

# 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: Скворцова Дарья Дмитриевна

Группа: НКАбд-05-25

МОСКВА

2025 г.

## **5.1 Цель работы:**

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

### 5.3. Порядок выполнения лабораторной работы:

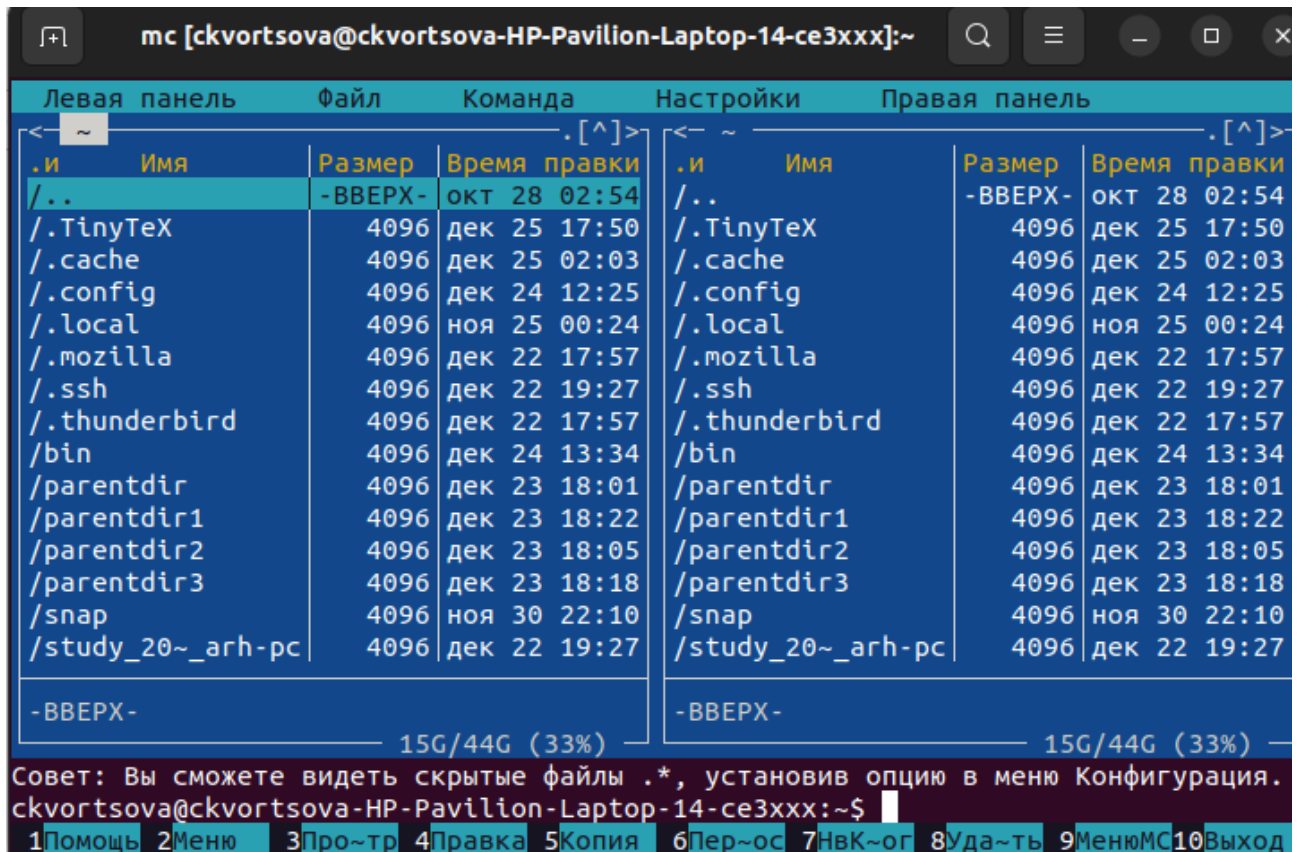
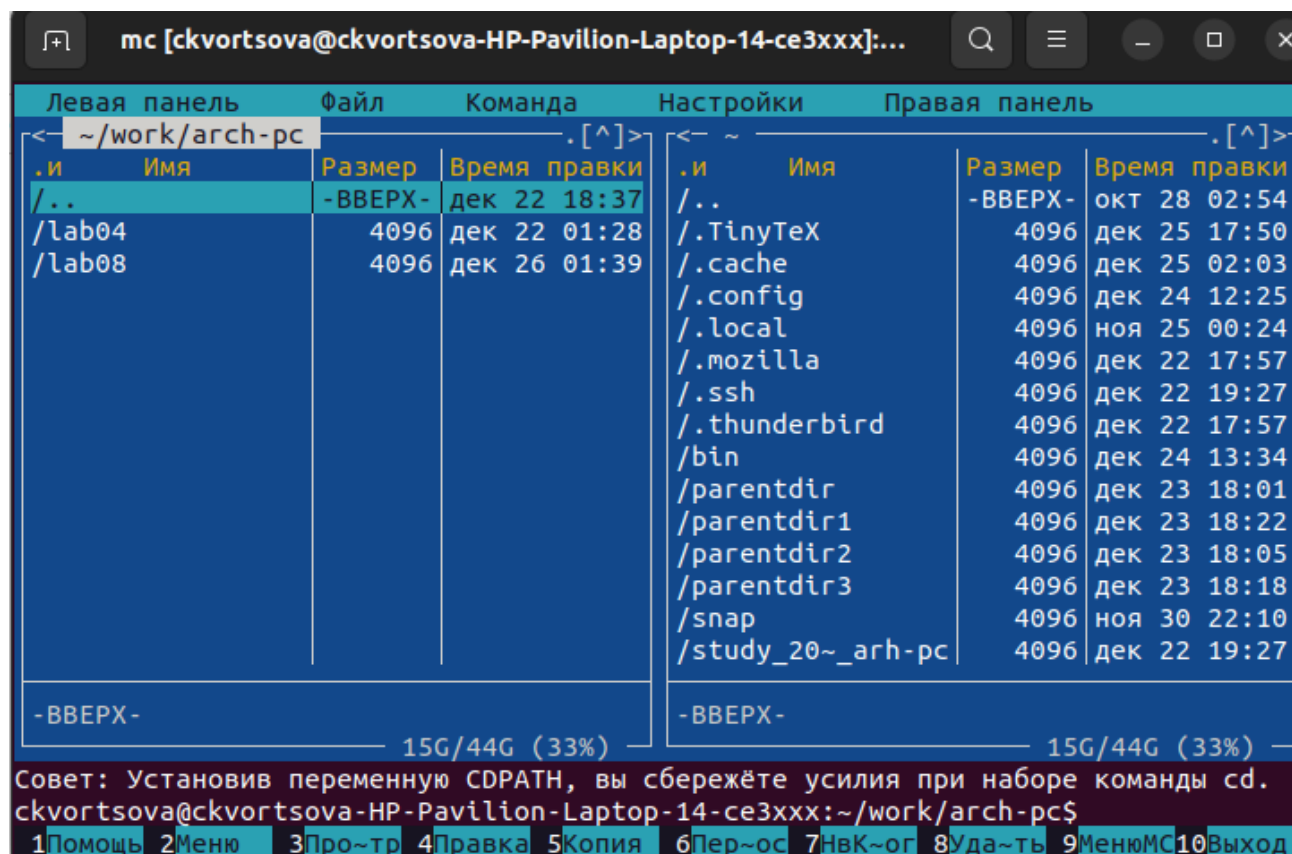


Рис. 1

1) Открыл Midnight Commander с помощью команды mc.(см Рис. 1)



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

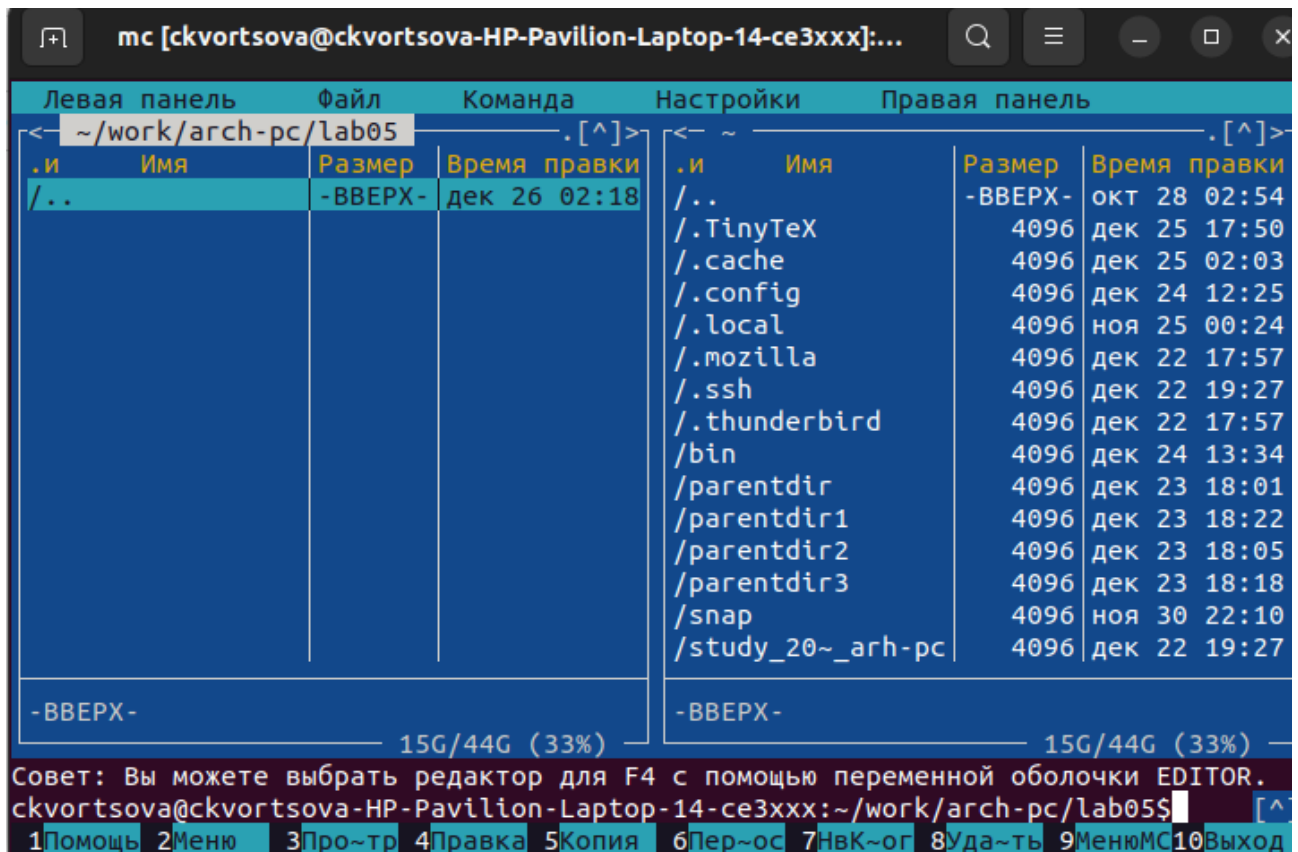


Рис. 3

3) С помощью функциональной клавиши F7 создал папку lab05 и перешёл в созданный каталог. (см Рис. 3)

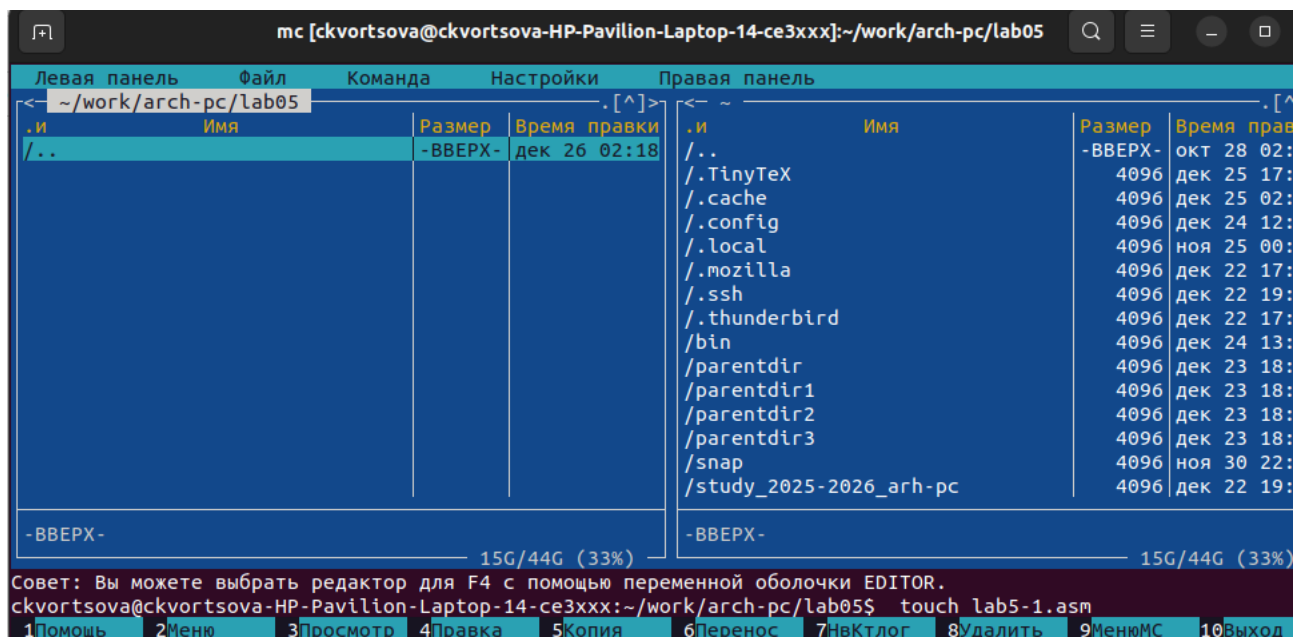


Рис. 4.1

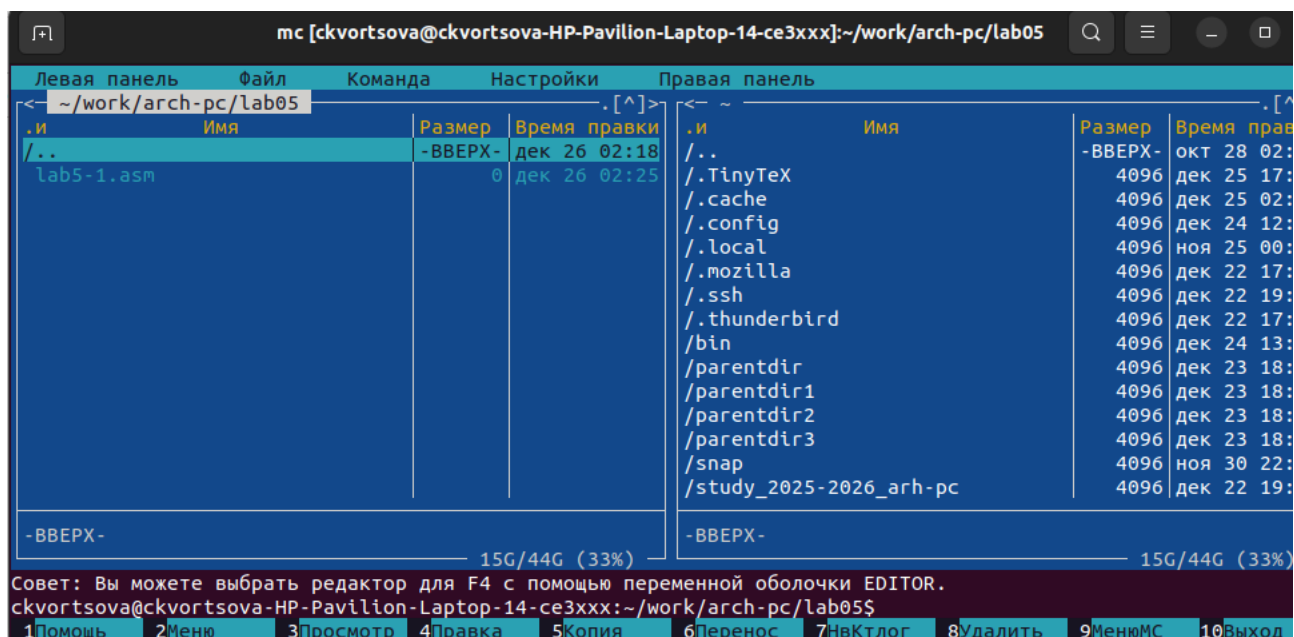


Рис. 4.2

4. Пользуясь строкой ввода и командой touch создал файл lab5-1.asm (см Рис. 4.1 и Рис. 4.2)

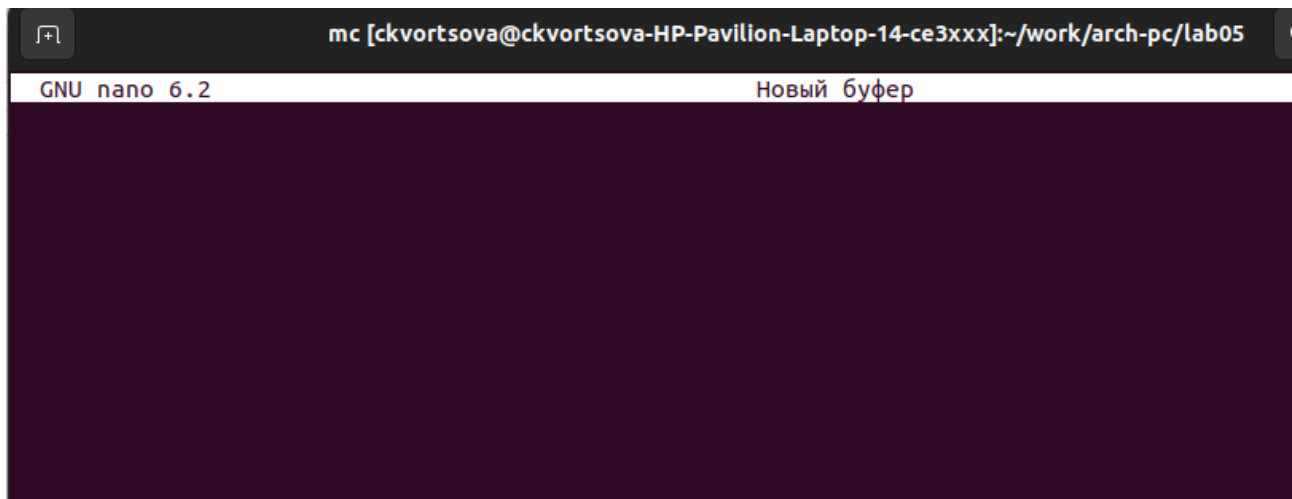
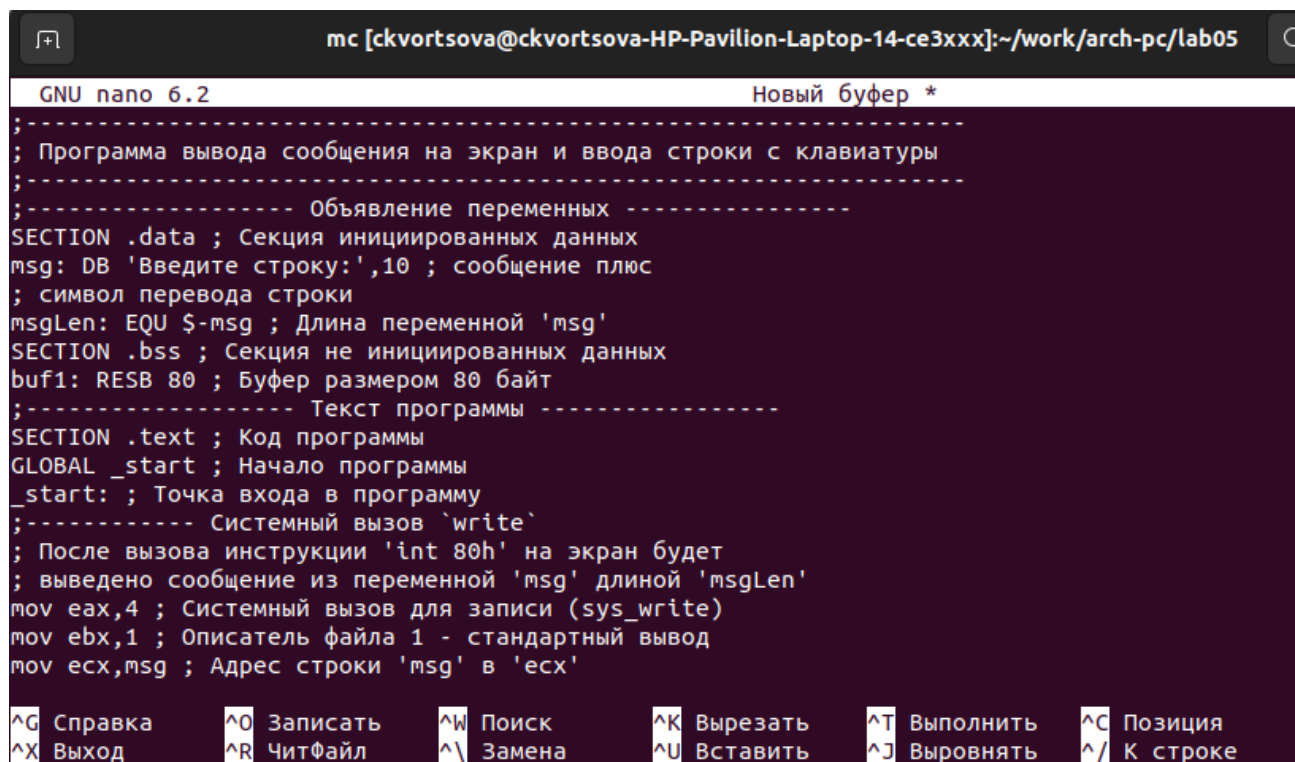


Рис. 5

5. С помощью функциональной клавиши F4 открыл файл lab5-1.asm для редактирования в редакторе nano. (см Рис. 5)

6) Ввёл текст программы из листинга 5.1, сохранил изменения (*Ctrl + x (выход) > Y (сохранить изменения) > Enter*) и закрыл файл.

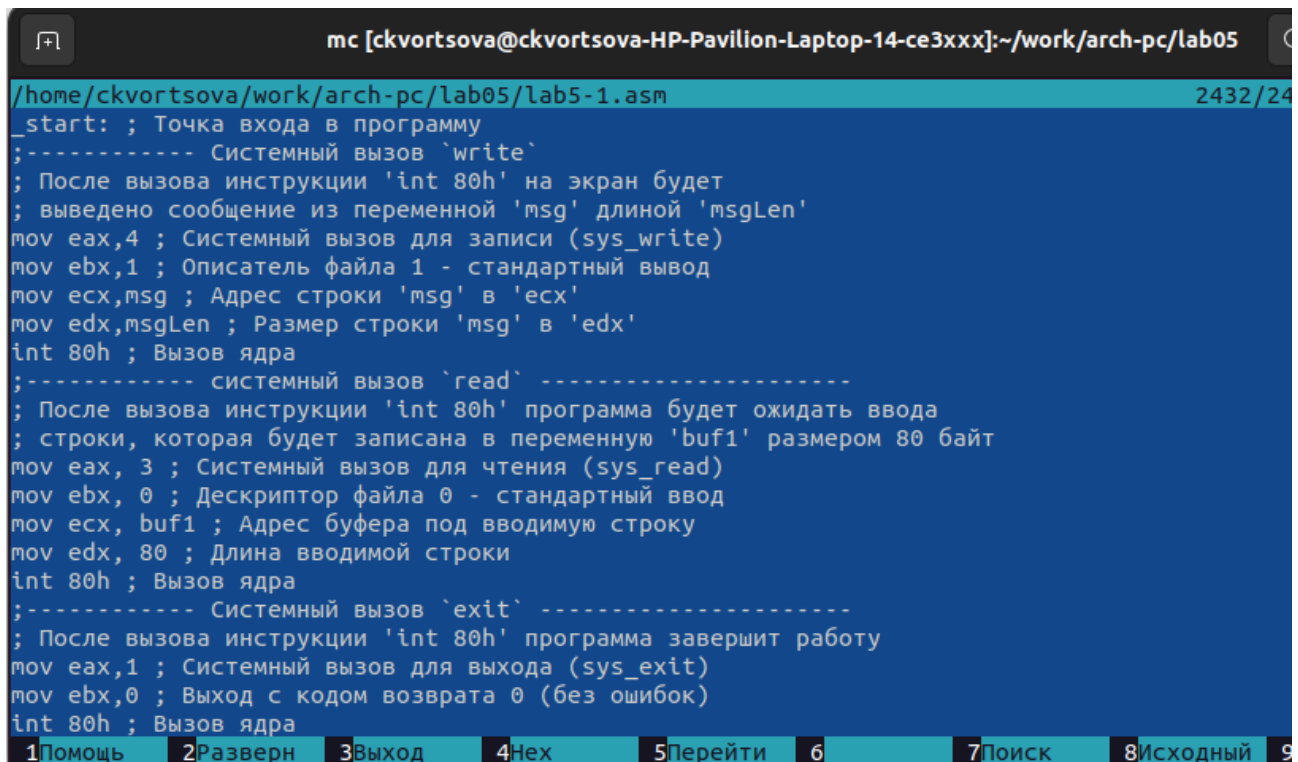


```
mc [ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx]:~/work/arch-pc/lab05
GNU nano 6.2                                     Новый буфер *
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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'
```

<b>^G</b> Справка	<b>^O</b> Записать	<b>^W</b> Поиск	<b>^K</b> Вырезать	<b>^T</b> Выполнить	<b>^C</b> Позиция
<b>^X</b> Выход	<b>^R</b> ЧитФайл	<b>^I</b> Замена	<b>^U</b> Вставить	<b>^J</b> Выводить	<b>^/_</b> К строке

Рис 6.1



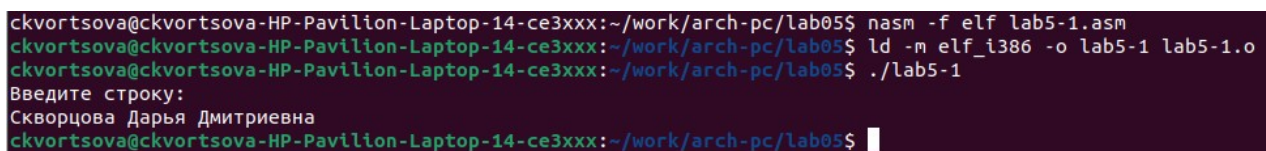


```
mc [ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx]:~/work/arch-pc/lab05
/home/ckvortsova/work/arch-pc/lab05/lab5-1.asm 2432/24
_start: ; Точка входа в программу
;----- Системный вызов `write`
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
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
1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9
```

Рис. 7

7) С помощью функциональной клавиши F3 открыл файл lab5-1.asm для просмотра.

Убедился, что файл содержит текст программы.



```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Скворцова Дарья Дмитриевна
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$
```

Рис. 8

8) 1. Скрыл панели Midnight Commander, с помощью комбинации Ctrl + o.

2. Оттранслировал текст программы lab5-1.asm в объектный файл.

3. Выполнил компоновку объектного файла и запустил получившийся исполняемый файл.

4. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос ввел свои ФИО.

(см Рис. 8)

### 5.3.1. Подключение внешнего файла in\_out.asm

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

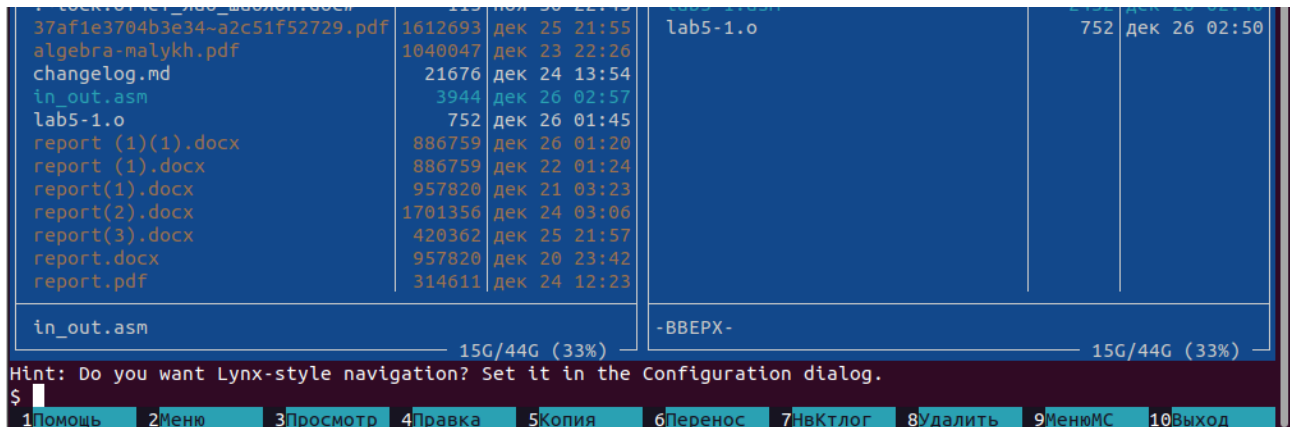


Рис. 10.1

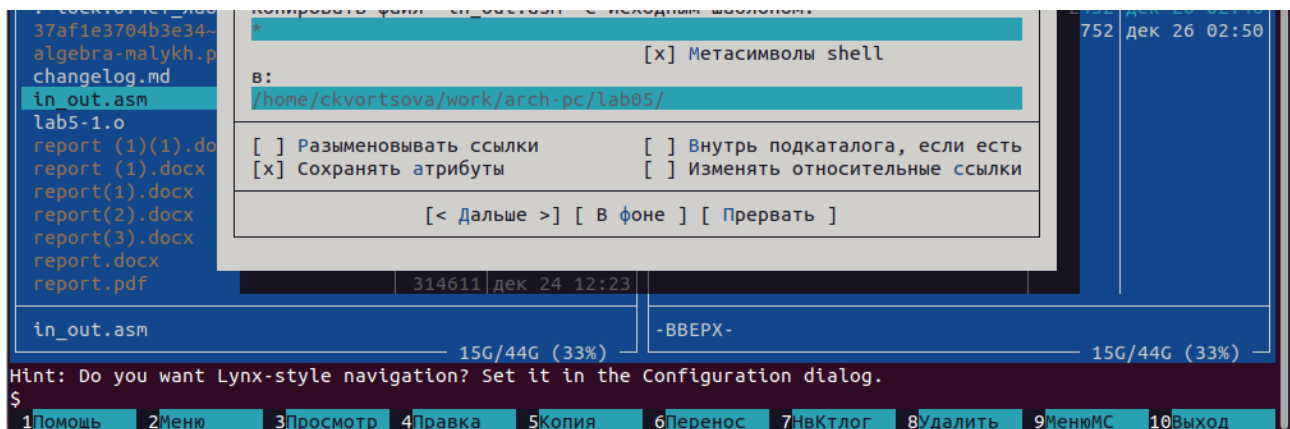


Рис. 10.2

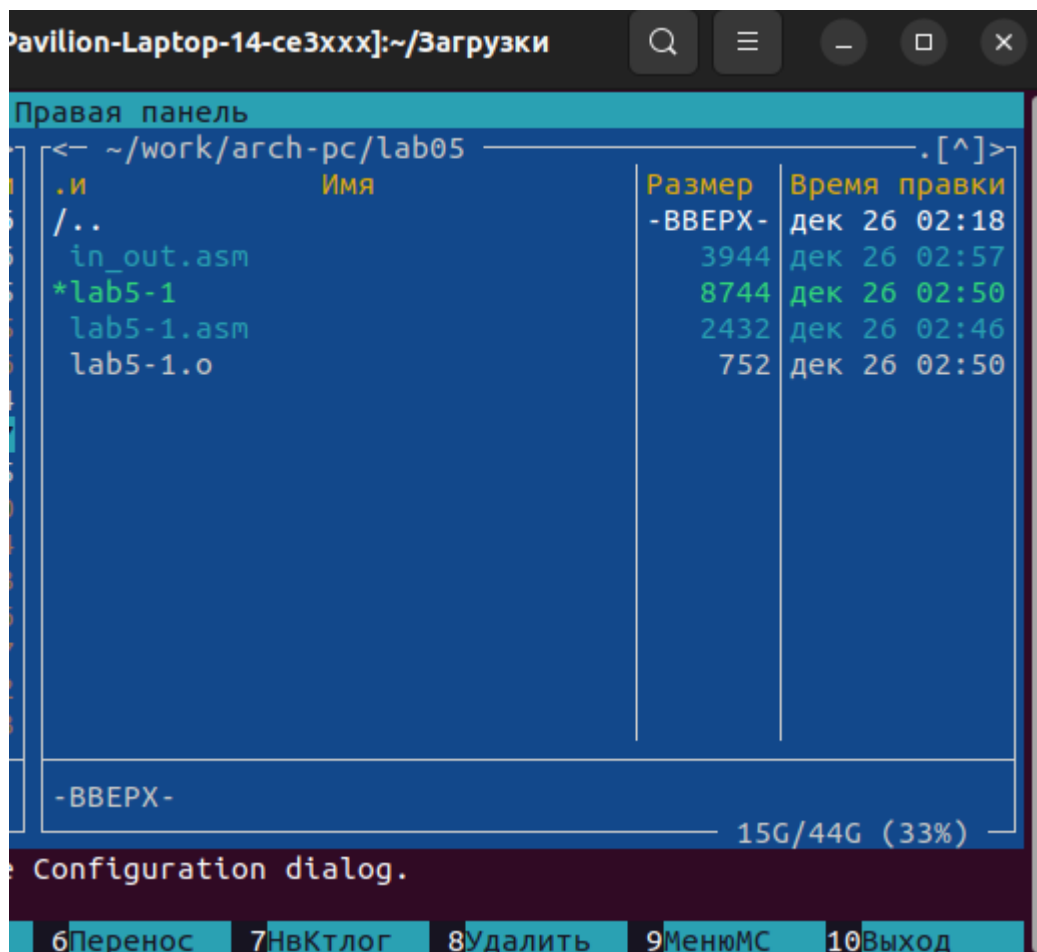


Рис. 10.3

- 10) 1. В одной из панелей mc открыл каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями использовал Tab). (см Рис. 10.1)
2. Скопировал файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5. (см Рис. 10.2 и Рис. 10.3)

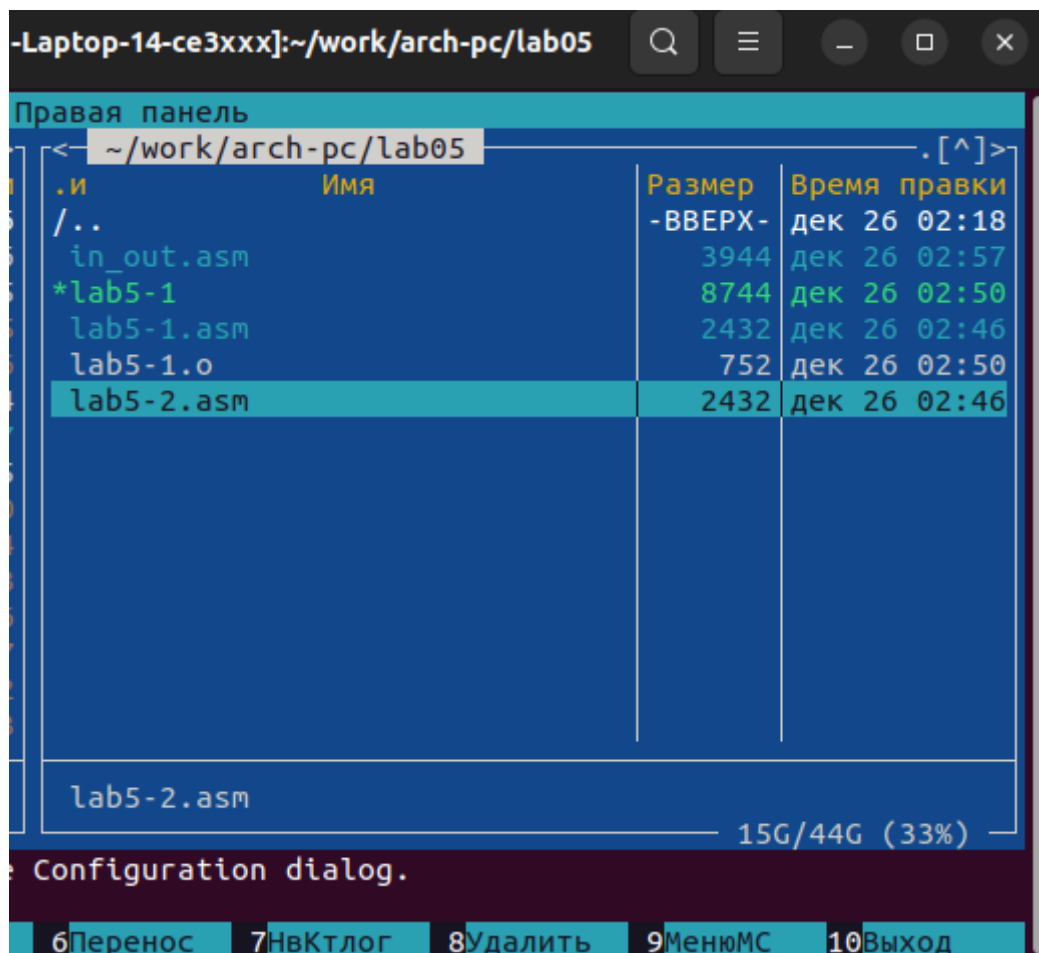


Рис. 11.1

11. С помощью функциональной клавиши F5 создал копию файла lab5-1.asm с именем lab5-2.asm. (см Рис. 11.1)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx: ~
GNU nano 6.2 /home/ckvortsova/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 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 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

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выровнять    ^/_ К строке    M-E Повтор
```

Рис. 12.1

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2.o
ld: no input files
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Скворцова Дарья Дмитриевна
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$
```

Рис. 12.2

12. Исправил текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используя подпрограммы sprintf, sread и quit) в соответствии с листингом 5.2 (см Рис. 12.1). Создал исполняемый файл и проверил его работу. (см Рис. 12.2)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Скворцова Дарья Дмитриевна
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ █
```

Рис. 13.1

13. В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Создал исполняемый файл и проверил его работу. Разница в том что sprintLF переносит ввод на следующую строку после “Введите строку”, а sprint нет.

## 5.4. 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 введённую строку на экран.

левая панель				правая панель			
файл				настройка			
команда				.[^]>			
Имя	Размер	Время правки		Имя	Размер	Время правки	
./..	-ВВЕРХ-	дек 26 02:18		./..	-ВВЕРХ-	окт 28 02:54	
in_out.asm	3944	дек 26 02:57		./TinyTeX	4096	дек 25 17:50	
*lab5-1	8744	дек 26 02:50		./cache	4096	дек 25 02:03	
lab5-1 copy.asm	2432	дек 26 02:46		./config	4096	дек 24 12:25	
lab5-1.asm	2432	дек 26 02:46		./local	4096	ноя 25 00:24	
lab5-1.o	752	дек 26 02:50		./mozilla	4096	дек 22 17:57	
*lab5-2	9092	дек 26 04:06		./ssh	4096	дек 22 19:27	
lab5-2.asm	1223	дек 26 03:58		./thunderbird	4096	дек 22 17:57	
lab5-2.o	1312	дек 26 04:05		/bin	4096	дек 24 13:34	
				/parentdir	4096	дек 23 18:01	
				/parentdir1	4096	дек 23 18:22	
				/parentdir2	4096	дек 23 18:05	
				/parentdir3	4096	дек 23 18:18	
				/snap	4096	ноя 30 22:10	

Рис. cp1.1



```
GNU nano 6.2 /home/ckvortsova/work/arch-pc/lab05/lab5-1copy.asm
;
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить     ^J Выворнять    ^/_ К строке    M-E Повтор
```

Рис. cp1.3

```

GNU nano 6.2 /home/ckvortsova/work/arch-pc/lab05/lab5-1copy.asm *
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
;----- Системный вызов повторный `write`
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес строки 'buf1' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- Системный вызов `exit` -----
; После вызова инструкции 'int 80h' программа завершит работу

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^\ Замена      ^U Вставить     ^J Выровнять    ^/_ К строке    M-E Повтор

```

Рис. cp1.4

```

GNU nano 6.2 /home/ckvortsova/work/arch-pc/lab05/lab5-1copy.asm *
; 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
;----- Системный вызов повторный `write`
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес строки 'buf1' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
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

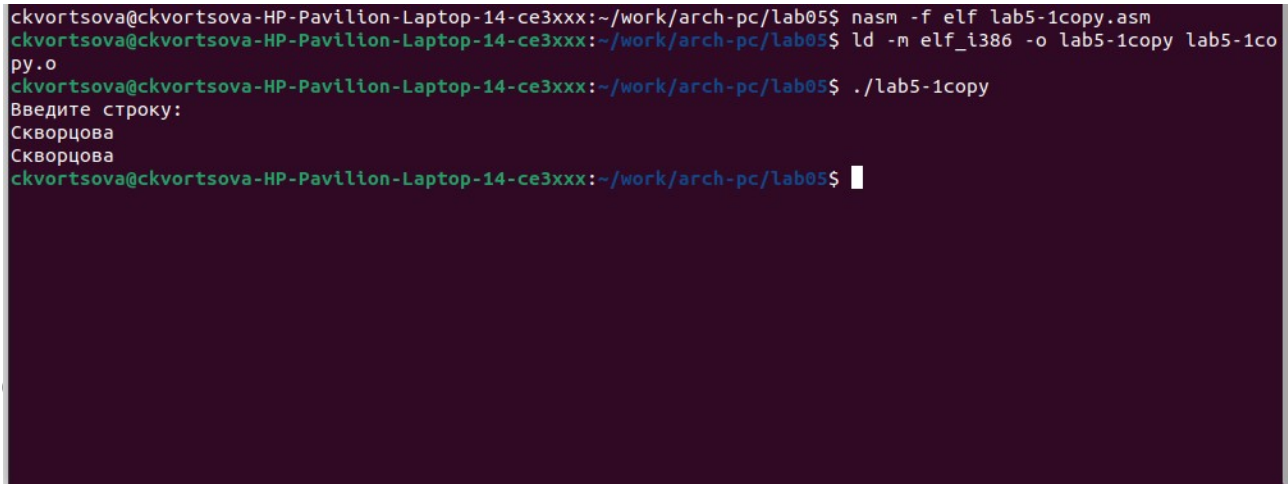
^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^\ Замена      ^U Вставить     ^J Выровнять    ^/_ К строке    M-E Повтор

```

Рис. cp1.5

Ответ: Создал копию файла lab5-1.asm с именем sm\_1.asm (см Рис. ср1.1). Внес изменения в программу (без использования внешнего файла in\_out.asm) (см Рис. ср1.5). Для вывода введенной ранее строки я добавил ещё один вызов 'write', и заменил адрес выводимой строки с msg на buf1. В buf1 ранее записалась введенная строка. (см Рис. ср1.5)

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



```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1copy lab5-1copy.o
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
Скворцова
Скворцова
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$
```

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.

```

GNU nano 6.2 /home/ckvortsova/work/arch-pc/lab05/lab5-2copy.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 `ECX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
;-----изменения-----
mov eax,buf1 ; из buf1 в `EAX`
mov ecx,eax ; из `EAX` в `ECX`

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^С Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выровнять    ^/ К строке     M-E Повтор

```

Рис. cp3.1

```

GNU nano 6.2 /home/ckvortsova/work/arch-pc/lab05/lab5-2copy.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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `ECX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
;-----изменения-----
mov eax,buf1 ; из buf1 в `EAX`
mov ecx,eax ; из `EAX` в `ECX`
call sprint ; вызов подпрограммы печати сообщения

call quit ; вызов подпрограммы завершения

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^С Позиция      M-U Отмена
^X Выход        ^R ЧитФайл     ^_ Замена       ^U Вставить     ^J Выровнять    ^/ К строке     M-E Повтор

```

Рис. cp3.2

Ответ: Создал копию файла lab5-2.asm с именем sm\_2.asm. Внес изменения в программу (с использованием внешнего файла in\_out.asm) (см Рис. ср3.2). Для вывода введенной ранее строки я переместил адрес переменной buf1 в есх, а затем из есх в еах и вызвал подпрограмму sprint. (см Рис. ср3.2)

#### 4. Создайте исполняемый файл и проверьте его работу.

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ nasm -f elf lab5-2copy.asm
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2copy lab5-2copy.o
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$ ./lab5-2copy
Введите строку:
Скворцова
Скворцова
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/arch-pc/lab05$
```

## Вывод

В ходе выполнения лабораторной работы я приобрел практические навыки работы в Midnight Commander. Освоил инструкции языка ассемблера mov и int

## Список литературы

- 1.Пример выполнения лабораторной работы
- 2.Курс на ТУИС
- 3.Лабораторная работа №5
- 4.Программирование на языке ассемблера NASM Столяров А. В.  
[https://esystem.rudn.ru/pluginfile.php/2088953/mod\\_resource/content/2/%D0%A1%D1%82%D0%BE%D0%BB%D1%8F%D1%80%D0%BE%D0%B2%20%D0%90.%20%D0%92.%20-%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM%20%D0%B4%D0%BB%D1%8F%20%D0%9E%D0%A1%20Unix.pdf](https://esystem.rudn.ru/pluginfile.php/2088953/mod_resource/content/2/%D0%A1%D1%82%D0%BE%D0%BB%D1%8F%D1%80%D0%BE%D0%B2%20%D0%90.%20%D0%92.%20-%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM%20%D0%B4%D0%BB%D1%8F%20%D0%9E%D0%A1%20Unix.pdf)