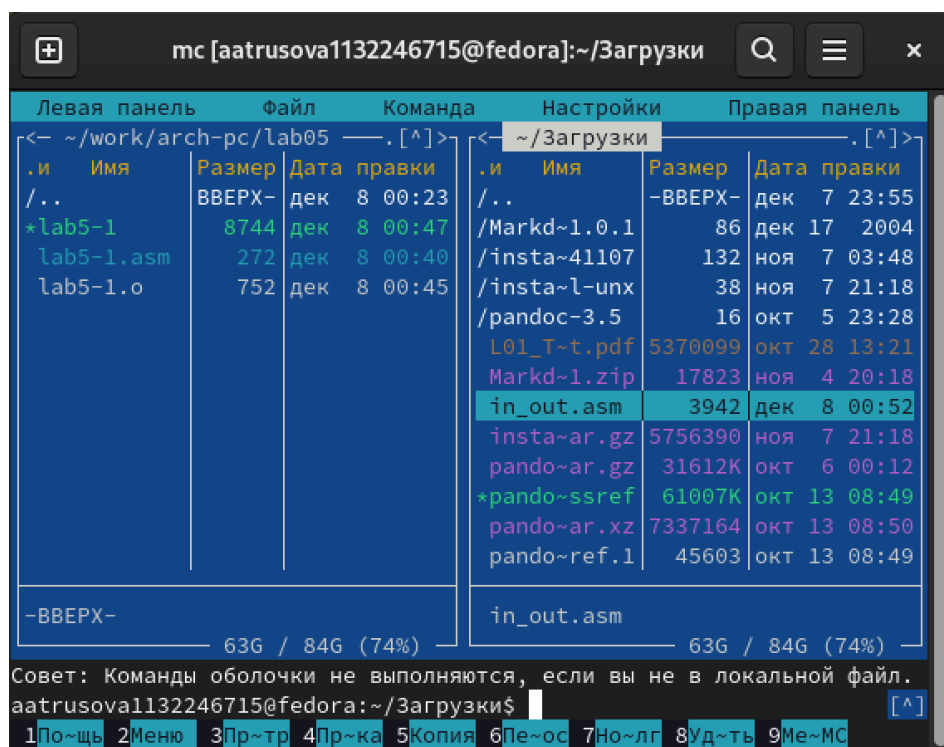


Рис.8

Скопировала файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис.9).

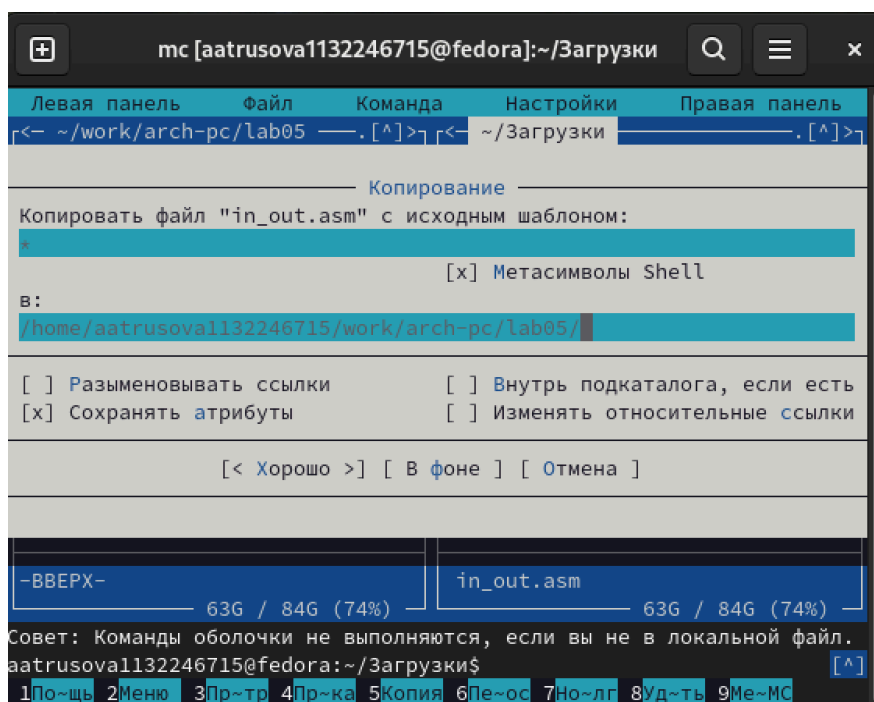


рис.9

С помощью функциональной клавиши F6 создала копию файла lab5-1.asm с именем lab5-2.asm (рис.10).

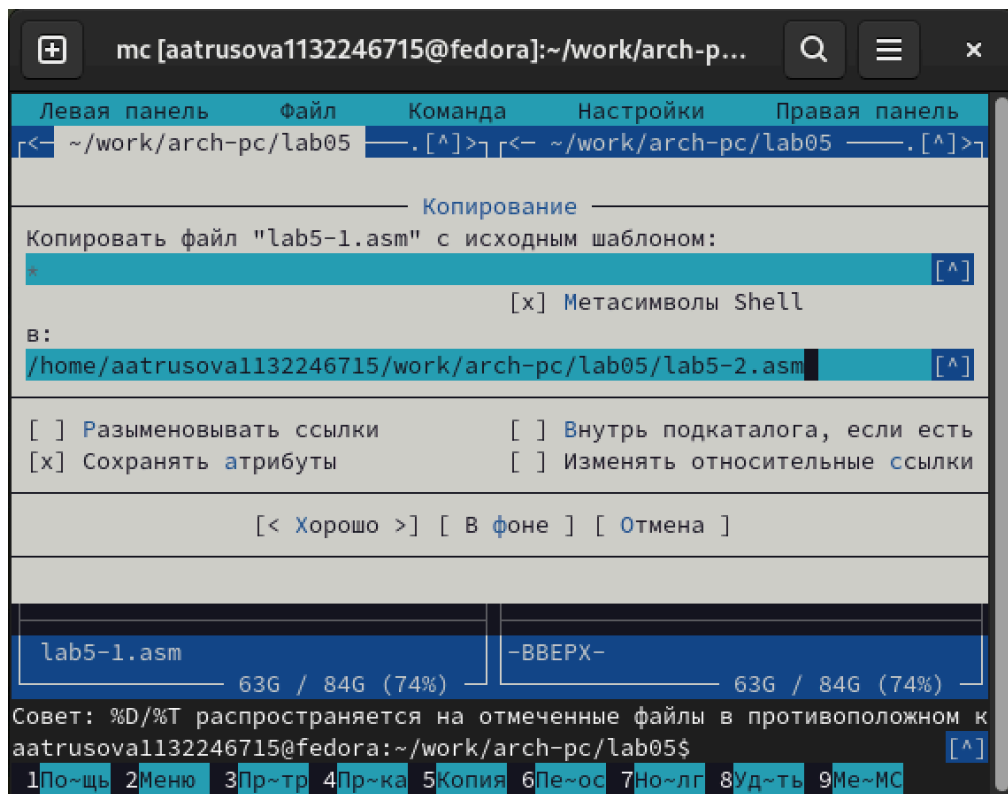


Рис.10

Исправила текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (рис.11).

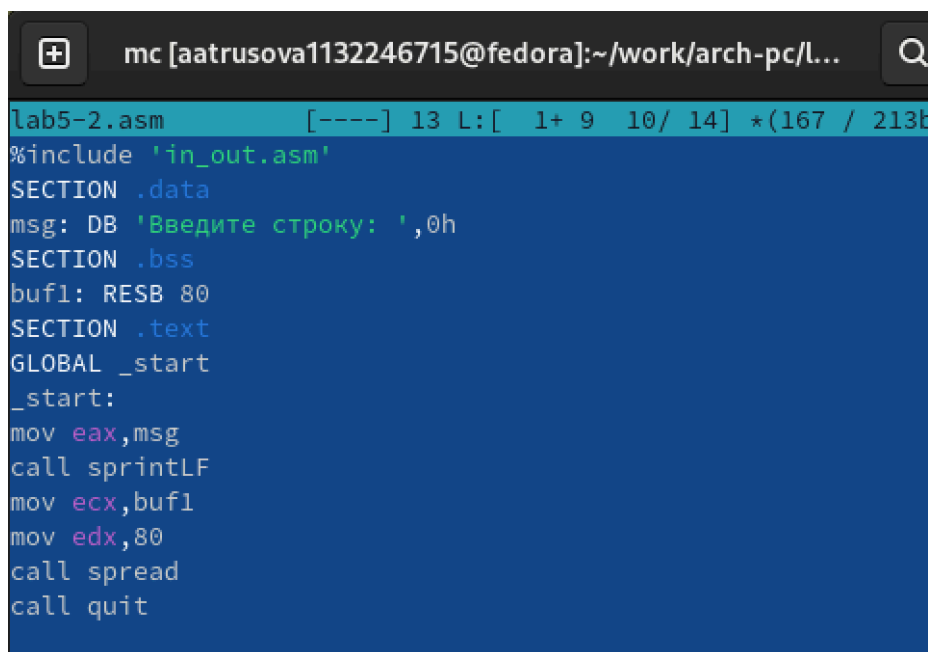


Рис.11

Создала исполняемый файл и проверила его работу (рис.12).

```
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o
lab5-2
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Trsuova Alina Alexandrovna
aatrusova1132246715@fedora:~/work/arch-pc/lab05$
```

Рис.12

В файле lab5-2.asm заменила подпрограмму sprintLF на sprint (рис.13).

```
lab5-2.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.13

Создала исполняемый файл и проверила его работу. Исправленный файл не переносит строку после запроса строки (рис.14).

```
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o lab
5-2
ld: lab5-2: в функции «_start»:
lab5-2.asm:(.text+0xe8): повторное определение «_start»; lab5-2.o:lab5-2.asm
:(.text+0xe8): здесь первое определение
ld: cannot use executable file 'lab5-2' as input to a link
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o lab
5-3
ld: невозможно найти lab5-3: Нет такого файла или каталога
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o
lab5-2
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Trusova Alina Alexandrovna
aatrusova1132246715@fedora:~/work/arch-pc/lab05$
```

Рис.14

4.3. Задание для самостоятельной работы

Создала копию файла lab5-1.asm с именем lab5-11.asm (рис.15).

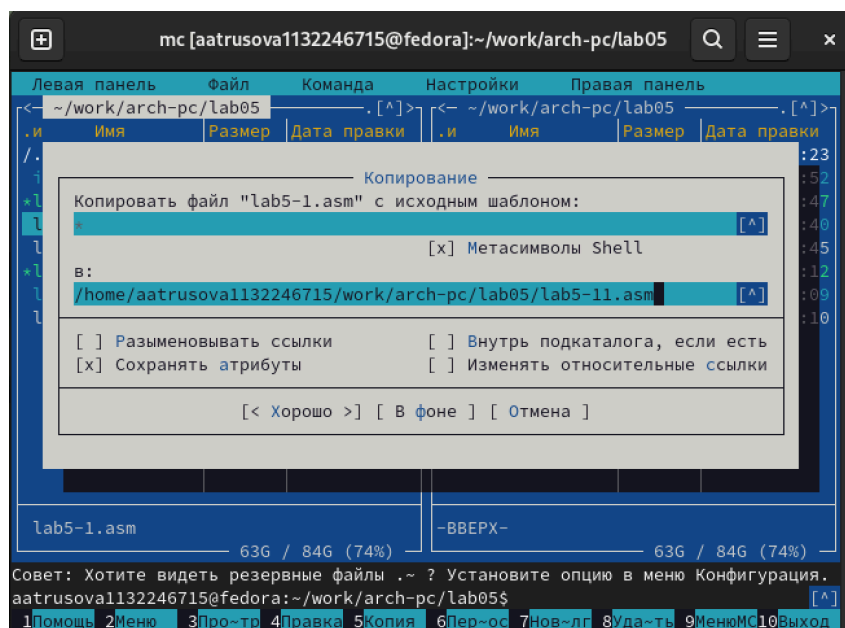


Рис.15

Внесла изменения в программу так, чтобы она работала по следующему алгоритму: 1) вывести приглашение типа "Введите строку:"; 2) ввести строку с клавиатуры; 3) вывести введенную строку на экран (рис.16).

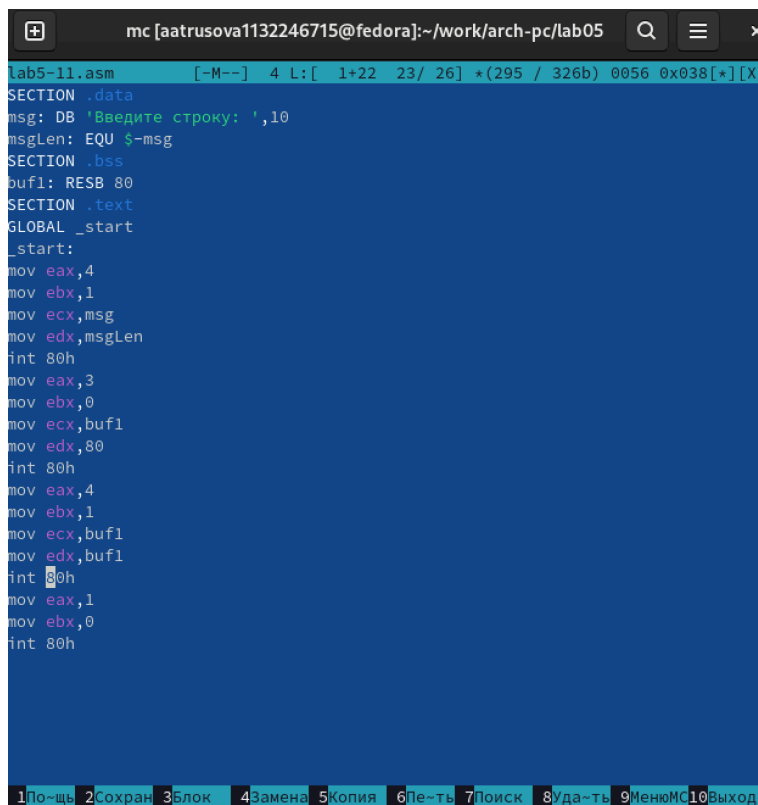


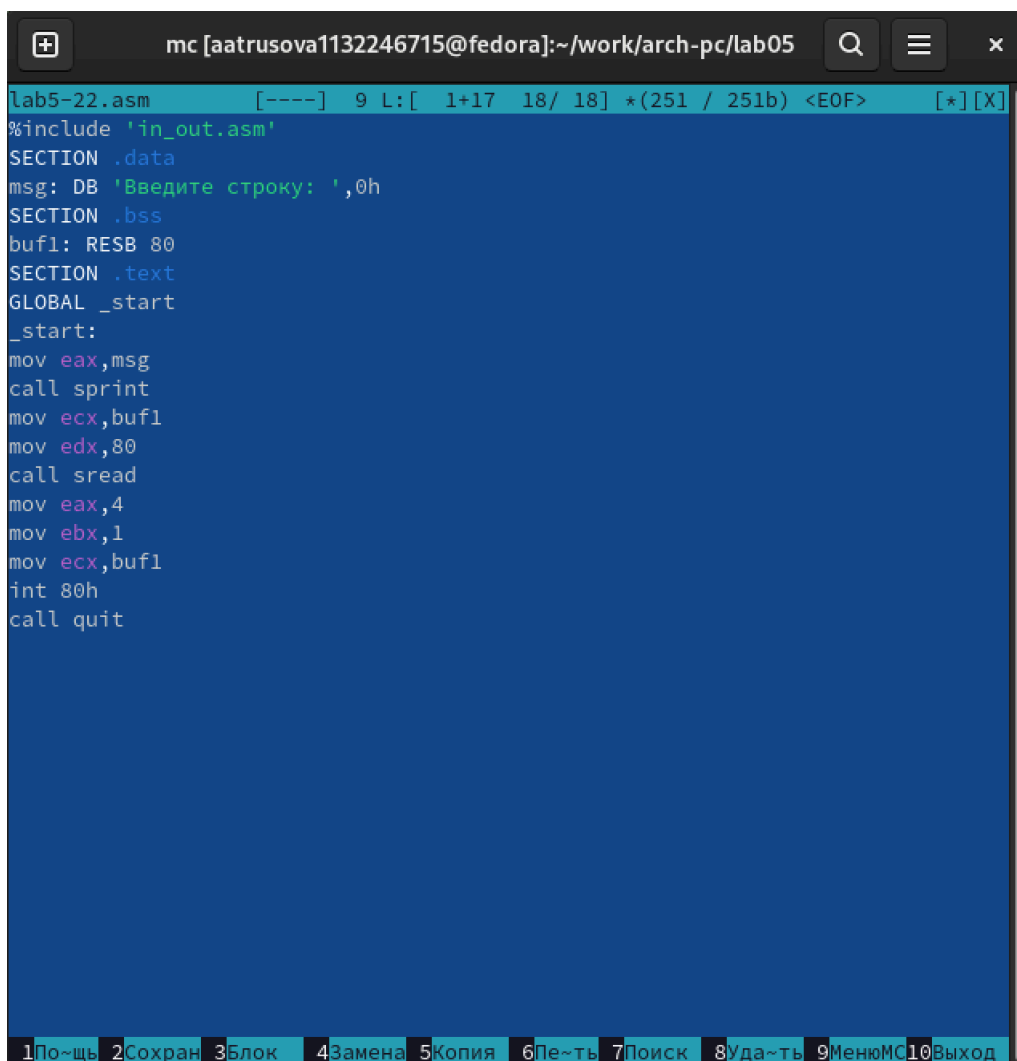
Рис.16

Получила исполняемый файл и проверила его работу. На приглашение ввести строку ввела свою фамилию (рис.17).

```
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-11.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-11.o -o lab5-11
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ./lab5-11
Введите строку:
Trusova
Trusova
aatrusova1132246715@fedora:~/work/arch-pc/lab05$
```

Рис.17

Создала копию файла lab5-2.asm. Исправила текст программы с использованием подпрограмм из внешнего файла in_out.asm так, чтобы она работала по следующему алгоритму: 1) вывести приглашение типа “Введите строку:”; 2) ввести строку с клавиатуры; 3) вывести введенную строку на экран (рис.18).



```
lab5-22.asm [-----] 9 L: [ 1+17 18/ 18] *(251 / 251b) <EOF> [*] [X]
#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,4
mov ebx,1
mov ecx,buf1
int 80h
call quit
```

1По~щъ 2Сохран 3Блок 4Замена 5Копия 6Пе~ть 7Поиск 8Уда~ть 9МенюМС10Выход

Рис.18

Создала исполняемый файл и проверила его работу (рис.19).

```
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-22.o -o lab5-22
aatrusova1132246715@fedora:~/work/arch-pc/lab05$ ./lab5-22
Введите строку: Trusova
Trusova
aatrusova1132246715@fedora:~/work/arch-pc/lab05$
```

Рис.19

Вывод

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы

1. [Лабораторная работа №6](#)