

# **Отчет по лабораторной работе №5**

**дисциплина: Архитектура компьютера**

Бондарь Татьяна Владимировна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Задания для самостоятельной работы</b>	<b>15</b>
<b>6</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

4.1	Переход в каталог ~/work/arch-rc . . . . .	8
4.2	Создание каталога и файла . . . . .	9
4.3	Открытие файла lab5-1.asm . . . . .	9
4.4	Ввод текста программы . . . . .	10
4.5	Проверка правильности сохранения файла . . . . .	10
4.6	Создание объектного и исполняемого файлов . . . . .	11
4.7	Запуск программы . . . . .	11
4.8	Каталог lab05 и каталог Downloads . . . . .	12
4.9	Копирование подключаемого файла in_out.asm в каталог lab05 . .	12
4.10	Копирование и переименование файла . . . . .	13
4.11	Исправление текста программы . . . . .	13
4.12	Запуск исполняемого файла lab5-2 . . . . .	14
4.13	Запуск файла lab5-2 с изменениями . . . . .	14
5.1	Внесение изменений в файл lab5-1-1.asm . . . . .	16
5.2	Запуск исполняемого файла . . . . .	17
5.3	Внесение изменений в файл lab5-2-2.asm . . . . .	17
5.4	Запуск исполняемого файла . . . . .	18

## **Список таблиц**

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

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.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использование подпрограмм из 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.
4. Создайте исполняемый файл и проверьте его работу.

### **3 Теоретическое введение**

# 4 Выполнение лабораторной работы

1. Открываю Midnight commander и перехожу в каталог ~/work/arch-pc, созданный при выполнении лабораторной работы №4.(рис. -fig. 4.1)

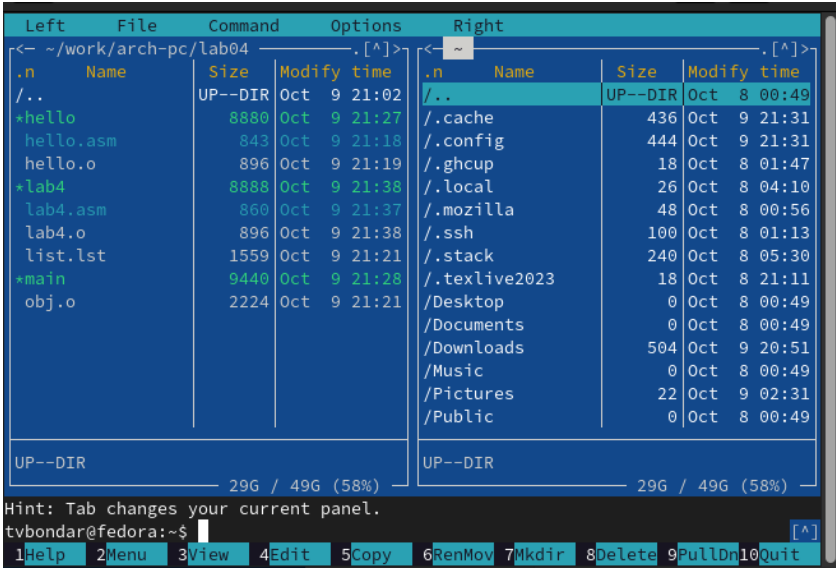


Рис. 4.1: Переход в каталог ~/work/arch-pc

2. Создаю каталог lab05 и в нем файл lab5-1.asm.(рис. -fig. 4.2)



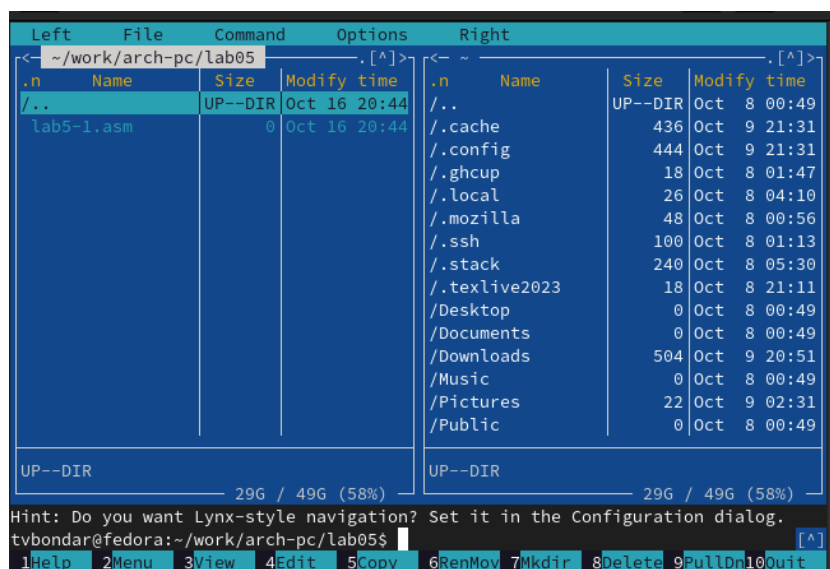


Рис. 4.2: Создание каталога и файла

- Открываю созданный файл с помощью встроенного редактора, начинаю вводить текст программы и сохраняю файл. (рис. -fig. 4.3, рис. -fig. 4.4)

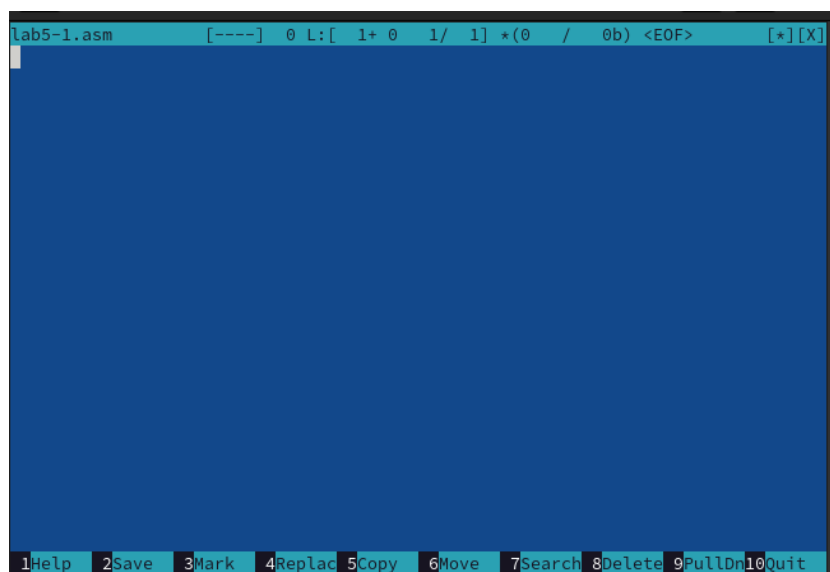


Рис. 4.3: Открытие файла lab5-1.asm

```

mc [tvbondar@fedora]:~/work/arch-pc/lab05
lab5-1.asm [-M--] 41 L:[ 1+ 0 1/ 32] *(74 /2130b) 1074 0x432 [*][X]
; Программа вывода сообщений на экран и ввода с клавиатуры
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
<-----><-----><-----><----->; символ перевода строки
msgLen: EQU $-msg ; длина переменной 'msg'

SECTION .bss ; секция неинициализированных 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
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit

```

Рис. 4.4: Ввод текста программы

4. Открываю файл для просмотра. Убеждаюсь что он содержит текст программы. (рис. -fig. 4.5)

```

tvbondar@fedora:~
GNU nano 7.2 /home/tvbondar/work/arch-pc/lab05/lab5-1.asm
; Программа вывода сообщений на экран и ввода с клавиатуры
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ;сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; длина переменной 'msg'

SECTION .bss ; секция неинициализированных 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
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

```

Рис. 4.5: Проверка правильности сохранения файла

5. Транслирую текст программы в объектный файл, а затем в исполняемый.

Запускаю исполняемый файл.(рис. -fig. 4.6, рис. -fig. 4.7)

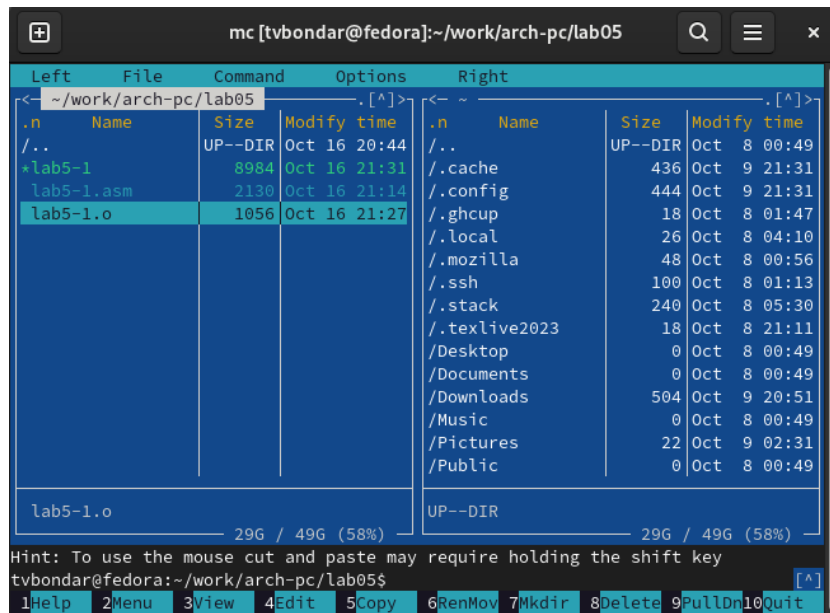


Рис. 4.6: Создание объектного и исполняемого файлов

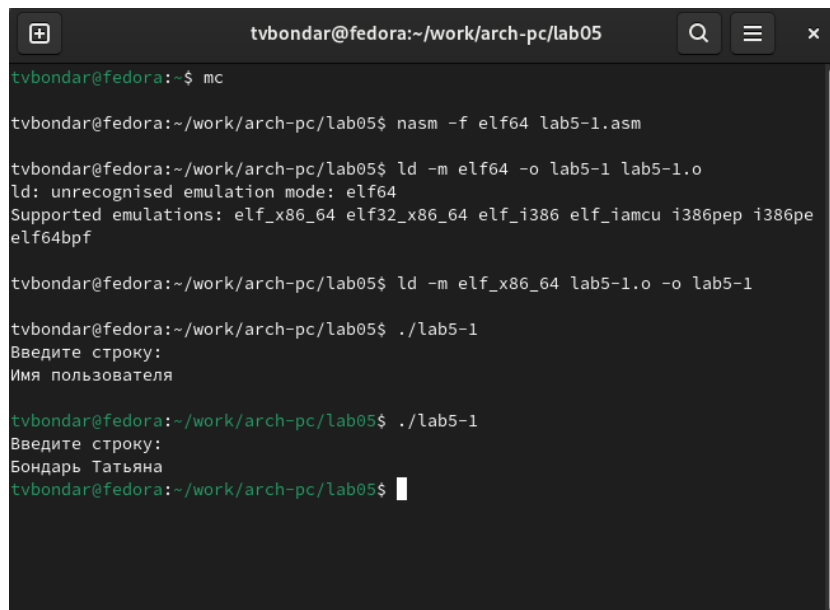


Рис. 4.7: Запуск программы

6. Скачиваю подключаемый файл in\_out.asm со страницы курса в ТУИС. Копирую его в каталог ~/work/arch-pc/lab05.(рис. -fig. 4.8, рис. -fig. 4.9)

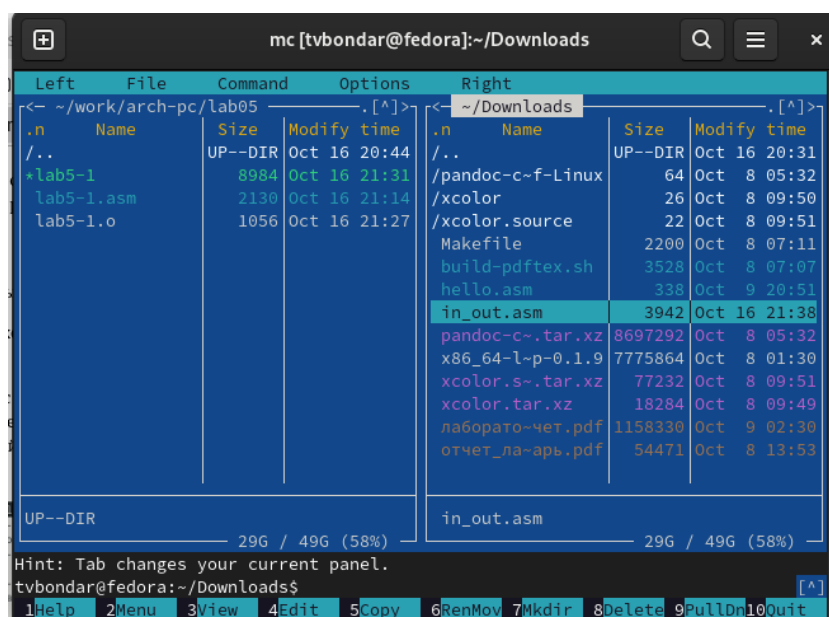


Рис. 4.8: Каталог lab05 и каталог Downloads

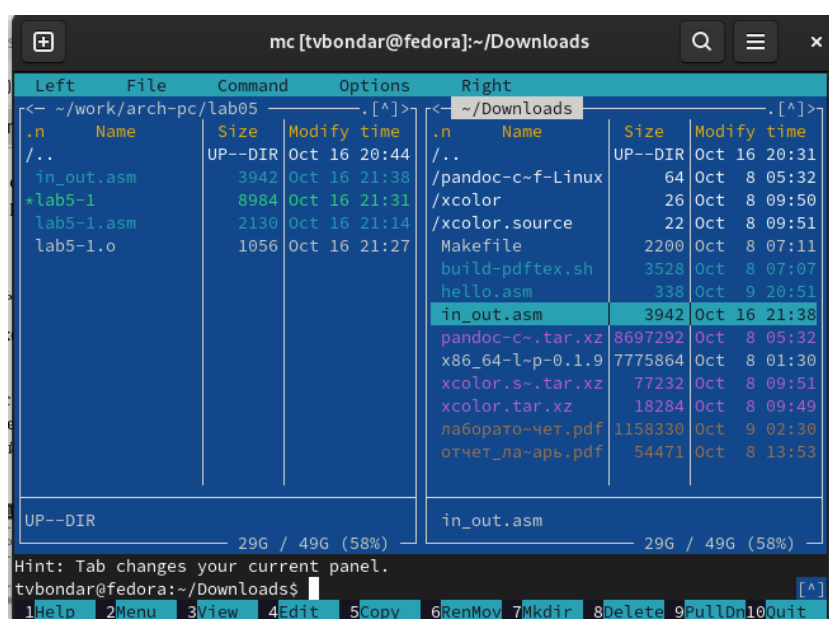


Рис. 4.9: КОпирование подключаемого файла in\_out.asm в каталог lab05

7. Создаю копию файла lab5-1.asm в том же каталоге и переименовываю ее в lab5-2.asm. (рис. -fig. 4.10)

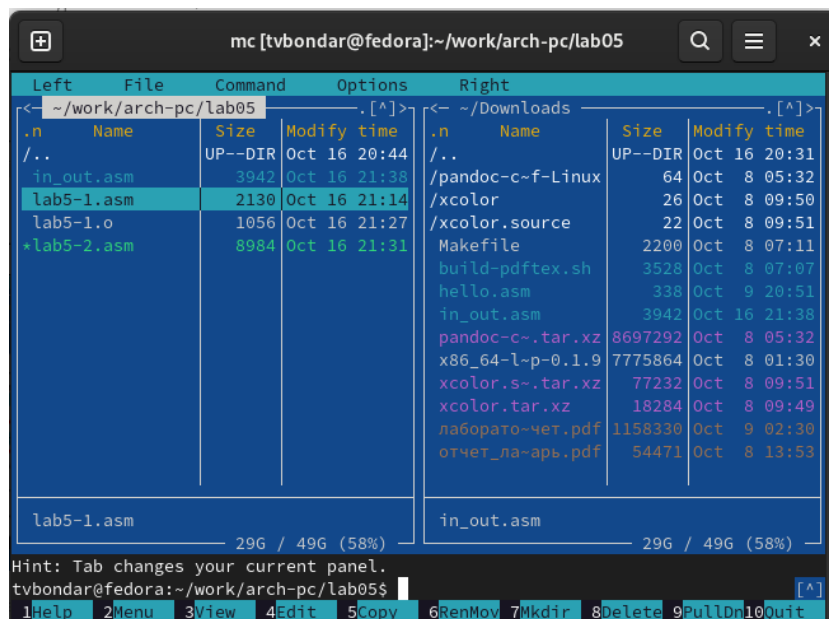


Рис. 4.10: Копирование и переименование файла

8. Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm. (рис. -fig. 4.11)

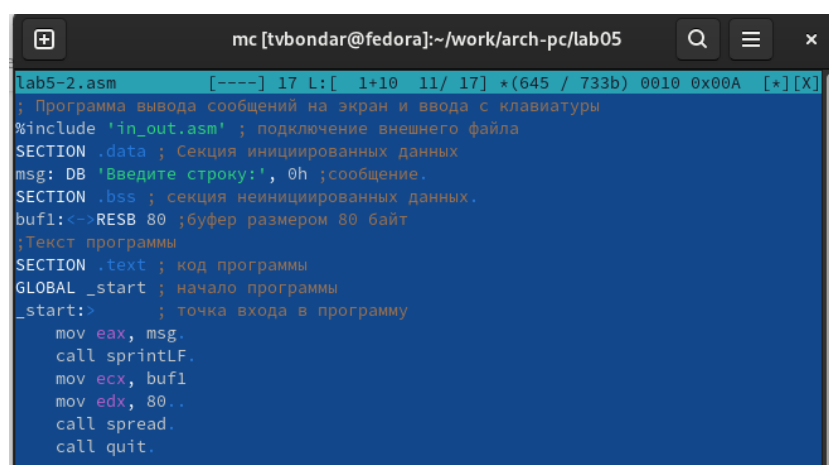


Рис. 4.11: Исправление текста программы

9. Компилирую исполняемый файл и проверяю его работу.(рис. -fig. 4.12)

```

tvbondar@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
tvbondar@fedora:~/work/arch-pc/lab05$ ld -m elf lab5-2.o -o lab5-2
ld: unrecognized emulation mode: elf
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe
elf64bpf
tvbondar@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
tvbondar@fedora:~/work/arch-pc/lab05$ ls
in_out.asm  lab5-1  lab5-1.asm  lab5-1.o  lab5-2  lab5-2.asm  lab5-2.o
tvbondar@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Бондарь Татьяна
tvbondar@fedora:~/work/arch-pc/lab05$

```

Рис. 4.12: Запуск исполняемого файла lab5-2

10. Заменяю в файле lab5-2.asm подпрограмму sprintLF на sprint. Проверяю работу программы. Ввод текста с клавиатуры теперь осуществляется на той же строке, где расположен текст, выводимый программой. (рис. -fig. 4.13)

```

tvbondar@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
tvbondar@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
tvbondar@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Бондарь Татьяна
tvbondar@fedora:~/work/arch-pc/lab05$

```

Рис. 4.13: Запуск файла lab5-2 с изменениями

## 5 Задания для самостоятельной работы

1. Создаю копию файла lab5-1.asm. Вношу 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. (рис. -fig. 5.1)

```

lab5-1-1.asm      [-M--] 13 L:
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 80h.

    mov eax, 3.
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    mov edx, buf1
    int 80h

    ....
    mov eax, 1
    mov ebx, 0
    int 80h.

```

Рис. 5.1: Внесение изменений в файл lab5-1-1.asm



2. Запускаю исполняемый файл lab5-1-1. Программа работает корректно.(рис. -fig. 5.2)

```
tvbondar@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
tvbondar@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1-1.o -o lab5-1-1
tvbondar@fedora:~/work/arch-pc/lab05$ ls
in_out.asm lab5-1 lab5-1-1 lab5-1-1.asm lab5-1-1.o lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o
tvbondar@fedora:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Бондарь Татьяна
Бондарь Татьяна
tvbondar@fedora:~/work/arch-pc/lab05$
```

Рис. 5.2: Запуск исполняемого файла

3. Создаю копию файла lab5-2.asm. Исправляю текст программы с использование 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. -fig. 5.3)

```
lab5-2-2.asm [-M--] 11 L:[ 1+17 18/ 20] *(735 / 750b) 00
; Программа вывода сообщений на экран и ввода с клавиатуры
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',0h ;сообщение.
SECTION .bss ; секция неиницированных данных.
buf1: RESB 80 ;буфер размером 80 байт
SECTION .text ; код программы
GLOBAL _start ; начало программы
_start: ; точка входа в программу
    mov eax, msg
    call sprint
    mov ecx, buf1
    mov edx, 80
    call sread
    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    int 80h
    call quit
```

Рис. 5.3: Внесение изменений в файл lab5-2-2.asm

4. Запускаю исполняемый файл lab5-2-2. Программа работает корректно. (рис. -fig. 5.4)

```
cvbondar@fedora: ~/work/arch-pc/lab05$ nasm -f elf lab5-2-2.asm
cvbondar@fedora: ~/work/arch-pc/lab05$ ld -m elf_386 lab5-2-2.o -o lab5-2-2
cvbondar@fedora: ~/work/arch-pc/lab05$ ls
in_out.asm lab5-1 lab5-1-1 lab5-1-1.asm lab5-1-1.o lab5-1.asm lab5-1.o lab5-2 lab5-2-2 lab5-2-2.asm lab5-2-2.o lab5-2.asm lab5-2.o
cvbondar@fedora: ~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку:Бондарь Татьяна
Бондарь Татьяна
cvbondar@fedora: ~/work/arch-pc/lab05$
```

Рис. 5.4: Запуск исполняемого файла

## 6 Выводы

В результате выполнения лабораторной работы я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.