

# **Отчет по лабораторной работе № 6**

**НММбд-03-22**

**Толстых Максим Алексеевич**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Работа с Midnight Commander . . . . .	6
3.2	Задание для самостоятельной работы . . . . .	13
<b>4</b>	<b>Выводы</b>	<b>16</b>

## Список иллюстраций

3.1	Midnight Commander . . . . .	6
3.2	Каталог ~/work/arch-pc . . . . .	6
3.3	lab06 . . . . .	7
3.4	touch . . . . .	7
3.5	lab6-1.asm . . . . .	8
3.6	Файл lab6-1.asm для редактирования . . . . .	8
3.7	Текст программы из листинга 6.1 . . . . .	9
3.8	Файл содержащий текст программы . . . . .	9
3.9	Получившийся исполняемый файл . . . . .	10
3.10	Скачанный файл in_out.asm . . . . .	10
3.11	Копирование файла in_out.asm . . . . .	11
3.12	lab6-2.asm . . . . .	11
3.13	sprintLF, sread и quit . . . . .	12
3.14	Исполняемый файл . . . . .	12
3.15	sprint . . . . .	12
3.16	Перенос строки перед вводом текста . . . . .	13
3.17	lab6-3.asm . . . . .	13
3.18	Изменения в программе . . . . .	14
3.19	Проверка работы файла . . . . .	14
3.20	lab6-4.asm . . . . .	15
3.21	Использование подпрограмм из внешнего файла in_out.asm . . . . .	15
3.22	Проверка работы исполняемого файла . . . . .	15

# 1 Цель работы

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

## 2 Задание

Создать исполняющие файлы из файлов ассемблера, реализующие ввод и вывод текста.

## 3 Выполнение лабораторной работы

### 3.1 Работа с Midnight Commander

Открыли Midnight Commander (рис. 3.1)

```
[matolstikh@fedora ~]$ mc
```

Рис. 3.1: Midnight Commander

Пользуясь клавишами **←**, **→** и **Enter** перешли в каталог `~/work/arch-pc`, созданный при выполнении лабораторной работы №5. (рис. 3.2)

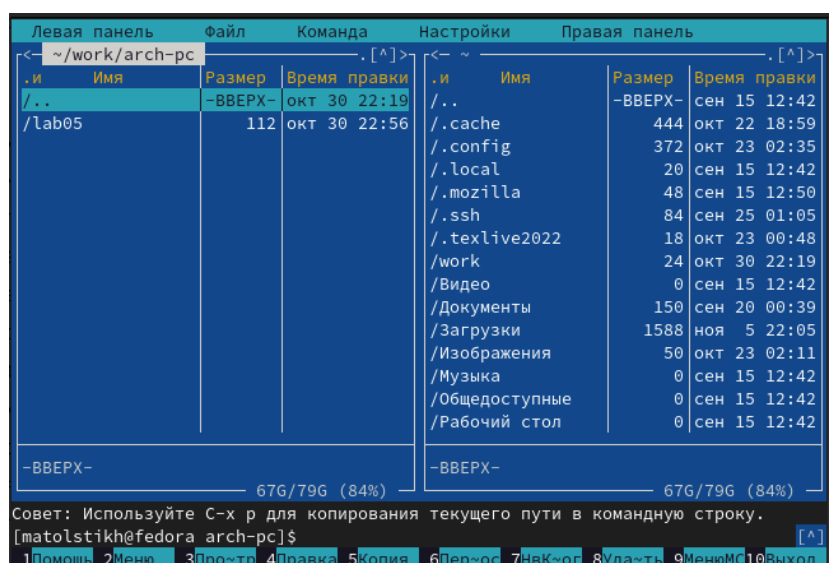


Рис. 3.2: Каталог `~/work/arch-pc`

С помощью функциональной клавиши **F7** создали папку `lab06` и перешли в созданный каталог. (рис. 3.3)



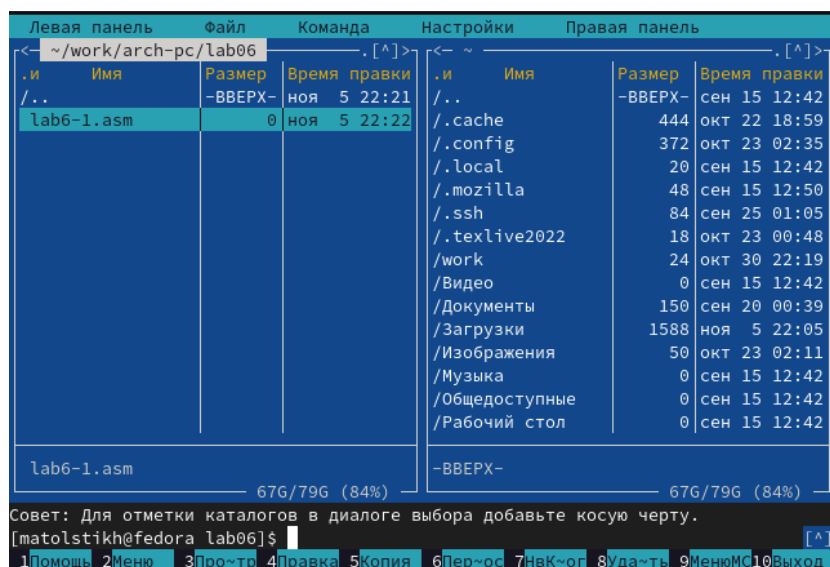


Рис. 3.5: lab6-1.asm

С помощью функциональной клавиши F4 откройте файл lab6-1.asm для редактирования во встроенном редакторе. (рис. 3.6)

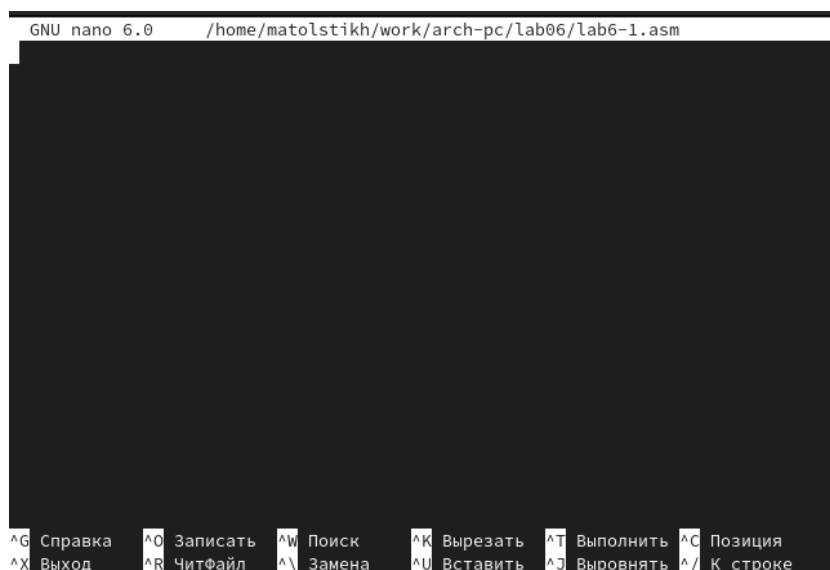


Рис. 3.6: Файл lab6-1.asm для редактирования

Ввели текст программы из листинга 6.1, сохранили изменения и закрыли файл. (рис. 3.7)



```

GNU nano 6.0 /home/matolstikh/work/arch-pc/lab06/lab6-1.asm Изменён
;-----
; 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
100 Демидова А. В.
Архитектура ЭВМ
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)

```

Рис. 3.7: Текст программы из листинга 6.1

С помощью функциональной клавиши F3 открыли файл lab6-1.asm для просмотра. Убедились, что файл содержит текст программы. (рис. 3.8)

```

/home/matolstikh/work/arch-pc/lab06/lab6-1.asm 1446/2433 59%
;-----
; 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

```

Рис. 3.8: Файл содержащий текст программы

Оттранслировали текст программы lab6-1.asm в объектный файл. Выполнили компоновку объектного файла и запустили получившийся исполняемый файл. Программа вывела строку 'Введите строку:' и ожидала ввода с клавиатуры. На

запрос ввели Maksim. (рис. 3.9)

```
[matolstikh@fedora lab06]$ nasm -f elf lab6-1.asm
[matolstikh@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[matolstikh@fedora lab06]$ ./lab6-1
Введите строку:
Maksim
[matolstikh@fedora lab06]$
```

Рис. 3.9: Получившийся исполняемый файл

Скачали файл in\_out.asm со страницы курса в ТУИС. (рис. 3.10)

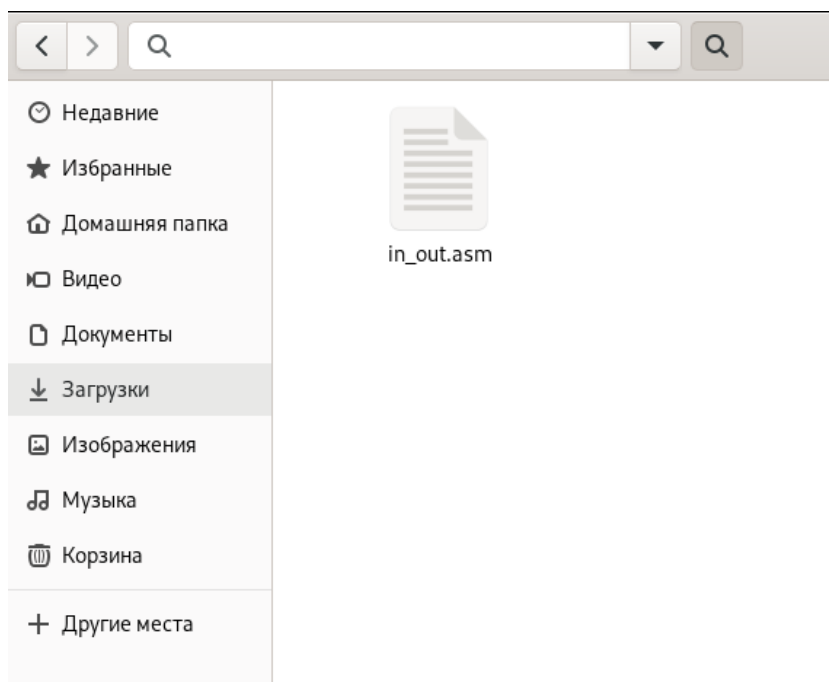


Рис. 3.10: Скачанный файл in\_out.asm

Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется. В одной из панелей mc открыли каталог с файлом lab6-1.asm. В другой панели каталог со скаченным файлом in\_out.asm. Скопировали файл in\_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5.(рис. 3.11)

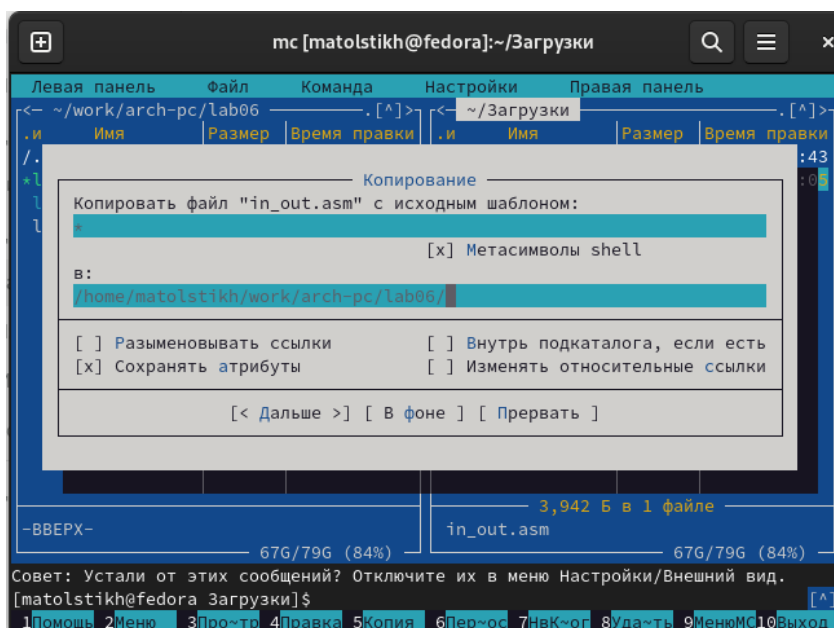


Рис. 3.11: Копирование файла in\_out.asm

С помощью функциональной клавиши F6 создали копию файла lab6- 1.asm с именем lab6-2.asm. Выделили файл lab6-1.asm, нажали клавишу F6 , ввели имя файла lab6-2.asm и нажали клавишу Enter. (рис. 3.12)

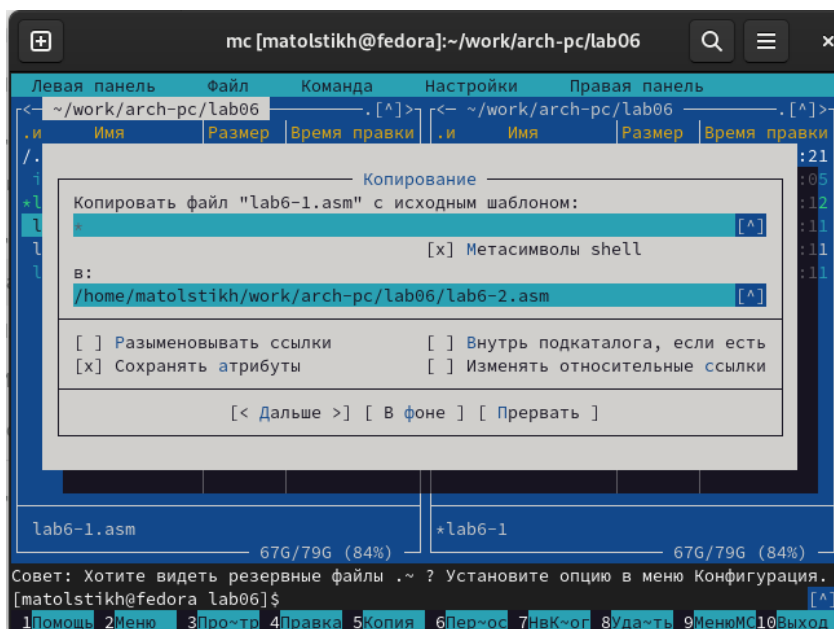


Рис. 3.12: lab6-2.asm

Исправили текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (использовали подпрограммы sprintf, sread и quit) в соответствии с листингом 6.2. Создали исполняемый файл и проверьте его работу. (рис. 3.13), (рис. 3.14)

```

GNU nano 6.0 /home/matolstikh/work/arch-pc/lab06/lab6-2.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

```

Рис. 3.13: sprintf, sread и quit

```

[matolstikh@fedora lab06]$ nasm -f elf lab6-2.asm
[matolstikh@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[matolstikh@fedora lab06]$ ./lab6-2
Введите строку:
Maksim
[matolstikh@fedora lab06]$

```

Рис. 3.14: Исполняемый файл

В файле lab6-2.asm заменили подпрограмму sprintf на sprintf. Создали исполняемый файл и проверили его работу. При замене подпрограмм пропадает перенос строки перед вводом текста. (рис. 3.15), (рис. 3.16)

```

call sprintf ; вызов подпрограммы печати сообщения

```

Рис. 3.15: sprintf

```

[matolstikh@fedora lab06]$ nasm -f elf lab6-2.asm
[matolstikh@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[matolstikh@fedora lab06]$ ./lab6-2
Введите строку: Maksim
[matolstikh@fedora lab06]$

```

Рис. 3.16: Перенос строки перед вводом текста

## 3.2 Задание для самостоятельной работы

Создали копию файла lab6-1.asm. Внесли изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: 1. вывести приглашение типа “Введите строку:”; 2. ввести строку с клавиатуры; 3. вывести введённую строку на экран.

(рис. 3.17), (рис. 3.18)

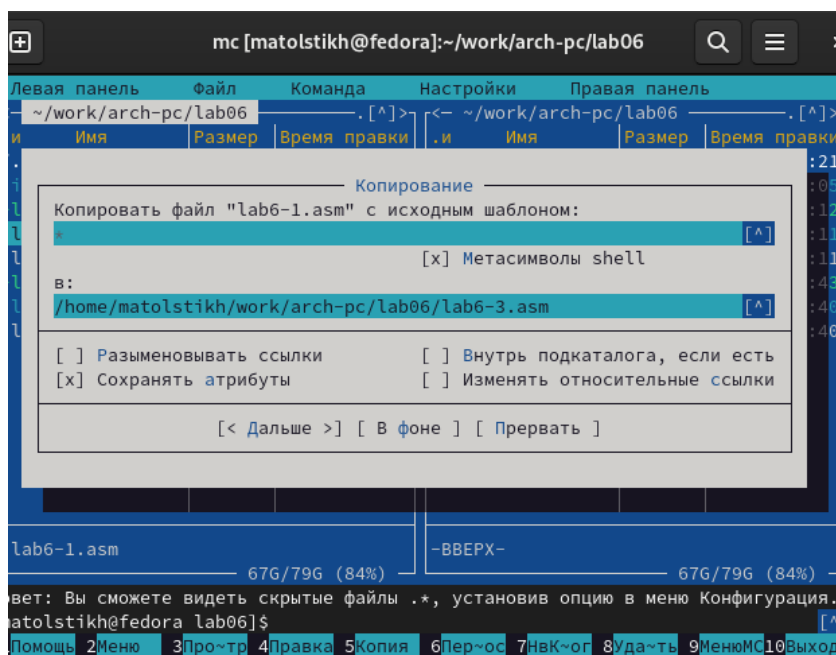
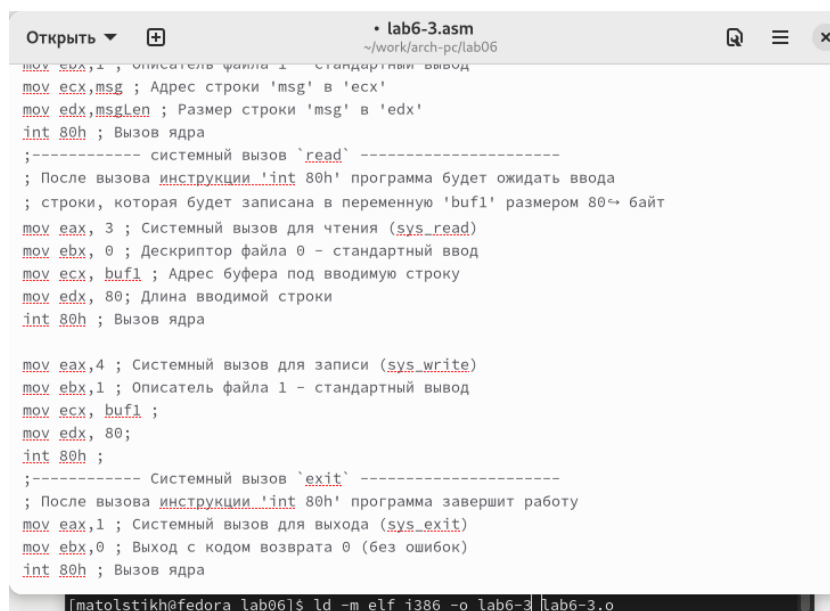


Рис. 3.17: lab6-3.asm



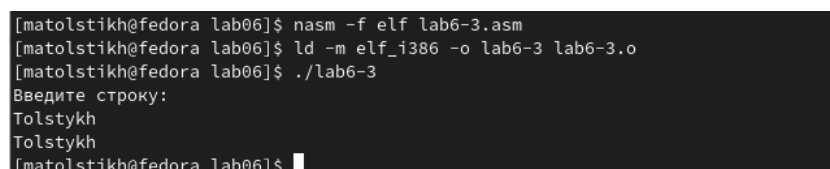
```
Открыть ▾ + • lab6-3.asm ~/work/arch-pc/lab06
mov eax,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать 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

mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx, buf1 ;
mov edx, 80 ;
int 80h ;
;----- 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

[matolstikh@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
```

Рис. 3.18: Изменения в программе

Получили исполняемый файл и проверили его работу. На приглашение ввести строку ввели свою фамилию. (рис. 3.19)



```
[matolstikh@fedora lab06]$ nasm -f elf lab6-3.asm
[matolstikh@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[matolstikh@fedora lab06]$ ./lab6-3
Введите строку:
Tolstykh
Tolstykh
[matolstikh@fedora lab06]$
```

Рис. 3.19: Проверка работы файла

Создали копию файла lab6-2.asm. Исправили текст программы с использованием подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: 1. вывести приглашение типа “Введите строку:”; 2. ввести строку с клавиатуры; 3. вывести введенную строку на экран.

(рис. 3.20), (рис. 3.21)

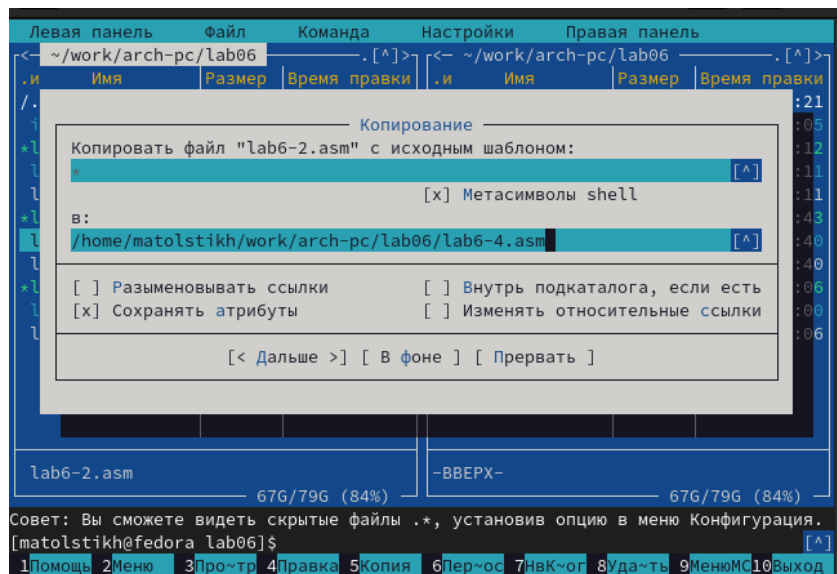


Рис. 3.20: lab6-4.asm

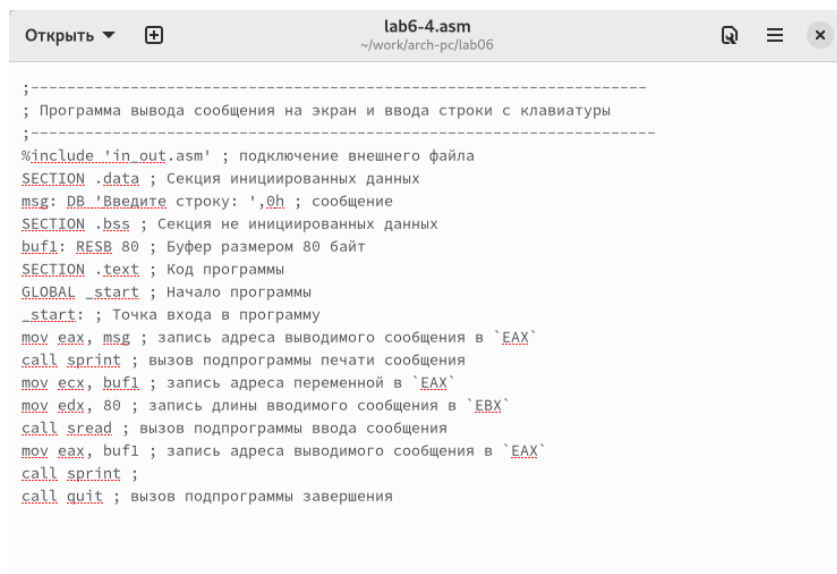


Рис. 3.21: Использование подпрограмм из внешнего файла in\_out.asm

Создали исполняемый файл и проверили его работу. (рис. 3.22)

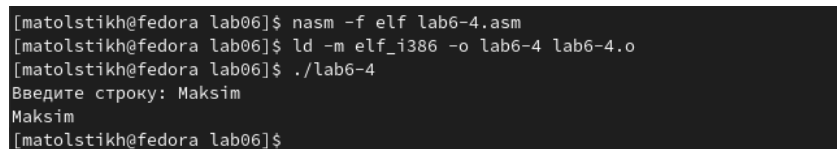


Рис. 3.22: Проверка работы исполняемого файла

## 4 Выводы

В ходе выполнения лабораторной работы были приобретены практически навыки работы в Midnight Commander и освоены инструкции языка ассемблера `mov` и `int`.