

Лабораторная работа №6

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Гузева Ирина Николаевна

Table of Contents

| | | |
|---|-------------------------------------|----|
| 1 | Цель работы | 1 |
| 2 | Выполнение лабораторной работы..... | 1 |
| 3 | Выводы..... | 10 |

1 Цель работы

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

2 Выполнение лабораторной работы

1. Открыла mc и зашла в папку lab5 (рис. 1)

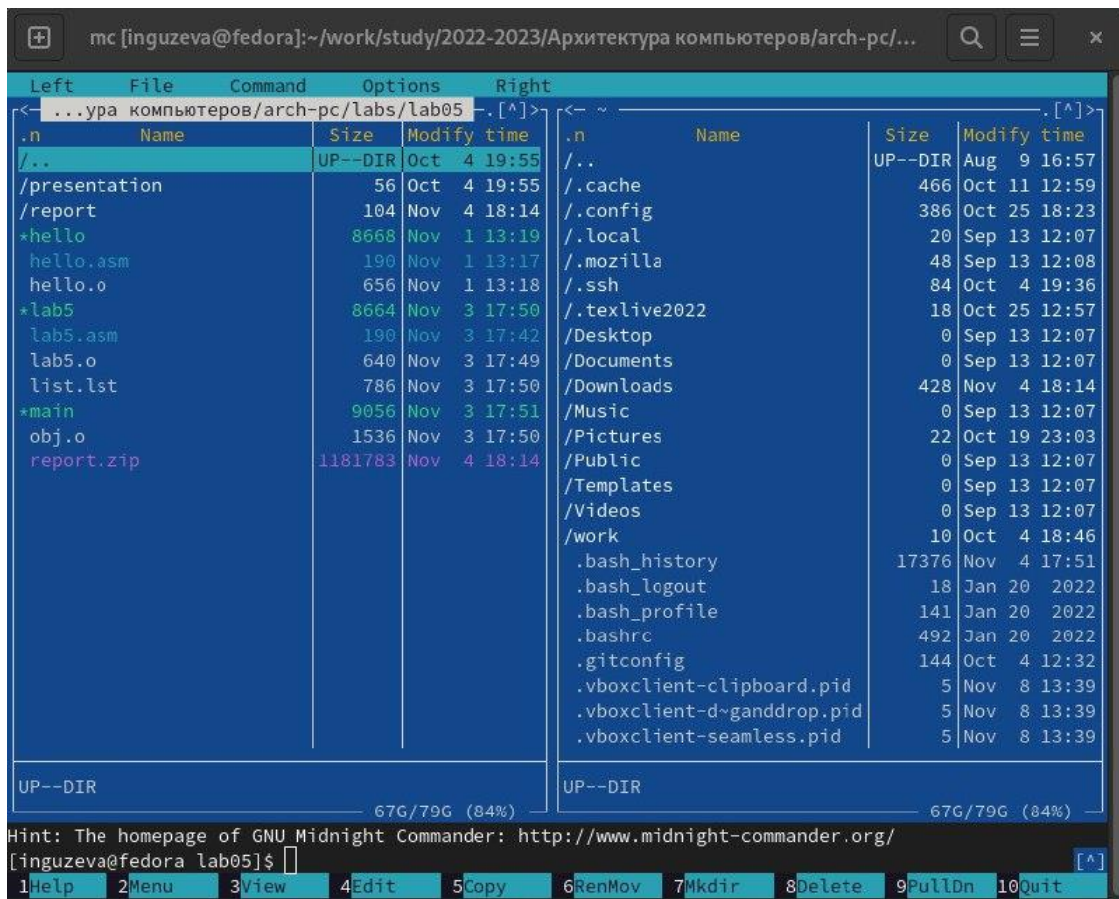
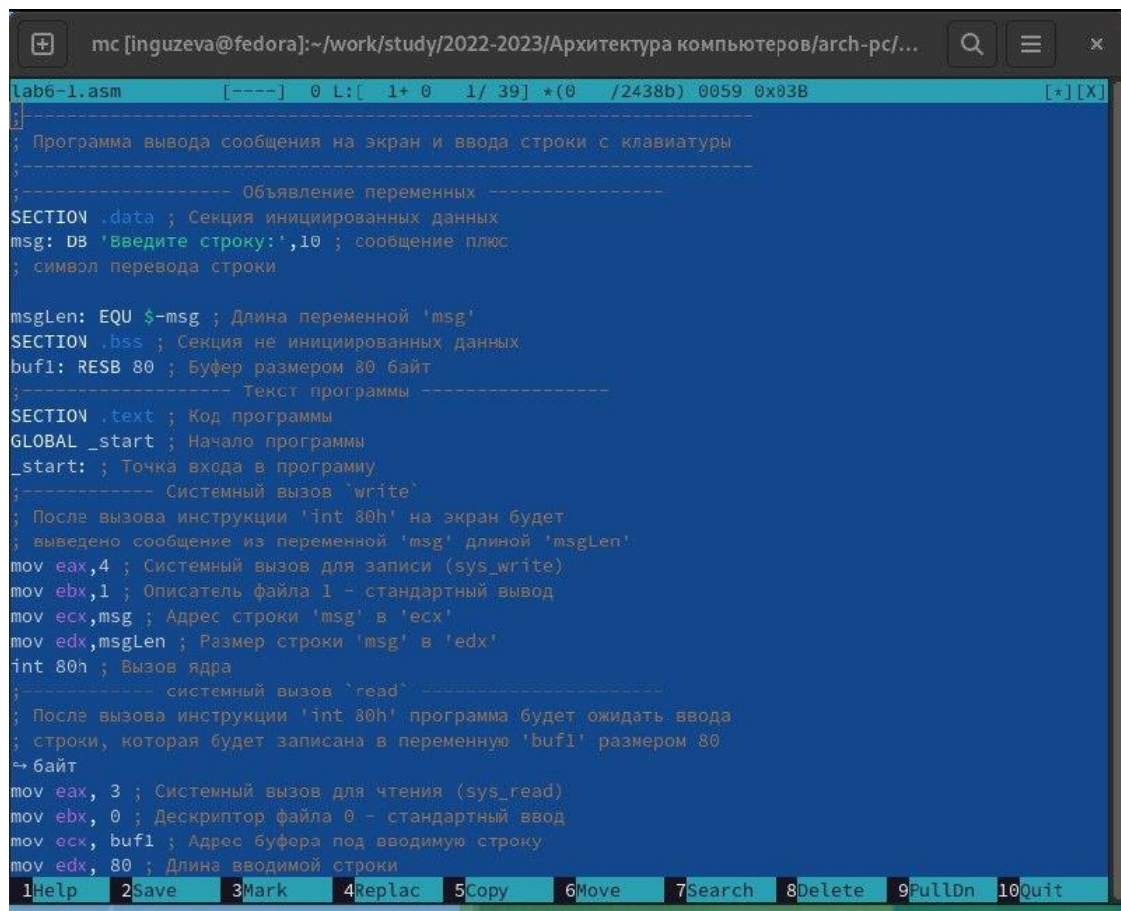


Figure 1: Панка lab5

2. С помощью команды touch в строке ввода создала файл lab6-1.asm (рис. 2)



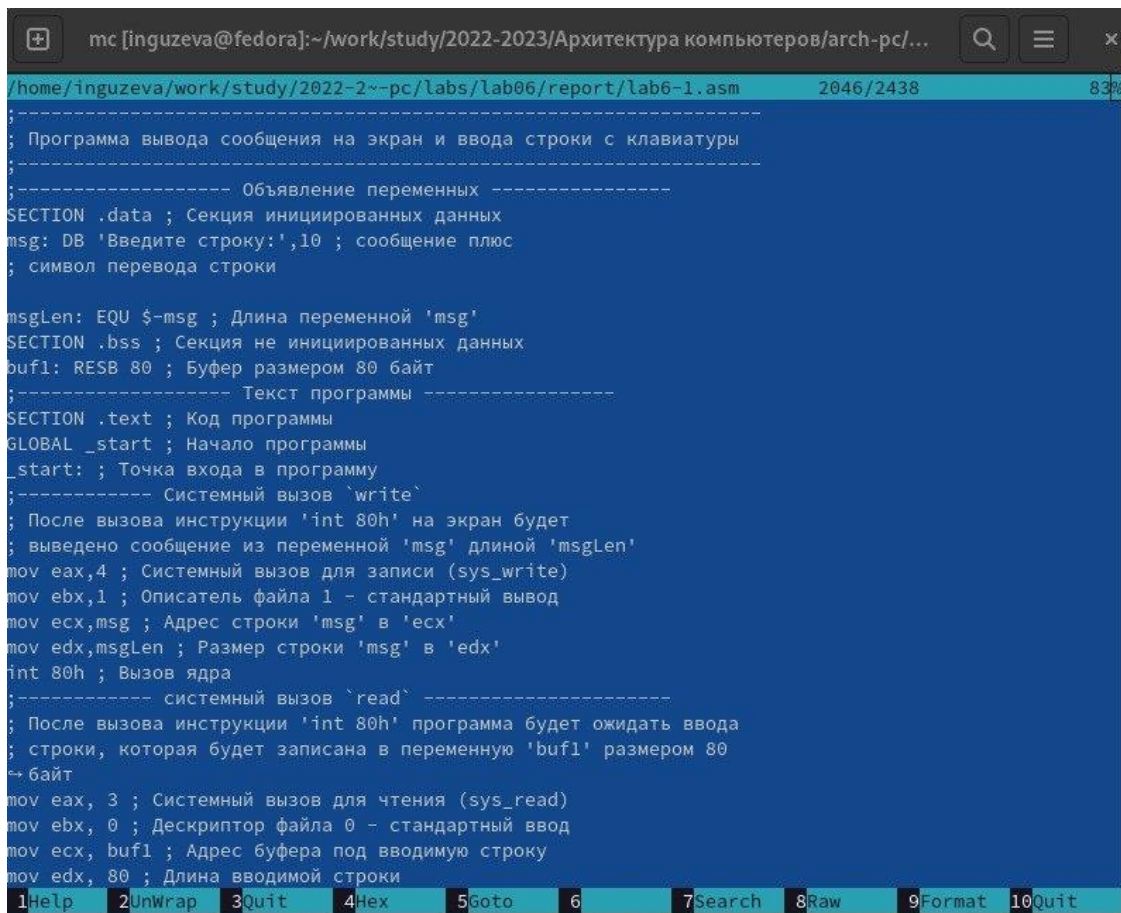
The screenshot shows a text editor window titled "mc [inguzeva@fedora]:~/work/study/2022-2023/Архитектура компьютеров/arch-pc/...". The editor displays assembly code for a file named "lab6-1.asm". The code includes comments in Russian and assembly instructions. At the bottom, there is a menu bar with 10 items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9FullDn, and 10Quit.

```
lab6-1.asm [----] 0 L: [ 1+ 0 1/ 39] *(0 /2438b) 0059 0x03B [*] [X]
;
; 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
;----- 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
```

Figure 3: Программа

4. С помощью клавиши F3 открыла текст программы (рис. 4)

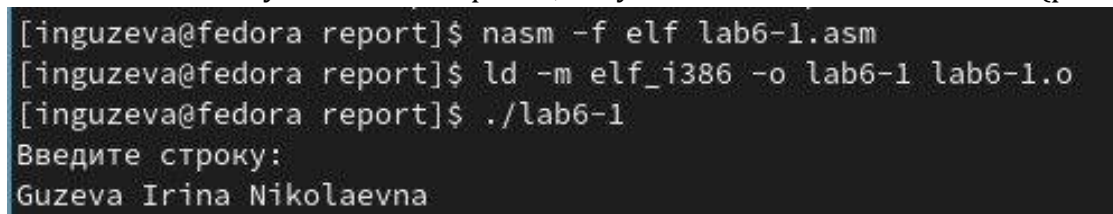


```
mc [inguzeva@fedora]:~/work/study/2022-2023/Архитектура компьютеров/arch-pc/...
/home/inguzeva/work/study/2022-2~pc/labs/lab06/report/lab6-1.asm 2046/2438 83%
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
SECTION .data ; Секция инициированных 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'
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
1Help 2Unwrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit
```

Figure 4: Текст программы

- Оттранслировала текст программы lab6-1.asm в объектный файл, выполнила компоновку объектного файла, запустила его и ввела свои ФИО (рис. 5)



```
[inguzeva@fedora report]$ nasm -f elf lab6-1.asm
[inguzeva@fedora report]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[inguzeva@fedora report]$ ./lab6-1
Введите строку:
Guzeva Irina Nikolaevna
```

Figure 5: Запуск программы

- Скачала файл in_out.asm с ТУИС и скопировала его в нужную папку с помощью клавиши F5 (рис. 6)

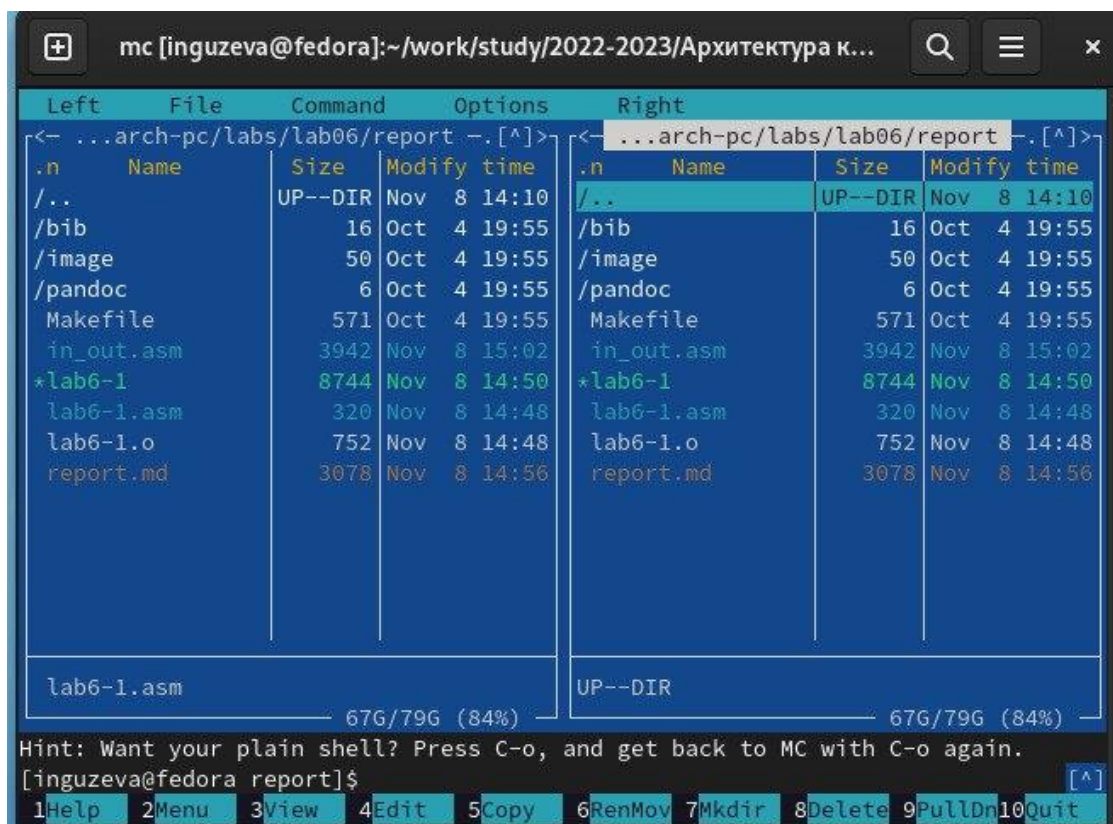
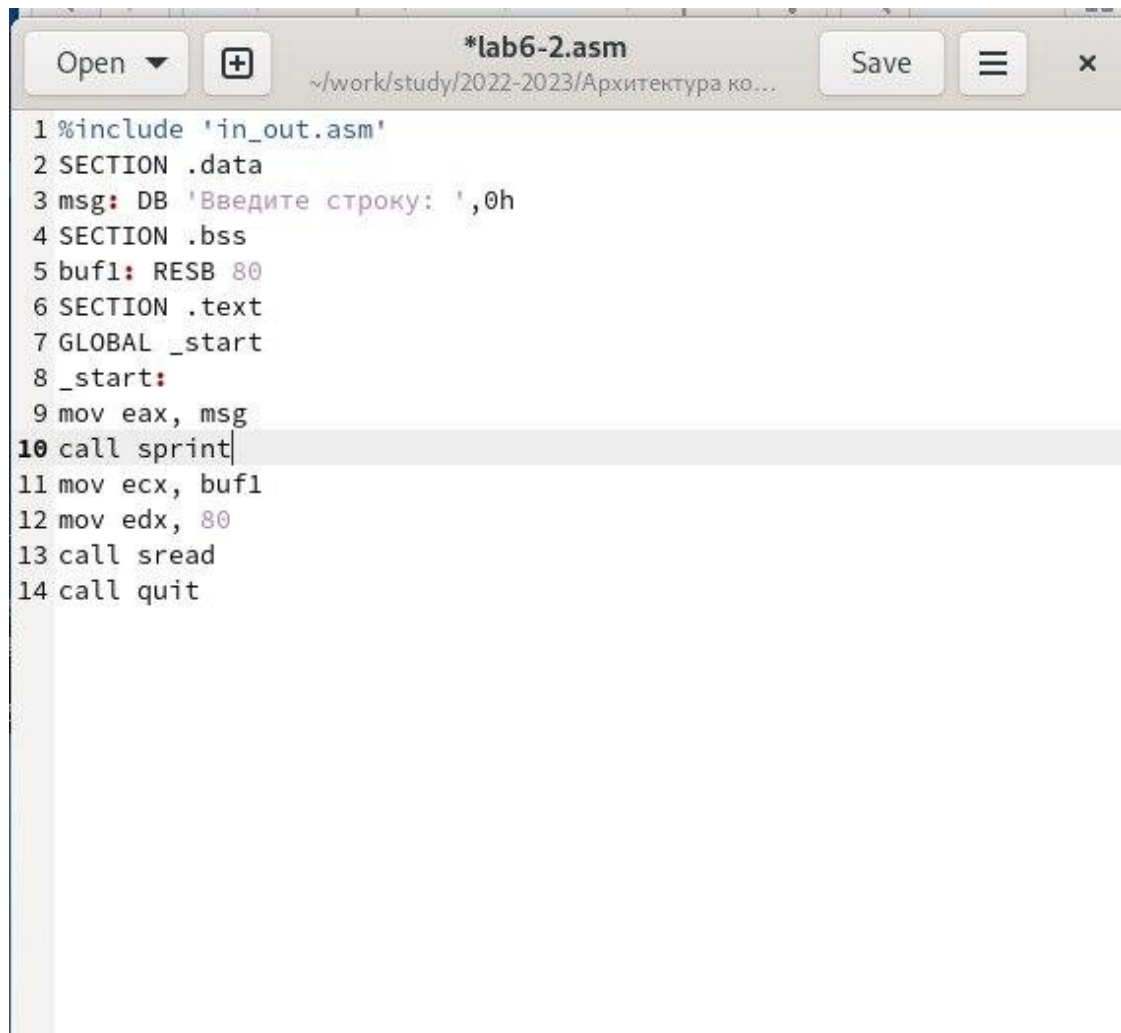


Figure 6: Копирование файла `in_out.asm`

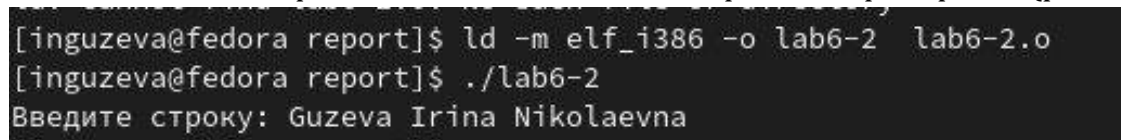
7. С помощью клавиши F5 скопировала файл `lab6-1.asm` и назвала его `lab6-2.asm` (рис. 7)



```
*lab6-2.asm
~/work/study/2022-2023/Архитектура ко...
Open Save
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите строку: ',0h
4 SECTION .bss
5 buf1: RESB 80
6 SECTION .text
7 GLOBAL _start
8 _start:
9 mov eax, msg
10 call sprint
11 mov ecx, buf1
12 mov edx, 80
13 call sread
14 call quit
```

Figure 8: Исправление текста программы

9. Создала исполняемый файл и проверила его работу. Разница в том что, введенная строка выводится не с новой строки, а через пробел (рис. 9)



```
[inguzeva@fedora report]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[inguzeva@fedora report]$ ./lab6-2
Введите строку: Guzeva Irina Nikolaevna
```

Figure 9: Исправление текста программы

10. Создала копии файлов lab6-1.asm и lab6-2.asm (рис. 10)

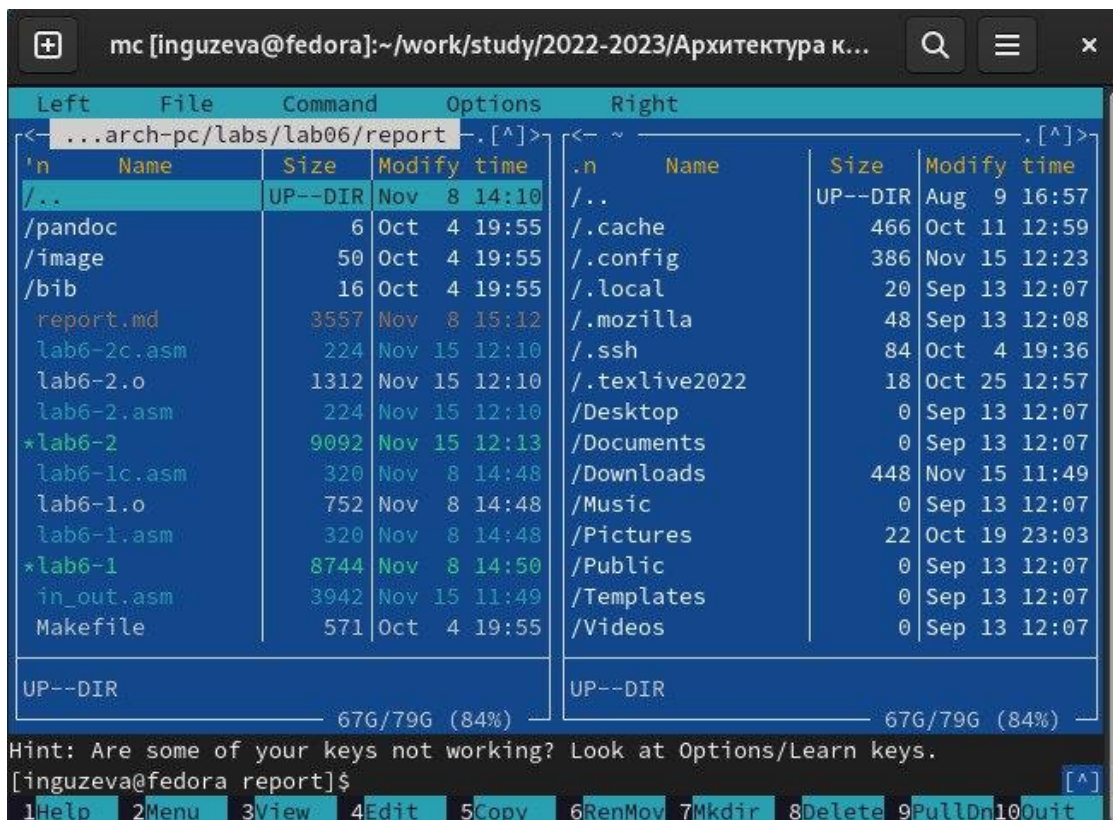


Figure 10: Копирование файлов

11. Изменила программу файла lab6-2c.asm (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. (рис. 11)

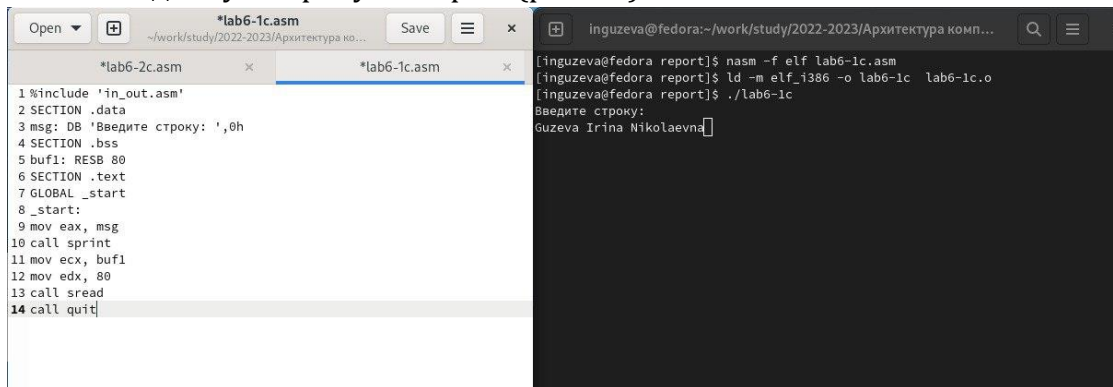
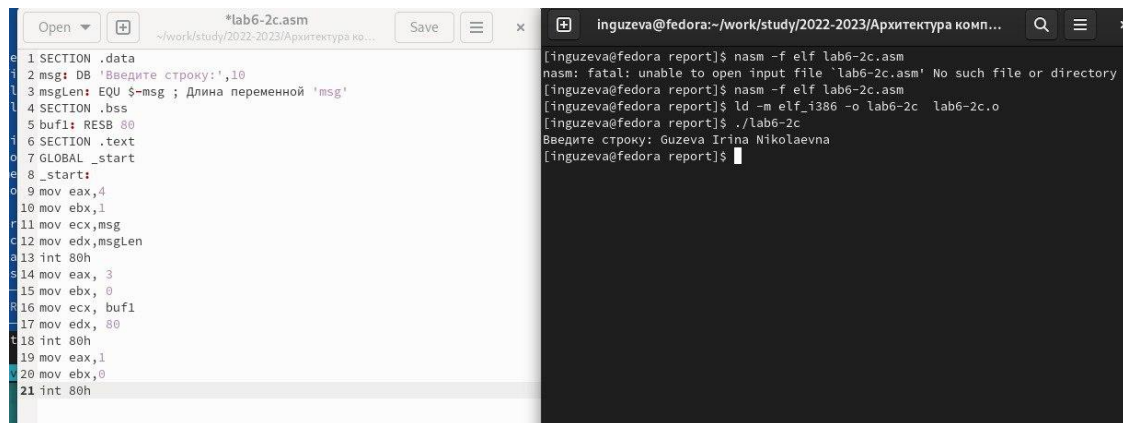


Figure 11: Изменение программы 2

12. Изменила программу файла lab6-1c.asm (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. (рис. 12)



The image shows a code editor window on the left and a terminal window on the right. The code editor displays an assembly file named `lab6-2c.asm` with the following content:

```
1 SECTION .data
2 msg: DB 'Введите строку:',10
3 msgLen: EQU $-msg ; Длина переменной 'msg'
4 SECTION .bss
5 buf1: RESB 80
6 SECTION .text
7 GLOBAL _start
8 _start:
9 mov eax,4
10 mov ebx,1
11 mov ecx,msg
12 mov edx,msgLen
13 int 80h
14 mov eax,3
15 mov ebx,0
16 mov ecx,buf1
17 mov edx,80
18 int 80h
19 mov eax,1
20 mov ebx,0
21 int 80h
```

The terminal window shows the following commands and output:

```
[inguzeva@fedora report]$ nasm -f elf lab6-2c.asm
nasm: fatal: unable to open input file 'lab6-2c.asm' No such file or directory
[inguzeva@fedora report]$ nasm -f elf lab6-2c.asm
[inguzeva@fedora report]$ ld -m elf_i386 -o lab6-2c lab6-2c.o
[inguzeva@fedora report]$ ./lab6-2c
Введите строку: Guzeva Irina Nikolaevna
[inguzeva@fedora report]$
```

Figure 12: Изменение программы 1

3 Выводы

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