

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Студент: Лебеденко Елена Викторовна

Группа: НКАбд-04-25

МОСКВА

2025 г.

# Содержание

1. Цель работы.....	2
2. Выполнение лабораторной работы.....	4-10
2.1 Подключение внешнего файла in_out.asm.....	7
3. Задания для самостоятельной работы.....	11-15
4. Вывод.....	16

## **1. Цель работы**

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

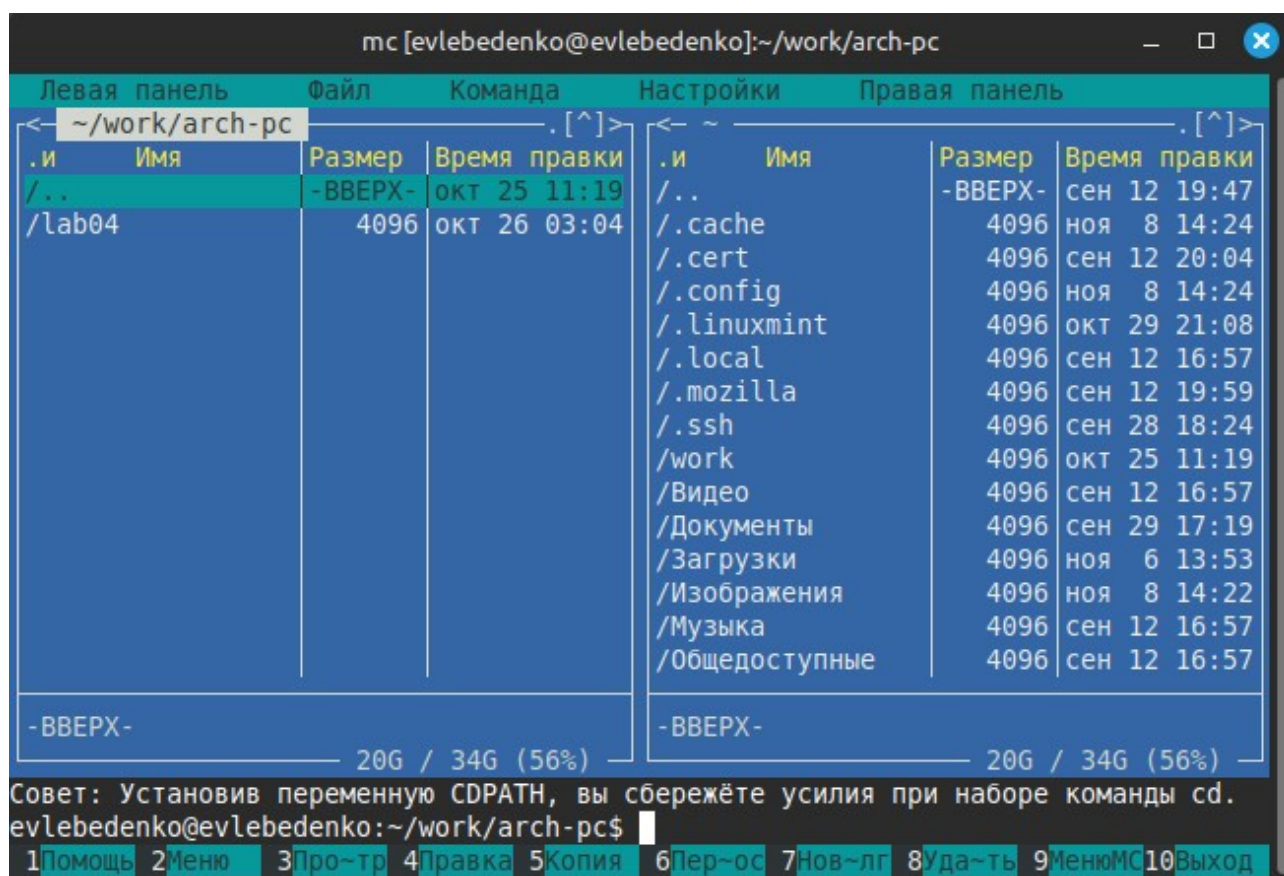
## 2. Выполнение лабораторной работы

1) Открываю Midnight Commander.

```
evlebedenko@evlebedenko:~$ mc
```

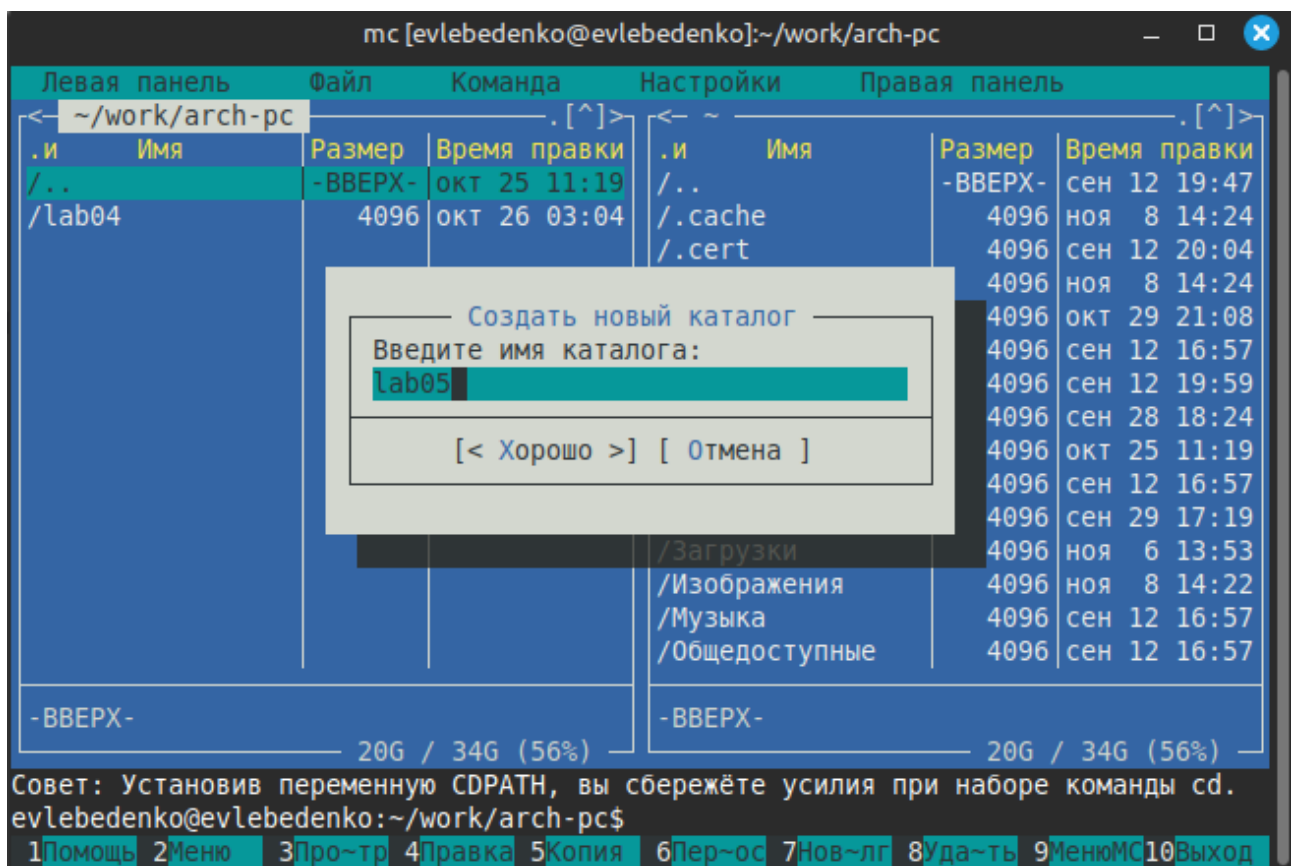
### 1.1 Запуск Midnight Commander

2) Пользуясь клавишами ↑, ↓ и Enter перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4



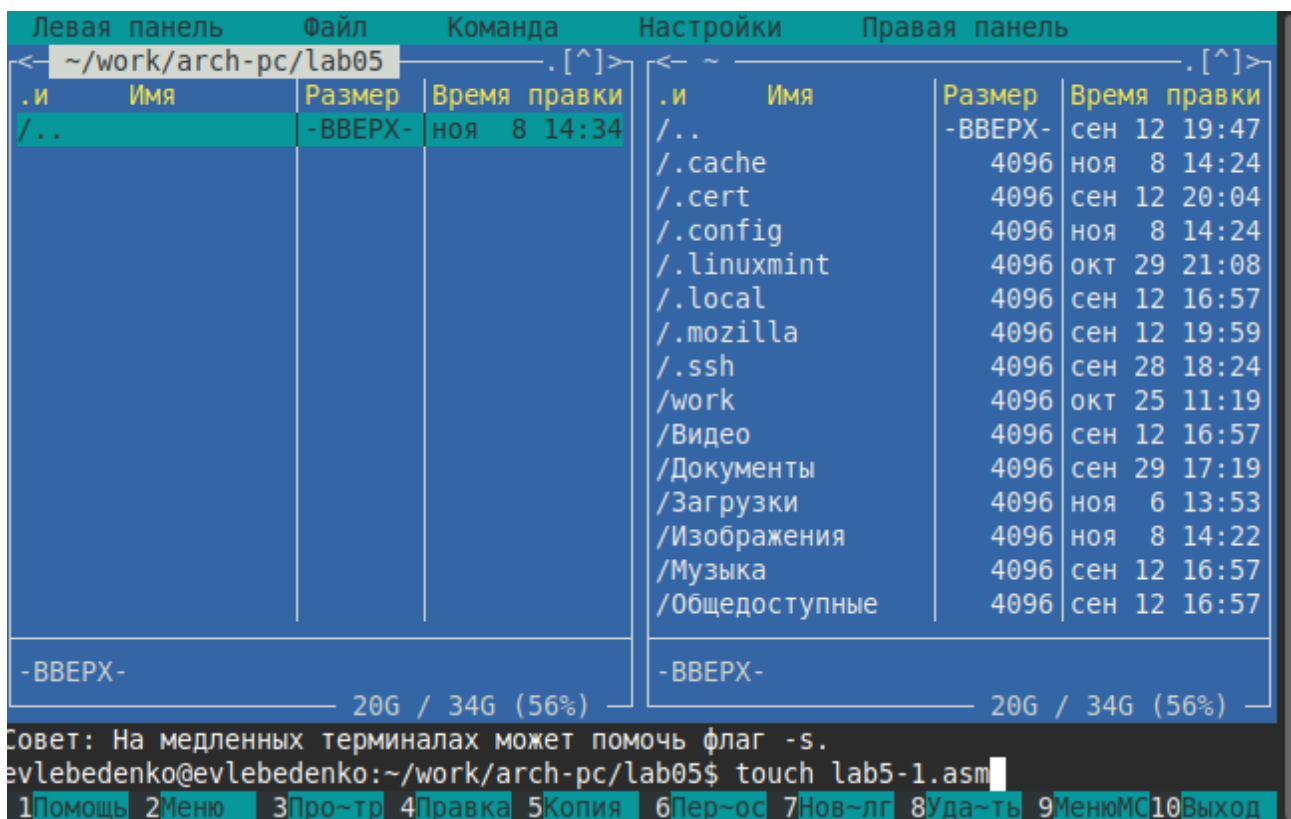
### 1.2 Переход в каталог ~/work/arch-pc

3) С помощью функциональной клавиши F7 создаю папку lab05 и перехожу в созданный каталог.



1.3 Создание папки lab05

4) Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm



1.4 Создание файла lab5-1.asm

5) С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе. Использую редактор nano.

```
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ touch lab5-1.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]: 1
```

1.5 Открытие файла с помощью редактора nano

6) Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл.

```
mc [evlebedenko@evlebedenko]:~/work/arch-pc/lab05
GNU nano 7.2 /home/evlebedenko/work/arch-pc/lab05/lab5-1.asm *
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4      ; Системный вызов для записи (sys_write)
mov ebx,1      ; Описатель файла 1 - стандартный вывод
mov ecx,msg    ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h        ; Вызов ядра
;----- системный вызов `read` -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3      ; Системный вызов для чтения (sys_read)
mov ebx,0      ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1   ; Адрес буфера под вводимую строку
mov edx,80     ; Длина вводимой строки
int 80h        ; Вызов ядра
;----- Системный вызов `exit` -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1      ; Системный вызов для выхода (sys_exit)
mov ebx,0      ; Выход с кодом возврата 0 (без ошибок)
int 80h        ; Вызов ядра

^G Справка      ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция
^X Выход        ^R ЧитФайл    ^\ Замена     ^U Вставить   ^J Выводить   ^_ К строке
```

1.6 Ввод текста программы

С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра. Убеждаюсь, что файл содержит текст программы.

```
mc [evlebedenko@evlebedenko]:~/work/arch-pc/lab05
/home/evlebedenko/work/~ch-pc/lab05/lab5-1.asm 1549/2585 59%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

1.7 Файл в режиме просмотра

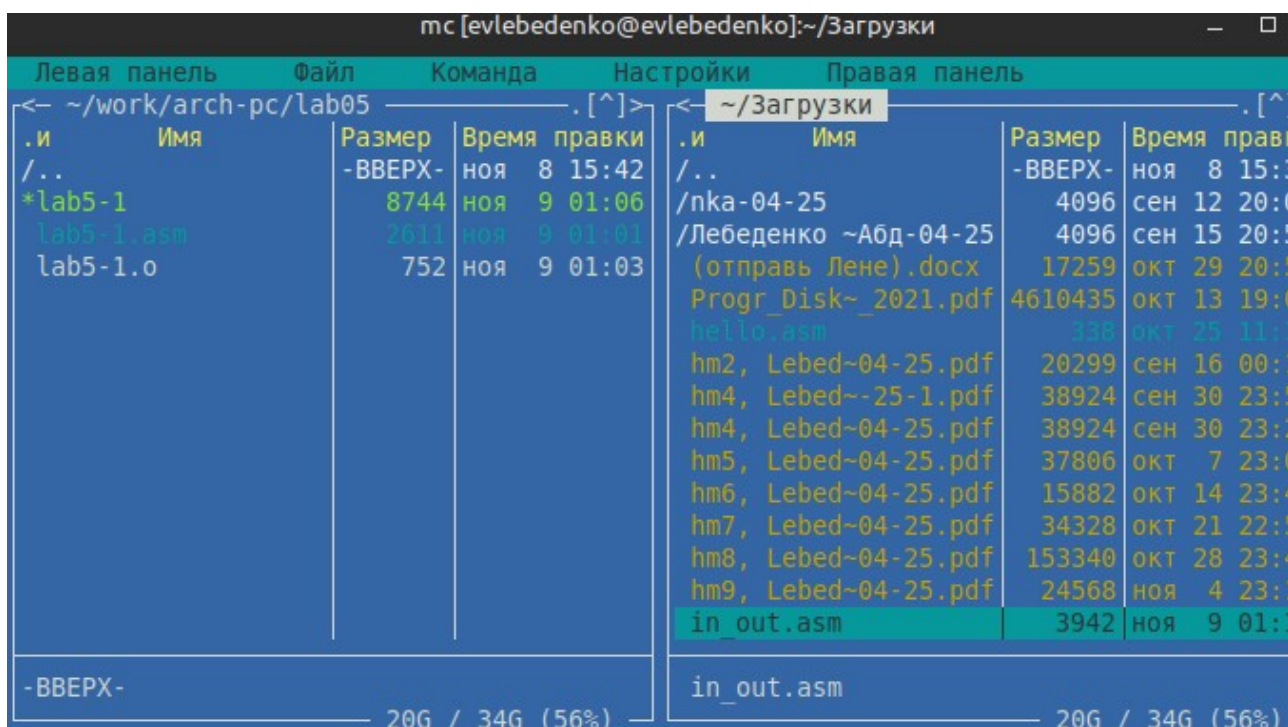
8) Транслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. Ввожу своё ФИО.

```
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Лебеденко Елена
```

1.8 Запуск программы

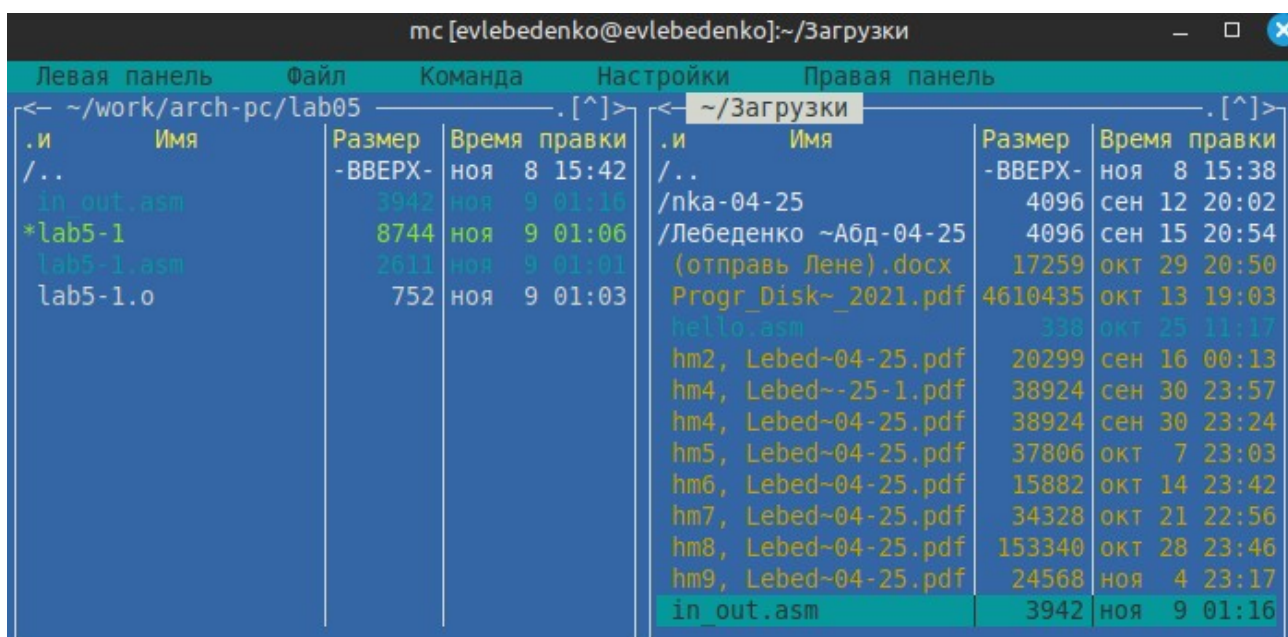
## 2.1 Подключение внешнего файла in\_out.asm

9) Скачиваю файл in\_out.asm со страницы курса в ТУИС.



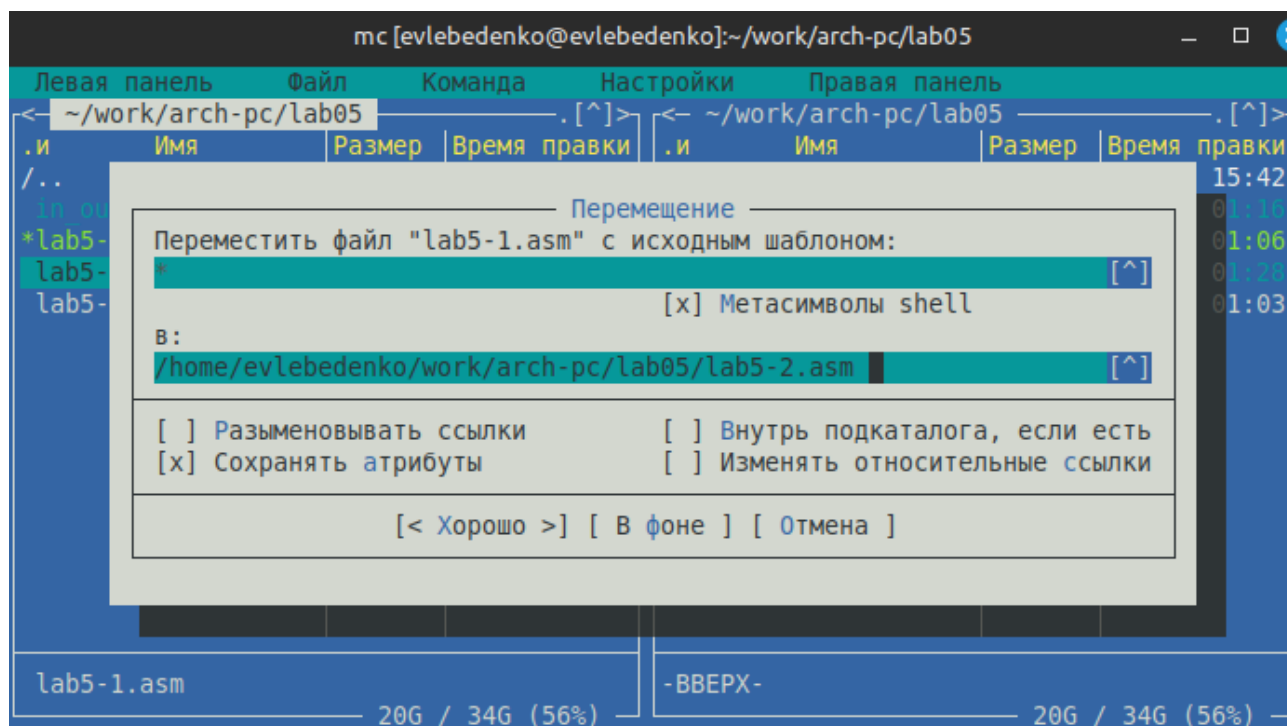
1.9 Скачивание файла in\_out.asm

10) Переношу файл в нужный каталог. В одной из панелей mc открываю каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm. Копирую файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5



1.10 Пренос файла

11) С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Выделяю файл lab5-1.asm, нажимаю клавишу F6, ввожу имя файла lab5-2.asm и нажимаю клавишу Enter



1.11 Создание файла lab5-2.asm

12) Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2. Создаю исполняемый файл и проверяю его работу.

```
mc [evlebedenko@evlebedenko]:~/work/arch-pc/lab05
GNU nano 7.2 /home/evlebedenko/work/arch-pc/lab05/lab5-2.asm
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

[ Прочитано 17 строк ]
```

1.12 Исправление файла

```
evlebedenko@evlebedenko: ~/work/arch-pc/lab05
evlebedenko@evlebedenko:~$ mc
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Лебеденко Елена
```

1.13 Запуск файла

14) В файле lab5-2.asm заменяю подпрограмму sprintf на printf. Создаю исполняемый файл и проверяю его работу.

В исправленном файле ввод строки начинается с той же строчки.

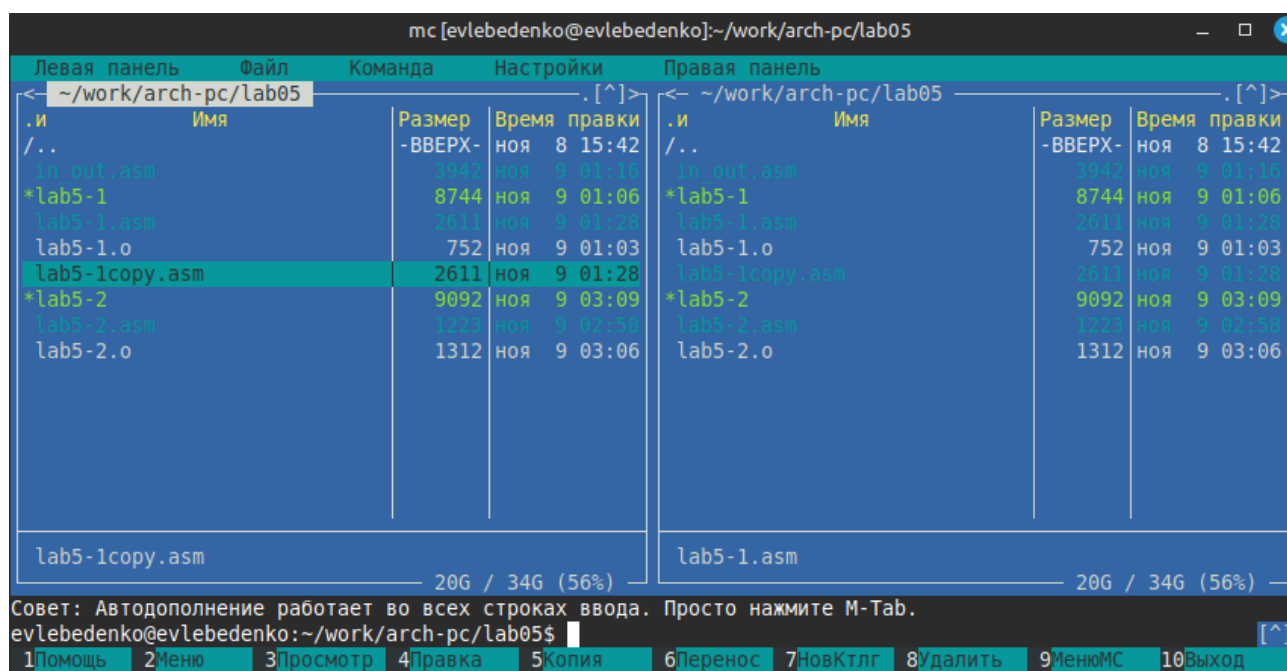
```
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Лебеденко Елена
```

1.14 Изменённый файл

### 3. Задание для самостоятельной работы

1) Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.



#### 2.1 Задание 1(1)

Копирую файл.

```
mc [evlebedenko@evlebedenko]:~/work/arch
GNU nano 7.2 /home/evlebedenko/work/arch-pc/lab05/
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, buf1
    int 80h
    mov eax,1
```

## 2.2 Задание 1(2)

Меняю программу

Листинг:

```
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, buf1
    int     80h
    mov     eax, 1
    mov     ebx, 0
    int     80h

```

2) Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

```

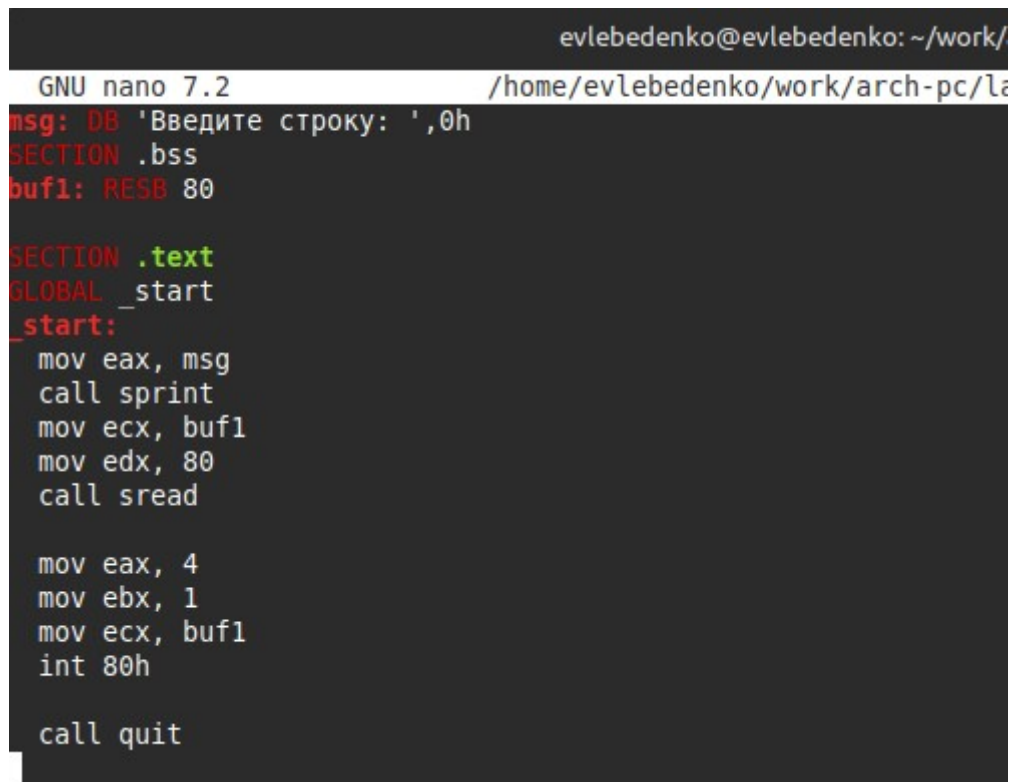
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1copy lab5-1copy.o
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
Лебеде́нко Елена
Лебеде́нко Елена

```

## 2.3 Задание 2

3) Создайте копию файла lab5-2.asm. Исправьте текст программы с использованием подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.



```
evlebedenko@evlebedenko: ~/work/
GNU nano 7.2 /home/evlebedenko/work/arch-pc/lab5-2.asm
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
```

2.4 Задание 3

Листинг:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg: DB 'Введите строку: ', 0h
```

```
msgLen: EQU $-msg
```

```
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

4) Создайте исполняемый файл и проверьте его работу.

```
nasm -f elf lab5-2copy.asmwork/arch-pc/lab05$  
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2copy lab5-2copy.o  
evlebedenko@evlebedenko:~/work/arch-pc/lab05$ ./lab5-2copy  
Введите строку: Лебеденко Елена  
Лебеденко Елена
```

2.5 Задание 4

## **4. Вывод**

При выполнении лабораторной работы я освоила базовые навыки работы в Midnight Commander.