

**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)

```
avbezhkodarnova1@avbezhkodarnova1:~$ mc
```

Рис. 2.1.1. Открытие MC

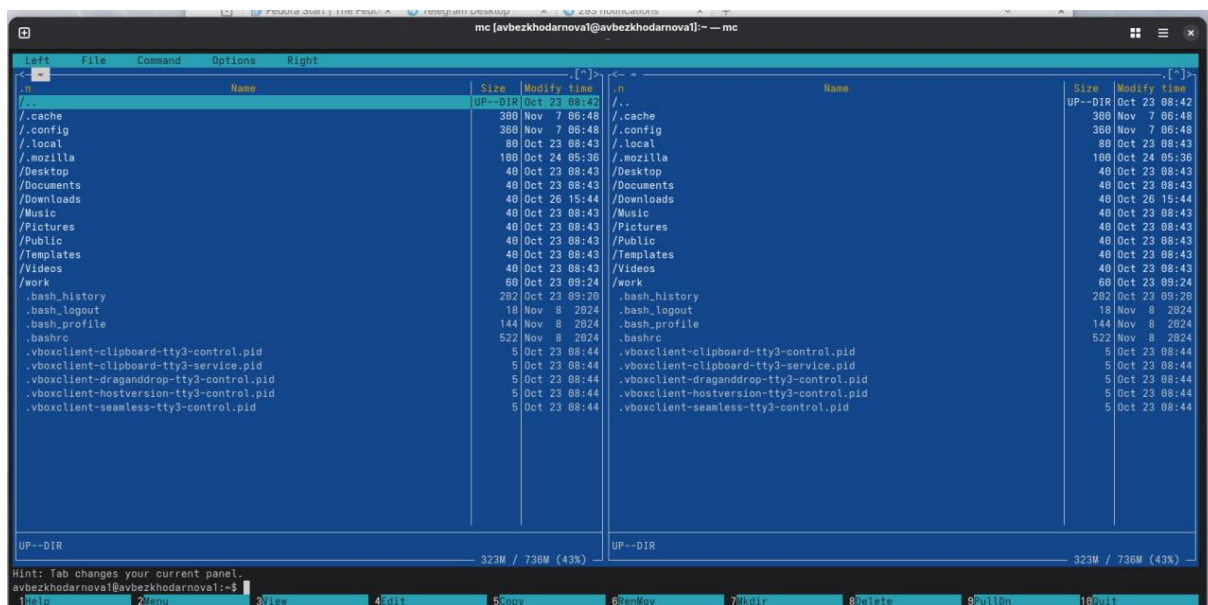


Рис. 2.1.2 Интерфейс Midnight Commander.

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

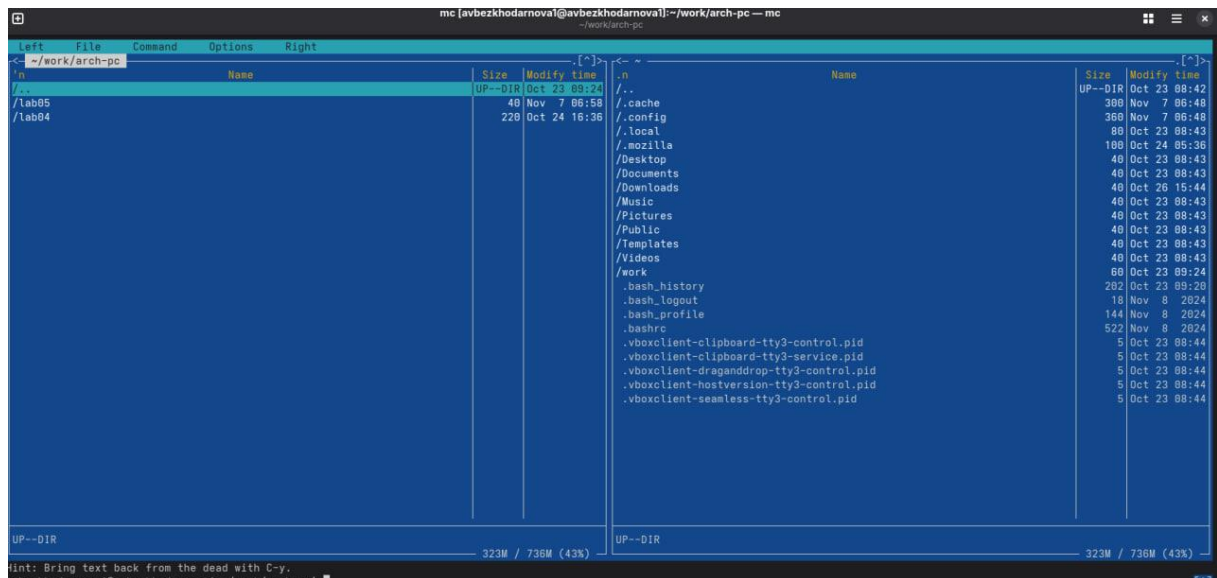


Рис. 2.1.3 Создание каталога

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

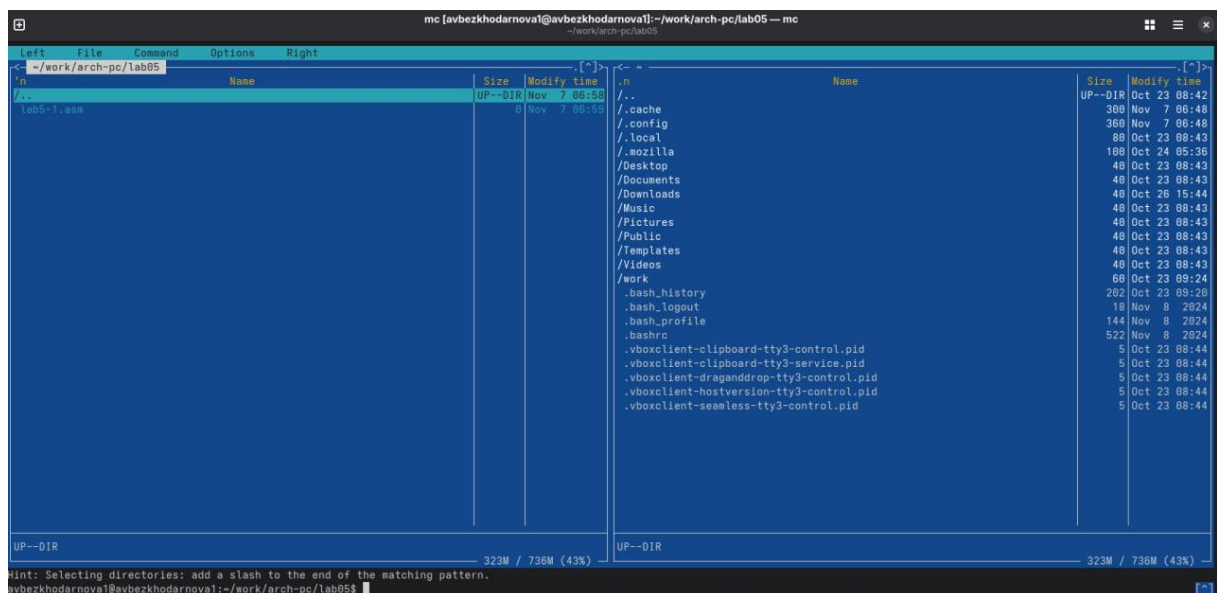
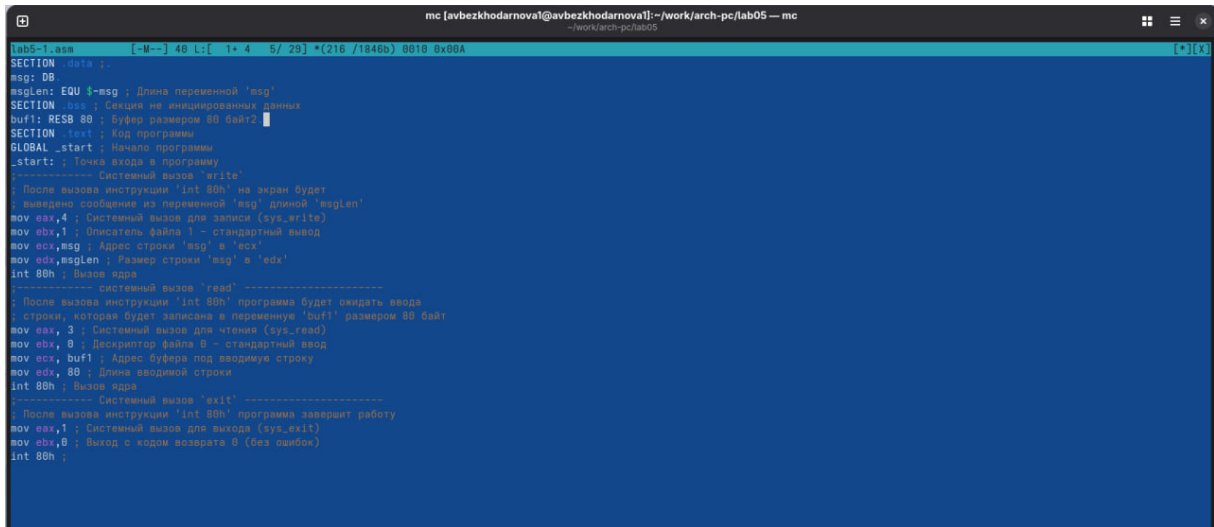


Рис. 2.1.4 Создание файла

## 2.2 Работа в NASM

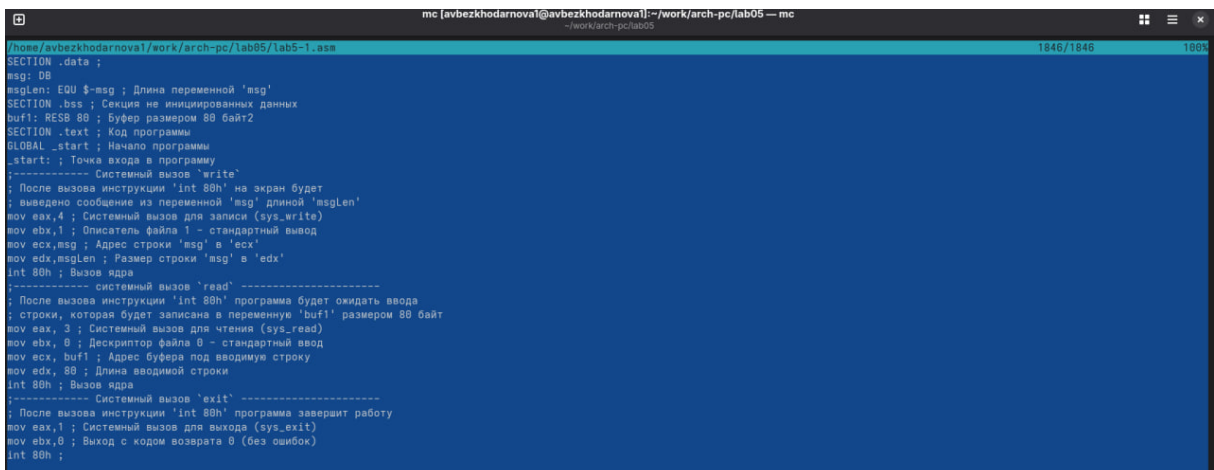
С помощью F4 открываю файл и ввожу код. (рис. 2.2.1)



```
lab5-1.asm [-M--] 40 L: 1+ 4 5/ 29 *(216 /1846b) 8010 8x80A
SECTION .data ;
msg: DB
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт2
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 ; Описание 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 ;
```

Рис.2.2.1 Ввод кода.

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



```
home/avbezhodarnova1/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
buf1: RESB 80 ; Буфер размером 80 байт2
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 ; Описание 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 ;
```

Рис. 2.2.2 Сохранение кода

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

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
avbezhodarnova1@avbezhodarnova1:~/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
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
avbezhodarnova1@avbezhodarnova1:~/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)

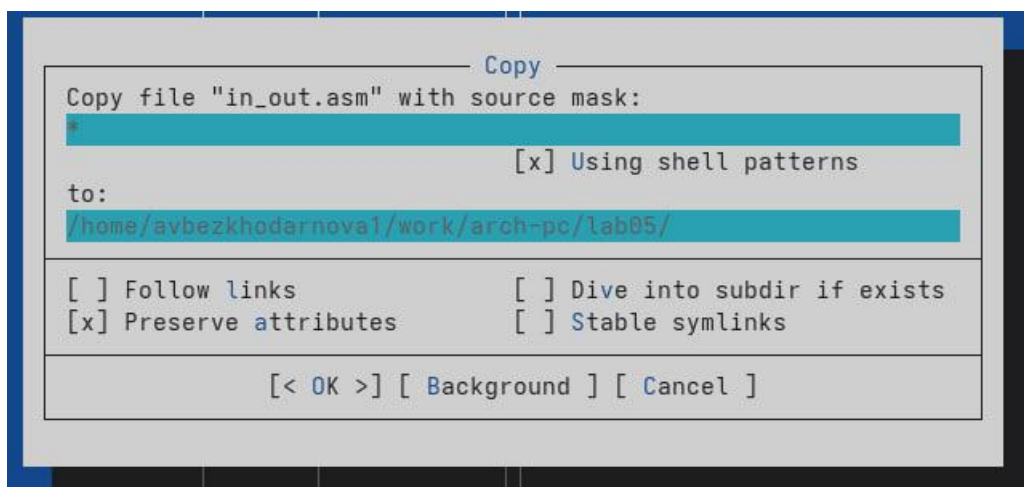


Рис 2.3.1 Копирование файла

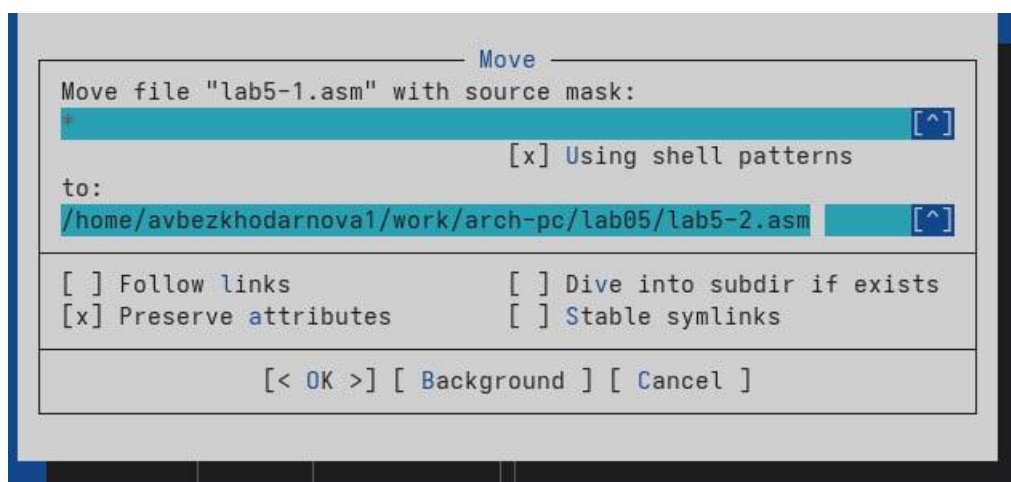


Рис. 2.3.2 Создание копии

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

```
SECTION .data ; Секция 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
```

Рис. 2.3.3 Изменение программы.

Далее запускаю программу. (Рис.2.3.4)

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Bezkhodarnova Alisa Viktorovna
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$
```

Рис. 2.3.4 Запуск измененного кода

Заменяю в коде sprintf на printf и также запускаю программу. (Рис.2.3.5) и (рис.2.3.6)

```
%include 'in_out.asm'
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 printf ; 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
```

Рис.2.3.5 изменение кода

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Bezkhodarnova Alisa Viktorovna
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab05$
```

Рис.2.3.6 запуск программы

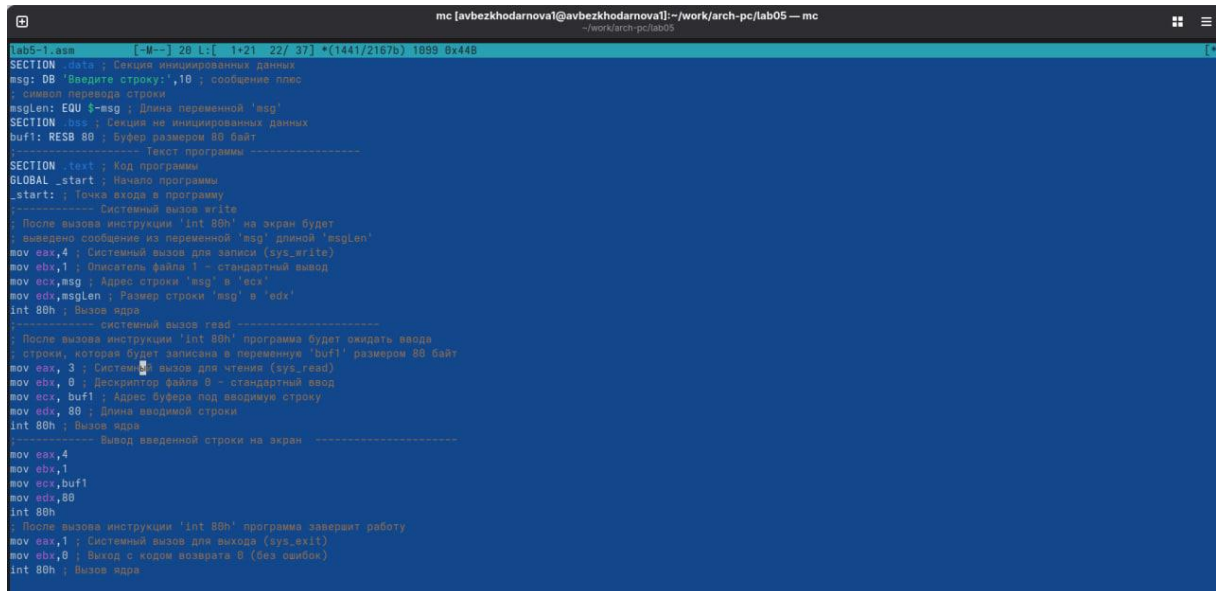
Разница заключается в том, что в первом коде программа выводит



данные на следующей строчке, а в изменненном 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.)



```
lab5-1.asm [-M--] 20 L: [ 1+21 22/ 37] *(1441/2167b) 1899 0x448
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 ; Файловый дескриптор 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
;----- 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:
Bezkhodarnova Alisa Viktorovna
Bezkhodarnova 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 иницированных данных
msg: DB 'Введите строку: ',0 ; сообщение
SECTION .bss ; Секция не иницированных 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`.