

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: Безходарнова А.В

Группа: НКАбд-01-25

МОСКВА

2025 г.

СОДЕРЖАНИЕ

1. Цель работы	3
2. Выполнение лабораторной работы.....	4
2.1 Основы работы с Midnight Commander.....	4
2.2 Работа в NASM	6
2.3 Подключение внешнего файла.....	7
3. Задание для самостоятельной работы.....	10
4. Выводы	12

1. Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

2. Выполнение лабораторной работы

2.1 Основы работы с Midnight Commander

Введя соответствующую команду в терминале (рис. 2.1.1), я открываю Midnight Commander.(рис.2.1.2)

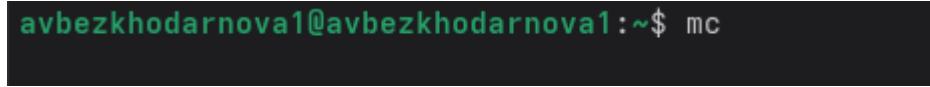


Рис. 2.1.1. Открытие МС

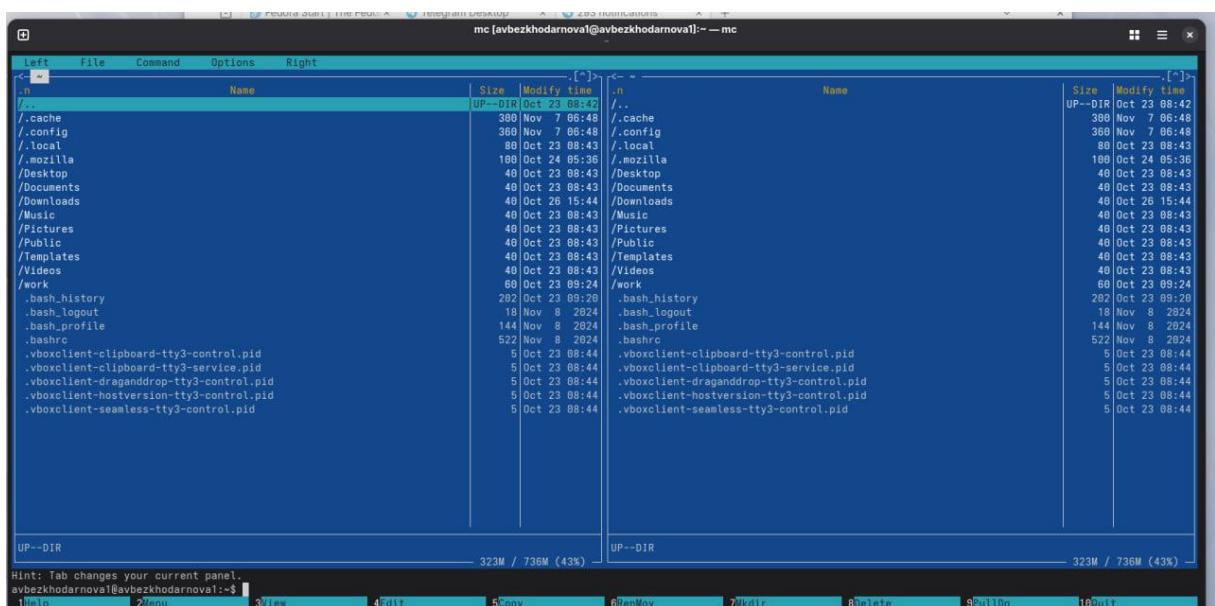


Рис. 2.1.2 Интерфейс Midnight Commander.

Открываю каталог, созданный в предыдущей работе и создаю подкаталог для дальнейшей работы. (рис.2.1.3)

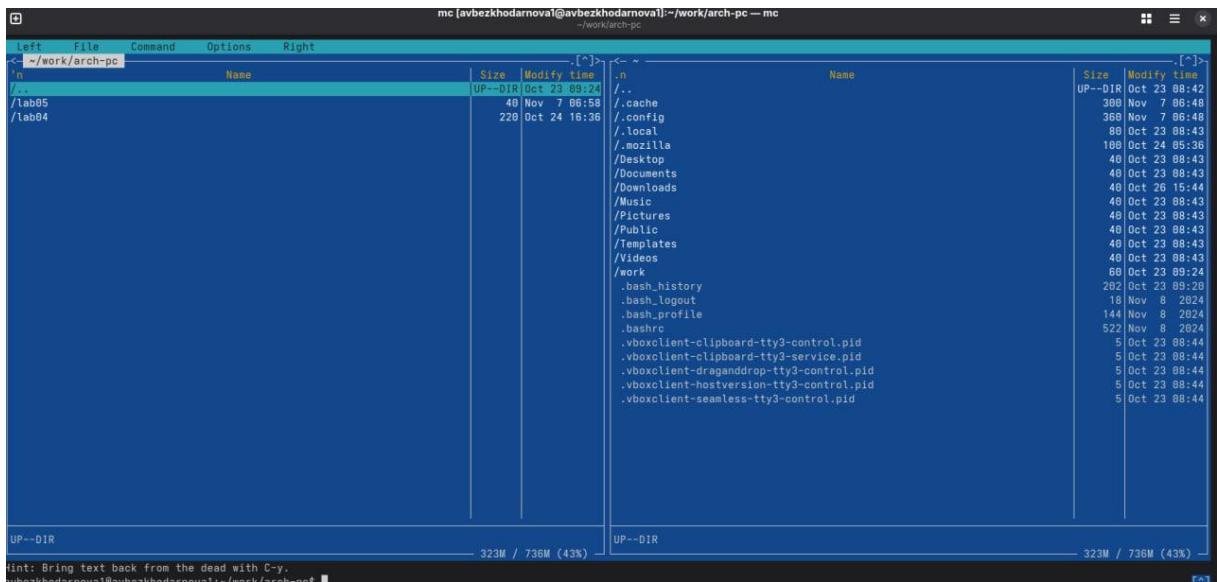


Рис. 2.1.3 Создание каталога

Далее с помощью команды touch в терминале создаю файл. (рис.2.1.4)

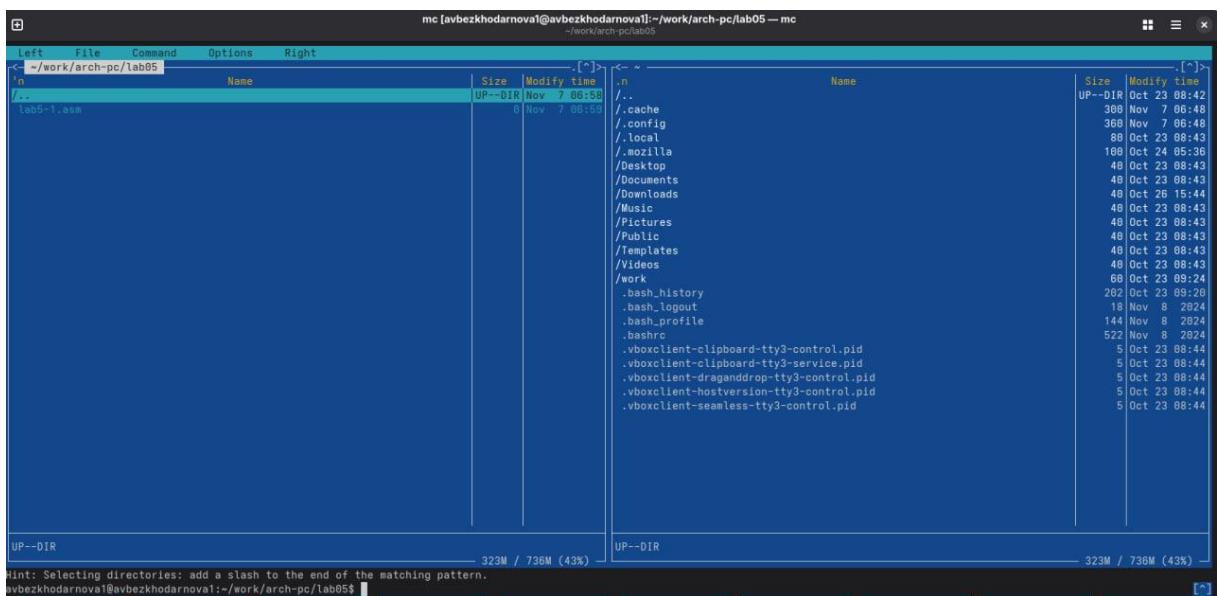
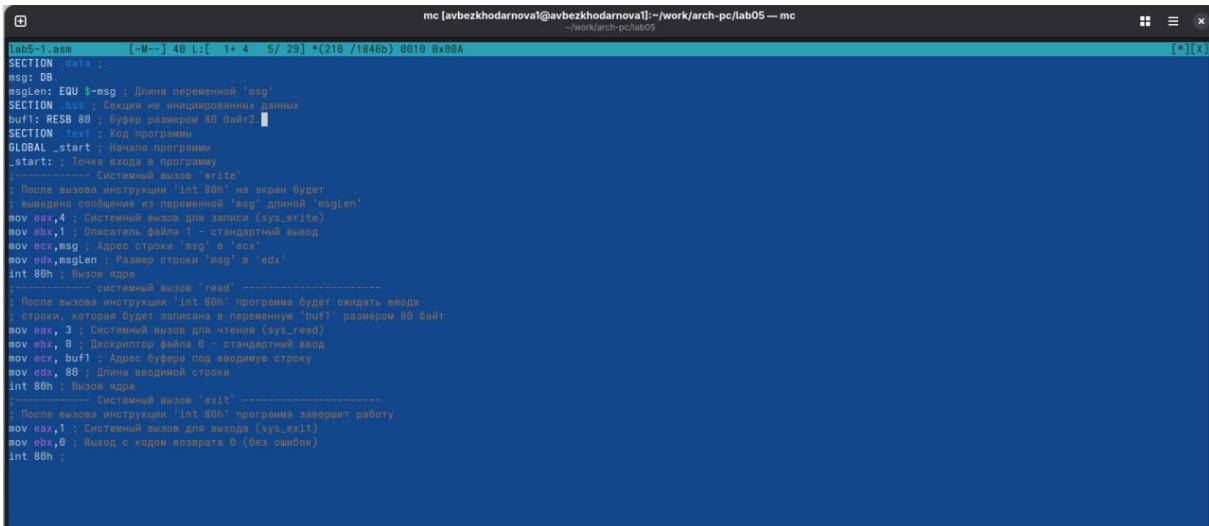


Рис. 2.1.4 Создание файла

2.2 Работа в NASM

С помощью F4 открываю файл и ввожу код. (рис. 2.2.1)

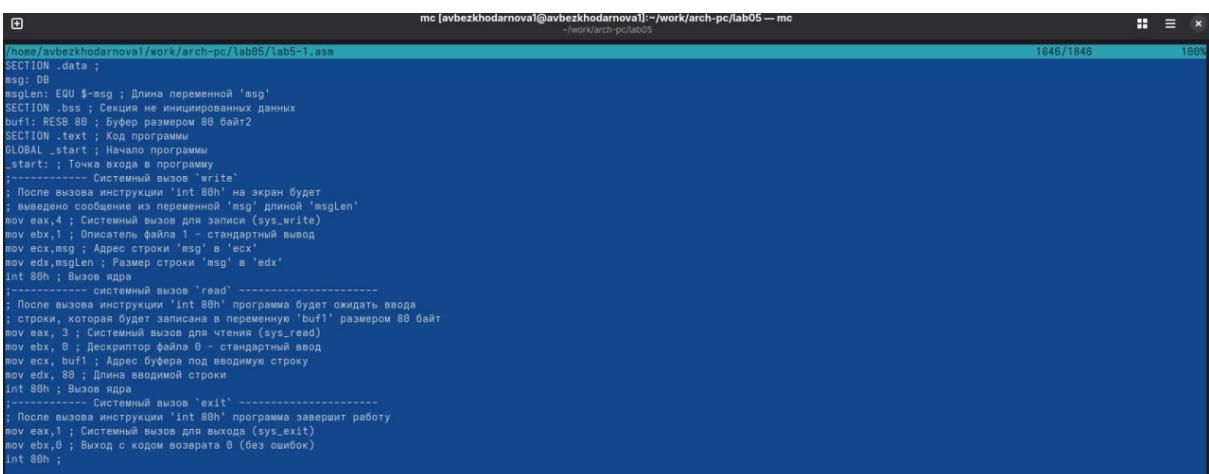


The screenshot shows the assembly code for the program. It includes sections for .data, .bss, and .text. The .text section contains instructions for system calls: write, read, and exit. The code is annotated with comments explaining the purpose of each instruction.

```
lab5-1.asm
[ -M-- ] 48 L:[ 1+ 4 5/ 29 ] *(216 /1846b) 0010 0x00A
mc [avbezkhodarnova@avbezkhodarnova:~/work/arch-pc/lab05] --> /work/arch-pc/lab05 — mc
SECTION .data :
msg: DB
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf: RESB 80 ; Буфер размером 80 байт, ■
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
    ;----- 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
```

Рис.2.2.1 Ввод кода.

Сохраняю файл и проверяю, что все сработало. (Рис.2.2.2)



The screenshot shows the same assembly code as in Figure 2.2.1, but it appears to be a copy of the original file. The code is identical to the one shown in Figure 2.2.1.

```
mc [avbezkhodarnova@avbezkhodarnova:~/work/arch-pc/lab05] --> /work/arch-pc/lab05 — mc
/home/avbezkhodarnova/work/arch-pc/lab05/lab5-1.asm
1846/1846 100%
SECTION .data :
msg: DB
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf: RESB 80 ; Буфер размером 80 байт
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
    ;----- 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
```

Рис. 2.2.2 Сохранение кода

Далее запускаю файл и проверяю его работоспособность. (Рис.2.2.3)

```
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab-1.o
ld: cannot find lab-1.o: No such file or directory
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ./lab5-1
введите текст
Bezkhodarnova Alisa Viktorovna
```

Рис. 2.2.3 Запуск программы

2.3 Подключение внешнего файла

Скачав нужный файл, копирую исходный файл, чтобы продолжить работу. (рис.2.3.1) и (рис.2.3.2)

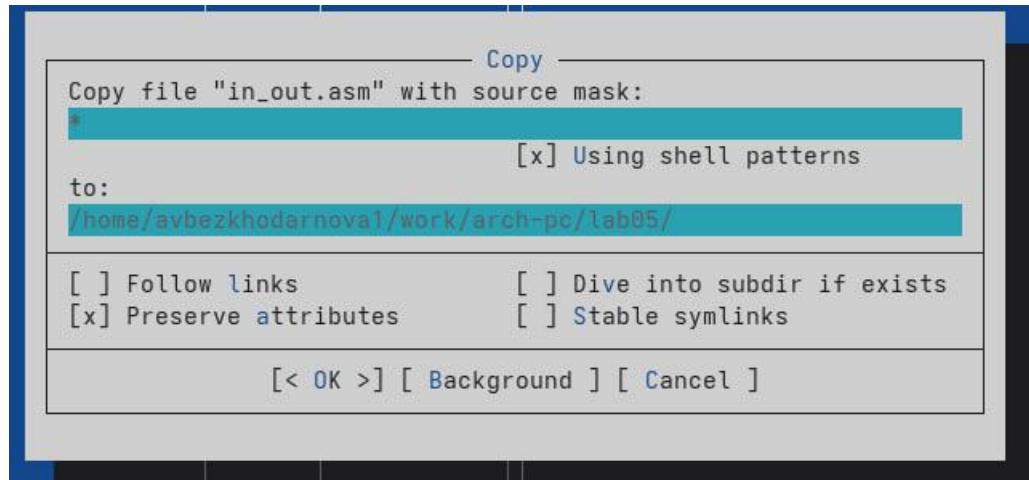


Рис 2.3.1 Копирование файла

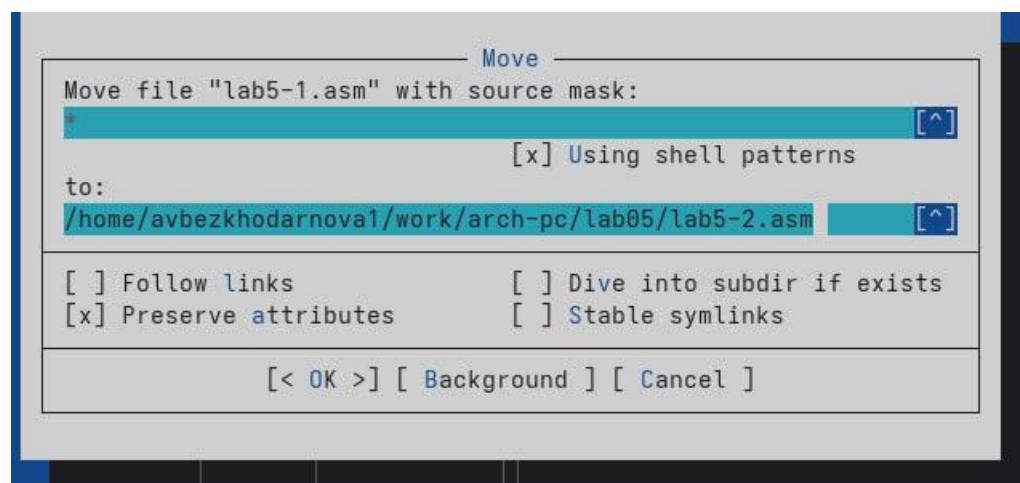


Рис. 2.3.2 Создание копии

Далее я подключаю созданную программу в исходный файл. (рис.2.3.3)

```

SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprintLF ; вызов подпрограммы печати сообщения
    mov ecx, buf1 ; запись адреса переменной в EAX
    mov edx, 80 ; запись длины выводимого сообщения в EBX
    call sread ; вызов подпрограммы ввода сообщения
    call quit

```

Рис. 2.3.3 Изменение программы.

Далее запускаю программу. (Рис.2.3.4)

```

avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Bezhkodarnova Alisa Viktorovna
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ █

```

Рис. 2.3.4 Запуск измененного кода

Заменяю в коде sprintLF на sprint и также запускаю программу. (Рис.2.3.5) и (рис.2.3.6)

```

%include 'in_out.asm'
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprint ; вызов подпрограммы печати сообщения
    mov ecx, buf1 ; запись адреса переменной в EAX
    mov edx, 80 ; запись длины выводимого сообщения в EBX
    call sread ; вызов подпрограммы ввода сообщения
    call quit

```

Рис.2.3.5 изменение кода

```

avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Bezhkodarnova Alisa Viktorovna
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ █

```

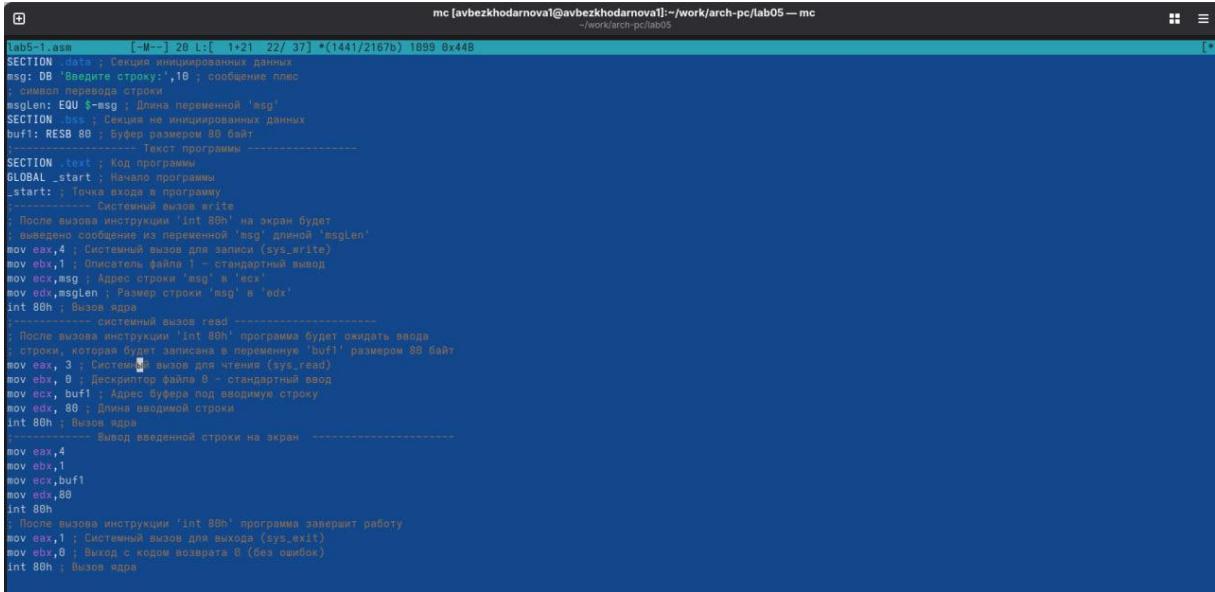
Рис.2.3.6 запуск программы

Разница заключается в том, что в первом коде программа выводит

данные на следующей строчке, а в изменненом коде вывод производится на текущую строку.

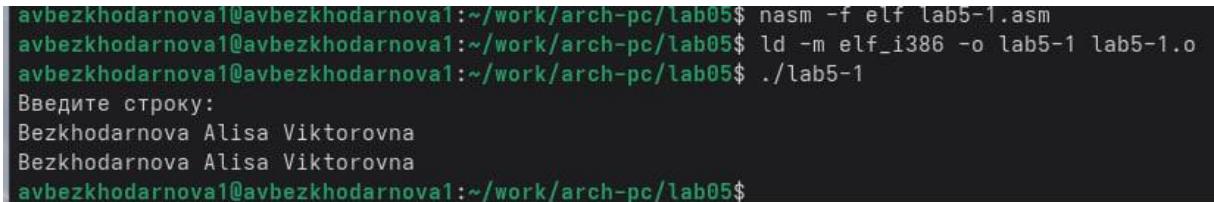
3. Задание для самостоятельной работы

Создаю копию файла изменяю код, чтобы в конце выводилась введенная мною строка с клавиатуры. (рис.3.1) и рис (3.2.)



```
mc [avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05] - mc  
Lab5-1.asm [-M--] 28 L:[ 1+21 22/ 37] *(1441/21676) 1099 0x44B  
SECTION .data ; Секция инициализированных данных  
msg: DB 'Введите строку:',10 ; сообщение плюс  
; символ перевода строки  
msgLen: EQU $-msg ; Длина переменной 'msg'  
SECTION .bss ; Секция на 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  
----- 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 ; Descriptor файла 0 – стандартный ввод  
mov ecx,buf1 ; Адрес буфера под вводимую строку  
mov edx,80 ; Длина вводимой строки  
int 80h ; Вызов ядра  
----- Вывод введенной строки на экран -----  
mov eax,4  
mov ebx,1  
mov ecx,buf1  
mov edx,80  
int 80h  
; 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
```

Рис.3.1 Редактирование файла.



```
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm  
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o  
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$ ./lab5-1  
Введите строку:  
Bezhodarnova Alisa Viktorovna  
Bezhodarnova Alisa Viktorovna  
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab05$
```

Рис.3.2 Запуск программы.

Также редактирую второй файл, предварительно создав его копию и запускаю программу. (рис.3.3) и (рис.3.4)

```
mc [avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05 — m
~/work/arch-pc/lab05

lab5-2(copy).asm  [-M--] 12 L:[ 1+16 17/ 17] *(936 / 936b) <EOF>
%include 'in_out.asm'; подключение внешнего файла.
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0 ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprint ; вызов подпрограммы печати сообщения
    mov ebx,0
    mov ecx, buf1 ; запись адреса переменной в EAX
    mov edx, 80 ; запись длины вводимого сообщения в EBX
    call sread ; вызов подпрограммы ввода сообщения
    mov eax, buf1
    call sprintLF
    call quit ; █
```

Рис.3.3 Редактирование файла.

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-2-2.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ./lab2-2
bash: ./lab2-2: No such file or directory
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку: Bezhodarnova Alisa Viktorovna
Bezhodarnova Alisa Viktorovna
```

Рис.3.4 запуск программы

4. Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.