

# **Лабораторная работа №5**

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

Баранов Никита Дмитриевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Самостоятельная работа</b>	<b>13</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>

# Список иллюстраций

3.1	Используем команду <code>mc</code> . . . . .	7
3.2	Создаем папку <code>lab05</code> . . . . .	8
3.3	С помощью команды <code>touch</code> создаем файл . . . . .	8
3.4	Редактируем файл . . . . .	9
3.5	Оттранслируем текст программы в объектный файл и запустим файл . . . . .	9
3.6	Копируем файл в нужный каталог . . . . .	10
3.7	Копируем файл, переименовывая его в <code>lab5-2.asm</code> . . . . .	11
3.8	Редактируем файл с использованием подпрограмм . . . . .	11
3.9	Запускаем файл . . . . .	11
3.10	Меняем подпрограммы . . . . .	12
3.11	Запускаем файл и сравниваем с прошлым результатом . . . . .	12
4.1	Копируя, создаем новый файл <code>lab5-3.asm</code> . . . . .	13
4.2	Редактируем файл . . . . .	14
4.3	Запускаем файл . . . . .	14
4.4	Создаем новый файл, копируя прошлые . . . . .	15
4.5	Редактируем файл . . . . .	15
4.6	Запускаем файл . . . . .	15

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

Написать 2 программы по примеру и впоследствии изменить их по условию.

### 3 Выполнение лабораторной работы

Откройте Midnight Commander. Пользуясь клавишами ↑, ↓ и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4. С помощью функциональной клавиши F7 создайте папку lab05 (рис. 5.3) и перейдите в созданный каталог. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm (рис. fig. 3.1)(рис. fig. 3.2)(рис. fig. 3.3).

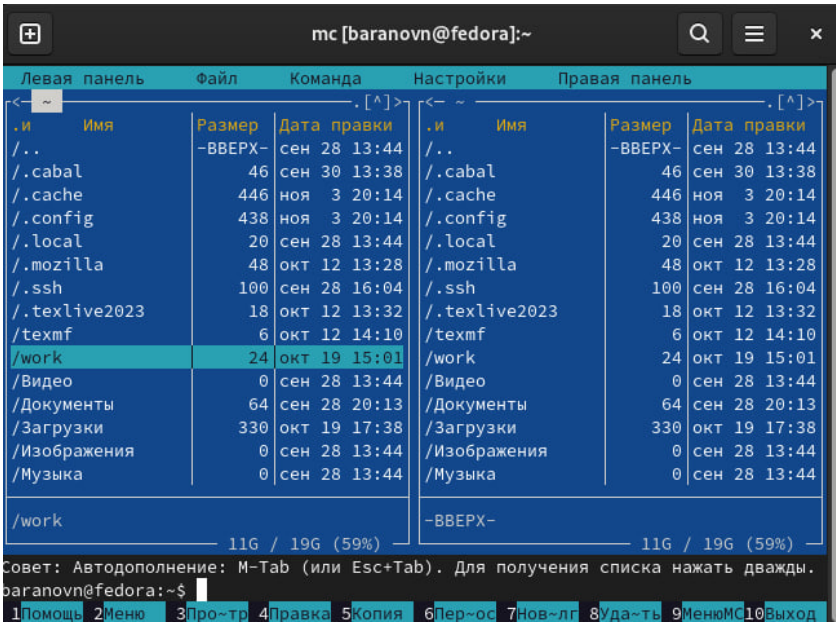


Рис. 3.1: Используем команду mc

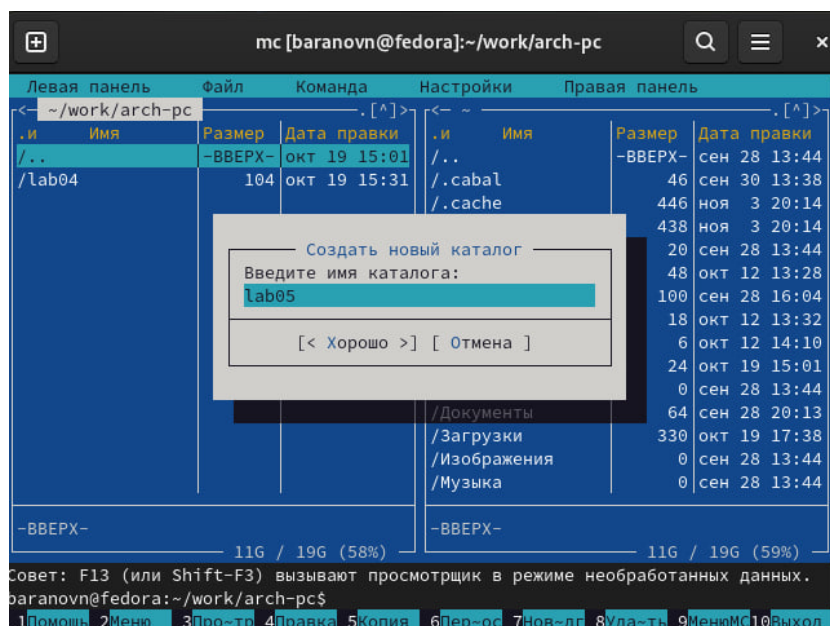


Рис. 3.2: Создаем папку lab05

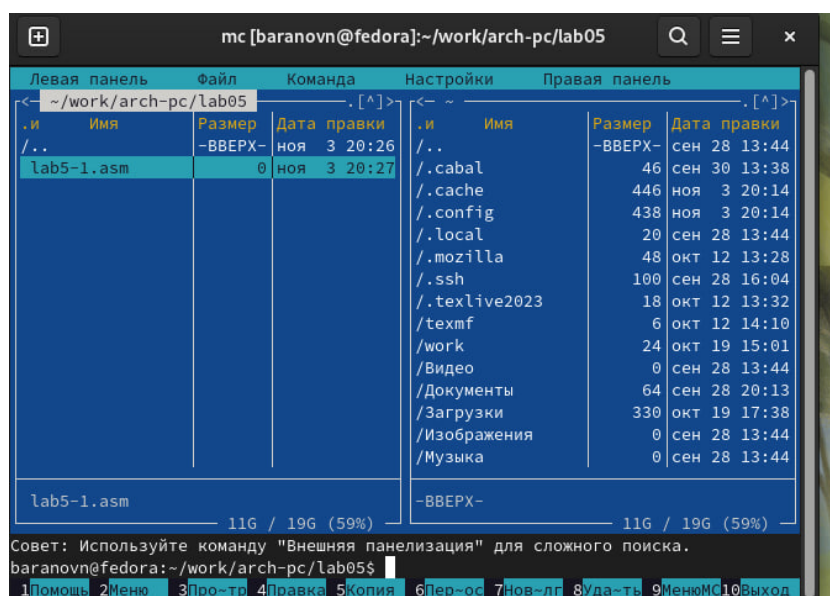
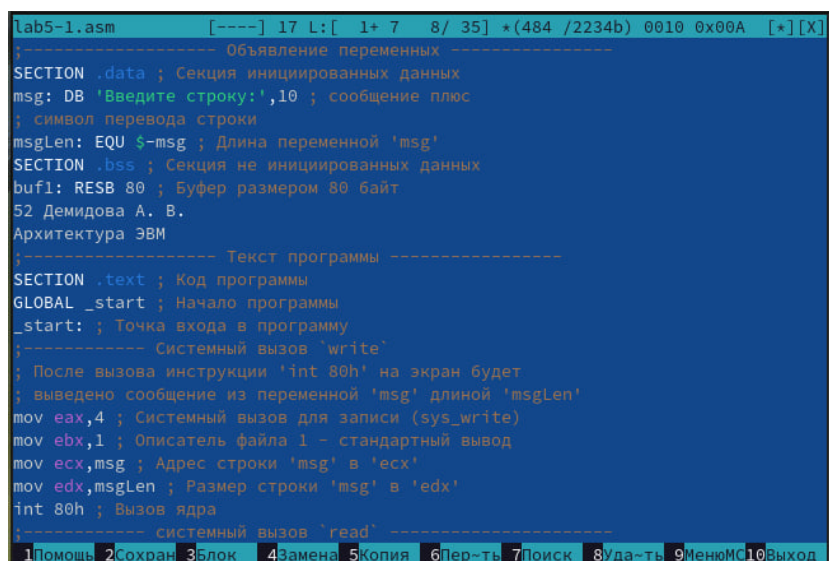


Рис. 3.3: С помощью команды touch создаем файл

С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano (рис. 5.5) или mcedit. Введите текст программы из листинга 5.1 (можно без комментариев),

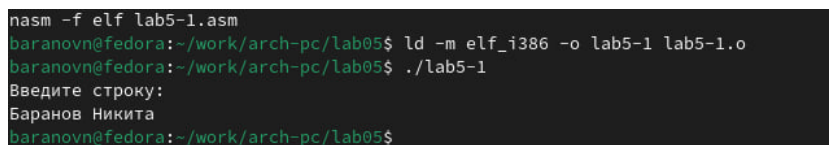


сохраните изменения и закройте файл. Оттранслируйте текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл(рис. fig. 3.4)(рис. fig. 3.5).



```
lab5-1.asm      [----] 17 L: [ 1+ 7 8/ 35] *(484 /2234b) 0010 0x00A [*] [X]
;----- 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
52 Демидова А. В.
Архитектура ЭВМ
;----- 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' -----
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 3.4: Редактируем файл



```
nasm -f elf lab5-1.asm
baranovn@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
baranovn@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Баранов Никита
baranovn@fedora:~/work/arch-pc/lab05$
```

Рис. 3.5: Оттранслируем текст программы в объектный файл и запустим файл

Скачайте файл in\_out.asm. В одной из панелей mc откройте каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями используйте Tab ).Скопируйте файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.(рис. fig. 3.6).

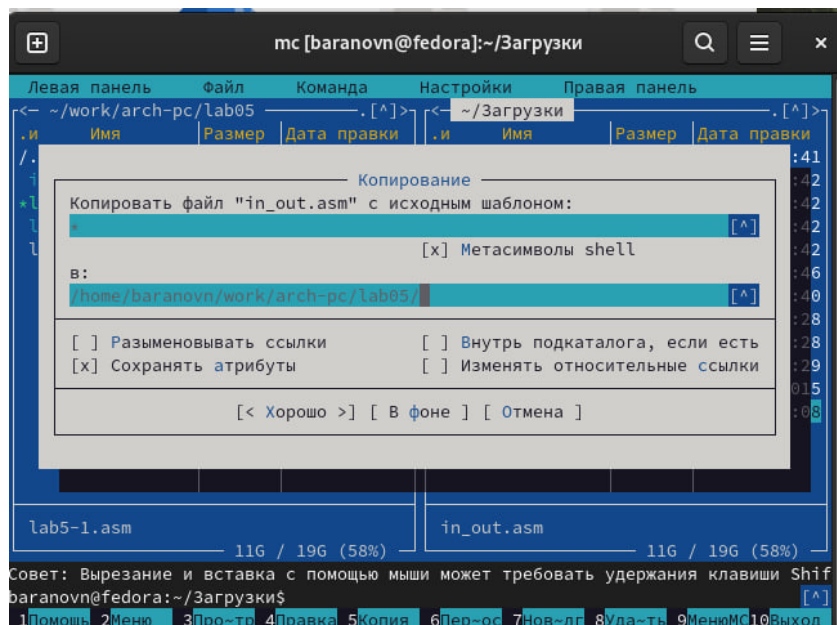


Рис. 3.6: Копируем файл в нужный каталог

С помощью функциональной клавиши F6 создайте копию файла lab5-1.asm с именем lab5-2.asm. Исправьте текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используйте подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу. В файле lab5-2.asm замените подпрограмму sprintLF на sprint. Создайте исполняемый файл и проверьте его работу. В чем разница? (рис. fig. 3.7)(рис. fig. 3.8)(рис. fig. 3.9)(рис. fig. 3.10)(рис. fig. 3.11).

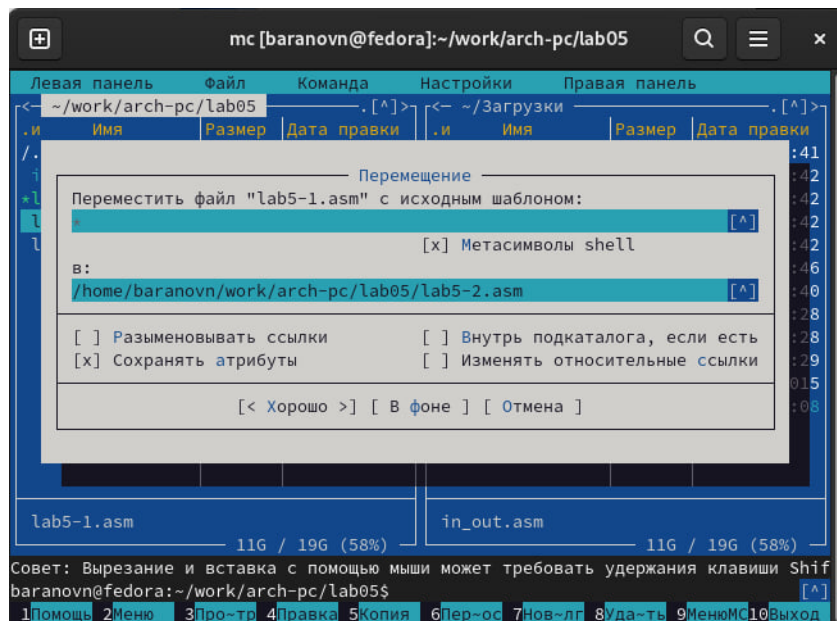


Рис. 3.7: Копируем файл, переименовывая его в lab5-2.asm

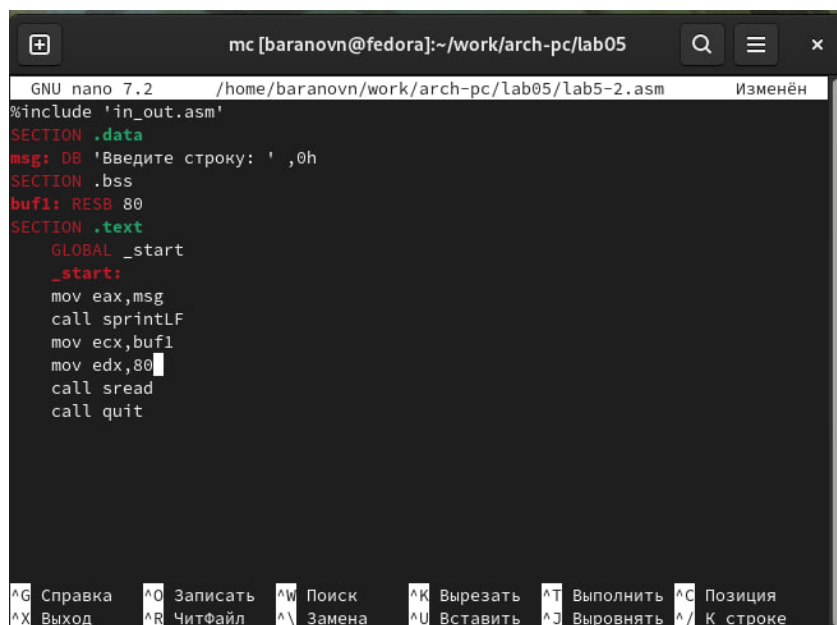


Рис. 3.8: Редактируем файл с использованием подпрограмм

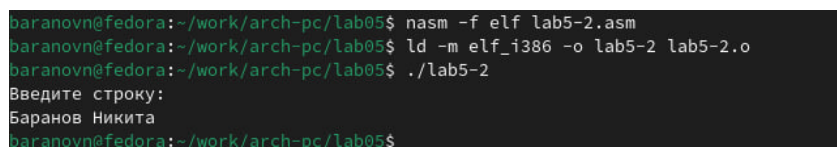
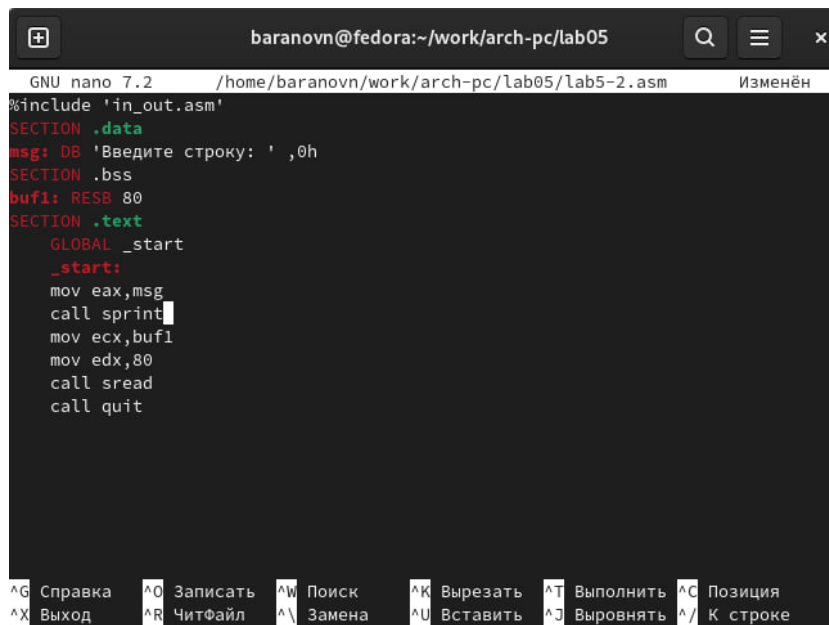


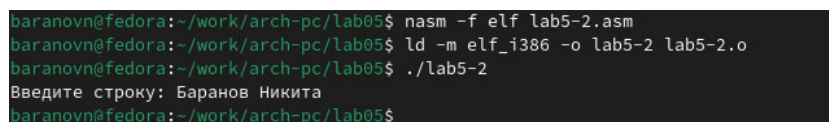
Рис. 3.9: Запускаем файл



```
GNU nano 7.2 /home/baranov/work/arch-pc/lab05/lab5-2.asm
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
bufi: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprint
mov ecx,bufi
mov edx,80
call sread
call quit
```

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

Рис. 3.10: Меняем подпрограммы



```
baranov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
baranov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
baranov@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Баранов Никита
baranov@fedora:~/work/arch-pc/lab05$
```

Рис. 3.11: Запускаем файл и сравниваем с прошлым результатом

Сравнивая два программы, я сделал вывод, что `sprintLF` переносит ввод на новую строку, тогда как `sprint` дает вводить строку без переноса.

## 4 Самостоятельная работа

Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она выводила строку которую вы ввели(рис. fig. 4.1)(рис. fig. 4.2)(рис. fig. 4.3).

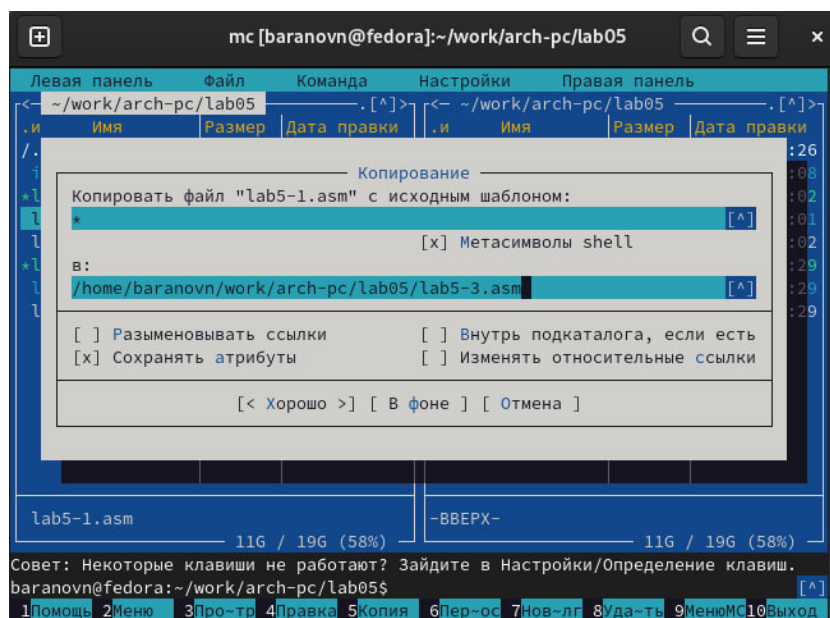


Рис. 4.1: Копируя, создаем новый файл lab5-3.asm

```
GNU nano 7.2 lab5-3.asm
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 0x80

    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 0x80

    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    mov edx, 80
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 4.2: Редактируем файл

```
nasm -f elf lab5-3.asm
baranov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
baranov@fedora:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Баранов Никита
Баранов Никита
baranov@fedora:~/work/arch-pc/lab05$
```

Рис. 4.3: Запускаем файл

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(рис. fig. 4.4)(рис. fig. 4.5)(рис. fig. 4.6).

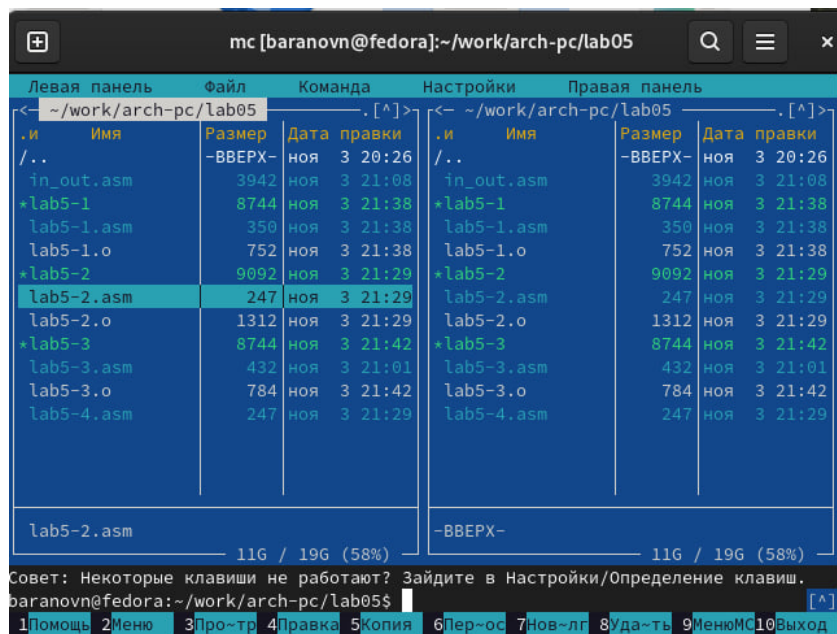


Рис. 4.4: Создаем новый файл, копируя прошлые

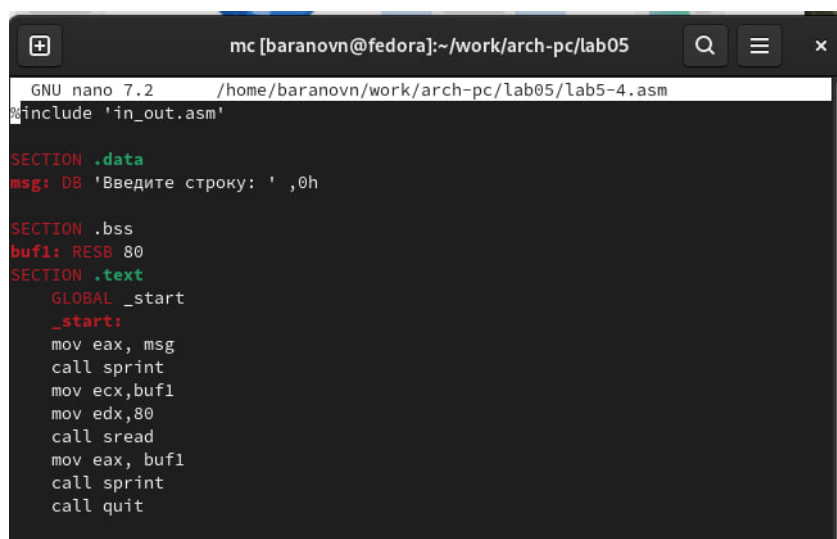


Рис. 4.5: Редактируем файл

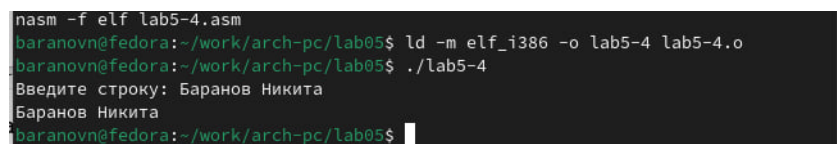


Рис. 4.6: Запускаем файл

## 5 Выводы

Мы приобрели навыки работы с Midnight Commander и освоили инструкцию mov.