

Отчёт по лабораторной работе №6

Компьютерные науки и технология программирования

Сячинова Ксения Ивановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задания для самостоятельной работы.	14
4	Выводы	17

Список иллюстраций

2.1	Команда для открытия МС	6
2.2	Переход в папку	6
2.3	Переход в папку	7
2.4	Создание папки	7
2.5	Создание файла	7
2.6	Открытие файла	8
2.7	Текст программы	8
2.8	Проверка файла	9
2.9	Запуск программы	9
2.10	Скачивание файла	10
2.11	Сохранение файла	10
2.12	Результат копирования	11
2.13	Копирование файла	11
2.14	Текст новой программы	12
2.15	Результат	12
2.16	Изменение программы	12
2.17	Результат	13
3.1	Копирование файла	14
3.2	Изменение файла	15
3.3	Выполнение программы	15
3.4	Копирование файла	16
3.5	Изменение команды	16
3.6	Результат команды	16

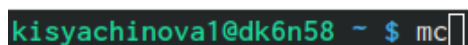
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

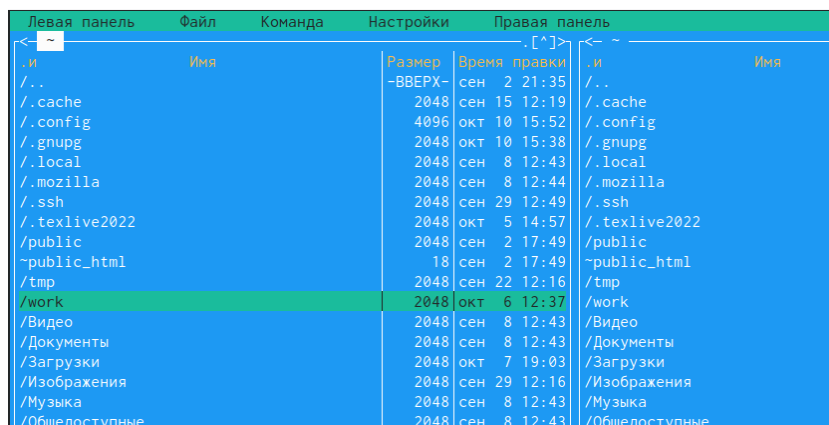
1. Открываем Midnight Commander. (рис. 2.1)



```
kisyachinova1@dk6n58 ~ $ mc
```

Рис. 2.1: Команда для открытия МС

2. После этого переходим в каталог ‘~/work/arch-рс’, который создали при выполнении лабораторной №5.(рис. 2.2),(рис. 2.3)



Левая панель	Файл	Команда	Настройки	Правая панель
< ~				> ^>
	Имя	Размер	Время	Имя
..		-BBERX-	сен 2 21:35	..
/.cache		2048	сен 15 12:19	/.cache
/.config		4096	окт 10 15:52	/.config
/.gnupg		2048	окт 10 15:38	/.gnupg
/.local		2048	сен 8 12:43	/.local
/.mozilla		2048	сен 8 12:44	/.mozilla
/.ssh		2048	сен 29 12:49	/.ssh
/.texlive2022		2048	окт 5 14:57	/.texlive2022
/public		2048	сен 2 17:49	/public
~public_html		18	сен 2 17:49	~public_html
/tmp		2048	сен 22 12:16	/tmp
/work		2048	окт 6 12:37	/work
/Видео		2048	сен 8 12:43	/Видео
/Документы		2048	сен 8 12:43	/Документы
/Загрузки		2048	окт 7 19:03	/Загрузки
/Изображения		2048	сен 29 12:16	/Изображения
/Музыка		2048	сен 8 12:43	/Музыка
/Общедоступные		2048	сен 8 12:43	/Общедоступные

Рис. 2.2: Переход в папку

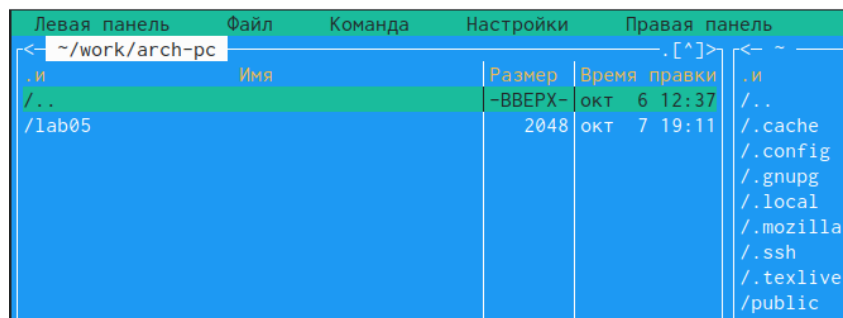


Рис. 2.3: Переход в папку

- После этого с помощью клавиши 'F7' создаём папку 'lab06' и переходим в неё.(рис. 2.4)

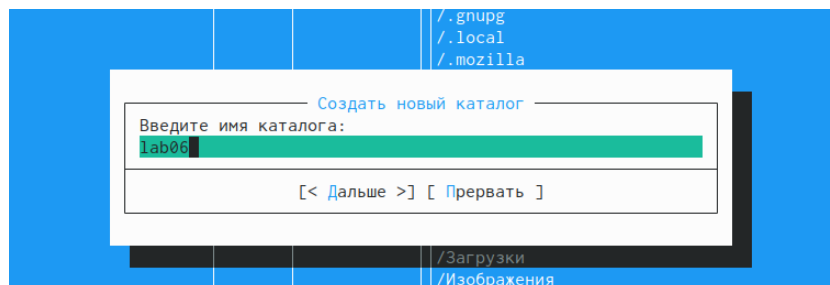


Рис. 2.4: Создание папки

- Используя строку ввода создаём файл lab6-1.asm с помощью команды 'touch'. (рис. 2.5)

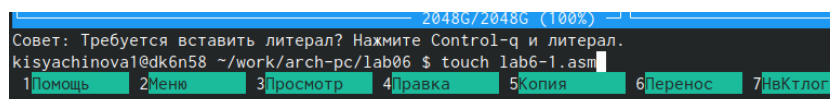


Рис. 2.5: Создание файла

- После этого с помощью клавиши 'F4' открываем созданный файл для редактирования. У меня редактор 'mcedit'. (рис. 2.6)

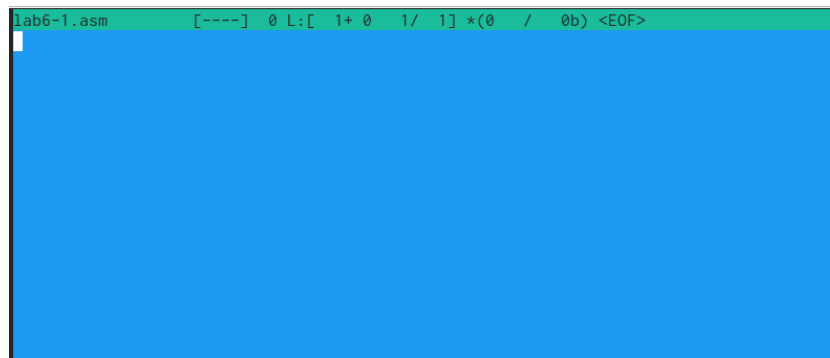


Рис. 2.6: Открытие файла

6. Вводим текст необходимой программы. Затем, сохраняем файл с помощью клавиши 'F2' и выходим из редактора с помощью клавиши 'F10'.(рис. 2.7)

```
lab6-1.asm      [-M--] 13 L:[ 1+ 8 9/ 26][*][X]
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

    ....

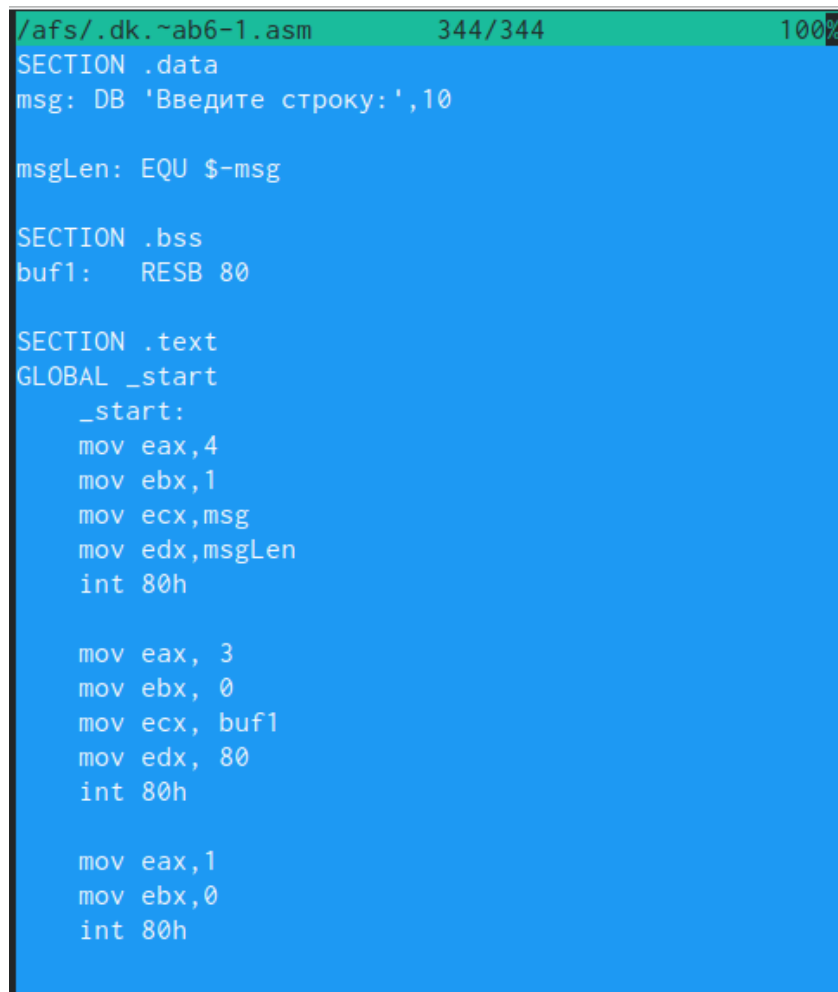
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

    ....

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

Рис. 2.7: Текст программы

7. С помощью клавиши 'F3', которая позволяет просмотреть файл, проверяем наличие текста.(рис. 2.8)



```
/afs/.dk.~ab6-1.asm 344/344 100%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

SECTION .bss
buf1:  RESB 80

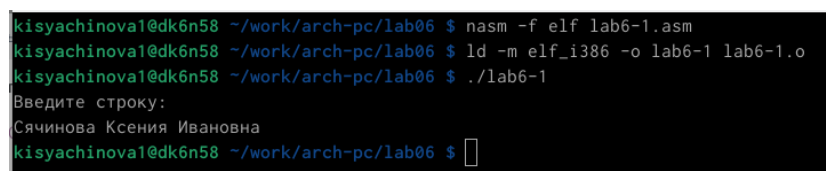
SECTION .text
GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h

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

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

Рис. 2.8: Проверка файла

8. После этого транслируем текст нашей программы в объектный файл, выполняем компоновку и запускаем. Результат: программа выводит “Введите строку”, вводим ФИО. (рис. 2.9)



```
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-1
Введите строку:
Сячинова Ксения Ивановна
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $
```

Рис. 2.9: Запуск программы

9. Скачиваем файл 'in_out.asm' с ТУИСа. (рис. 2.10)

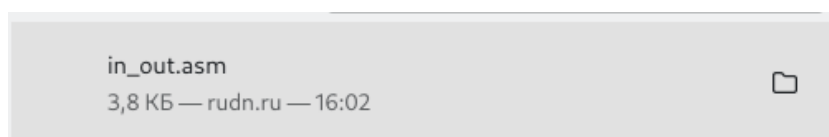


Рис. 2.10: Скачивание файла

10. Подключаемый файл должен лежать в том же каталоге, что и файл с программой, в которой он используется. С помощью 'mc' перемещаем файл 'in_out.asm' Копируем с помощью клавиши 'F5'. (рис. 2.11), (рис. 2.12)

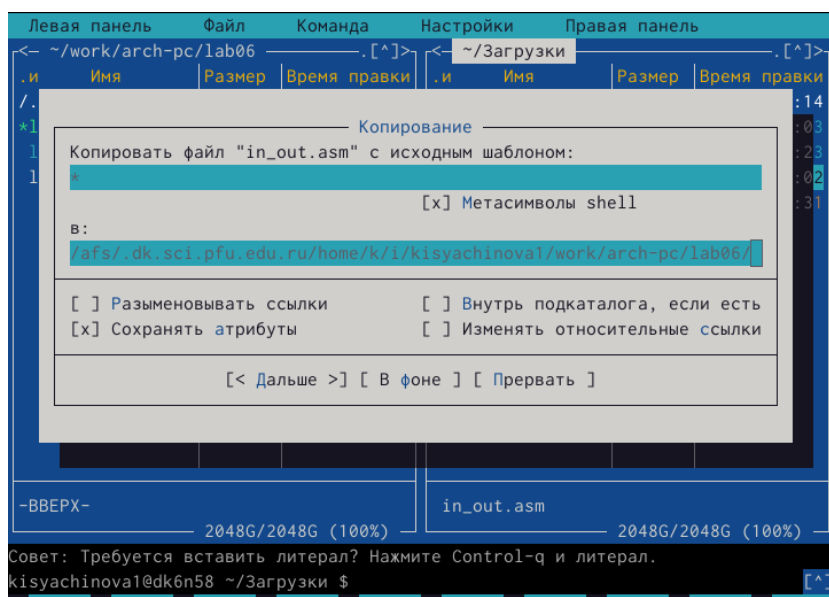


Рис. 2.11: Сохранение файла

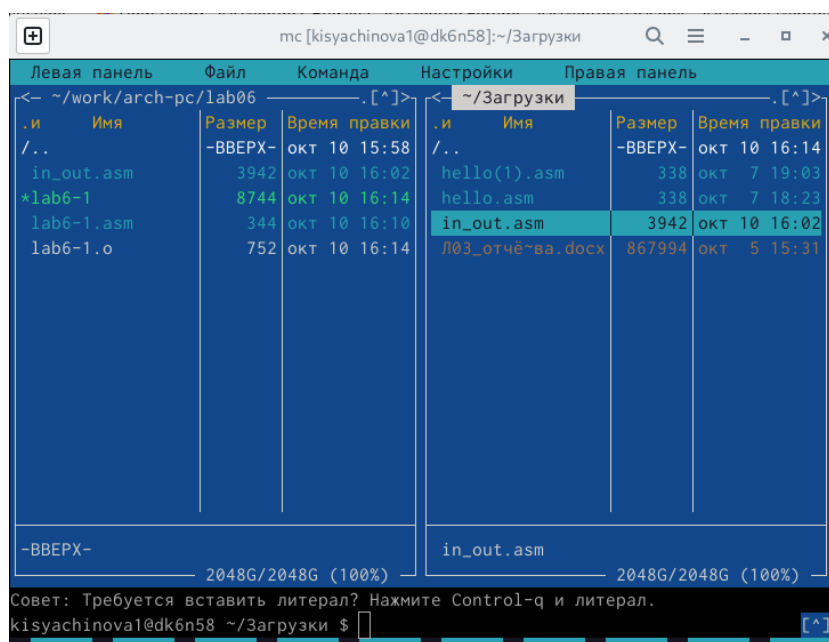


Рис. 2.12: Результат копирования

11. Создаём копию файла 'lab6-1.asm' с именем 'lab6-2.asm' с помощью клавиши 'F6'. (рис. 2.13)

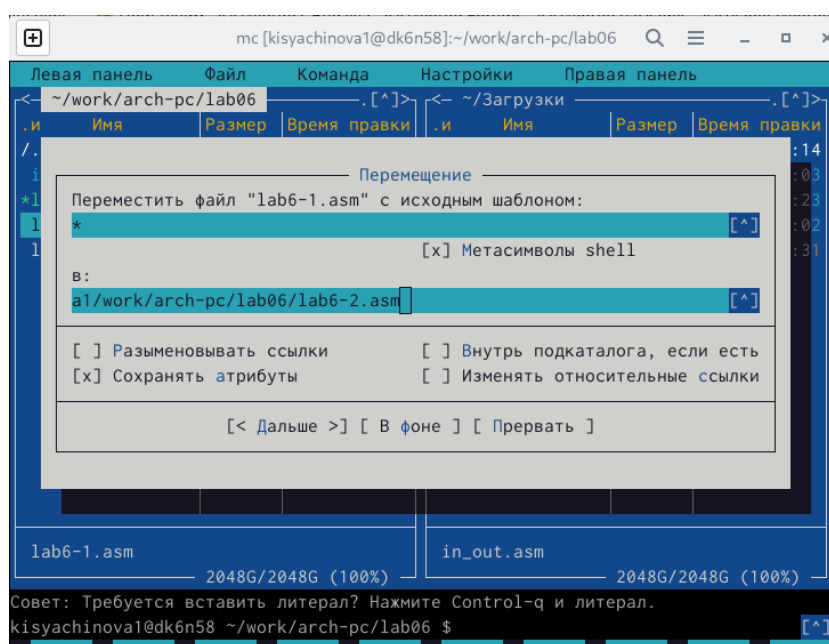


Рис. 2.13: Копирование файла

12. Затем исправляем текст файла, и проверяем работу. (рис. 2.14),(рис. 2.15)

```

lab6-2.asm      [----] 11 L:[ 1+10 11
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку:',0h

SECTION>.bss
buf1:<->RESB 80

SECTION .text
GLOBAL _start
_start:
....
    mov eax, msg
    call sprintfLF
....
    mov ecx, buf1
    mov edx, 80
    call sread
....
    call quit

```

Рис. 2.14: Текст новой программы

```

kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:
Сячинова Ксения Ивановна
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $

```

Рис. 2.15: Результат

13. Изменим в тексте программы 'sprintfLF' на 'sprintf'.(рис. 2.16), (рис. 2.17)

```

kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:Сячинова Ксения Ивановна
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $

```

Рис. 2.16: Изменение программы

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:Сячинова Ксения Ивановна
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $
```

Рис. 2.17: Результат

Разница в том, что команда 'sprintLF' позволяет нам делать ввод с клавиатуры в новой строке, в то время как команда 'sprint' делает это на одной строке

3 Задания для самостоятельной работы.

1. Создаём копию файла lab6-1.asm. (рис. 3.1)

.и	Имя	Размер	Время правки	.и	Имя
/..		-ВВЕРХ-	окт 10 15:58	/..	
	in_out.asm	3942	окт 10 16:02		in_out.asm
	*lab6-1	8744	окт 10 16:51		*lab6-1
	lab6-1.1asm	349	окт 10 16:51		lab6-1.1asm
	lab6-1.asm	349	окт 10 16:51		lab6-1.asm
	lab6-1.o	752	окт 10 16:51		lab6-1.o
	*lab6-2	9092	окт 10 16:40		*lab6-2
	lab6-2.asm	258	окт 10 16:40		lab6-2.asm
	lab6-2.o	1312	окт 10 16:40		lab6-2.o

Рис. 3.1: Копирование файла

Затем вносим изменение в программу, чтобы она работала по следующему режиму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.(рис. 3.2)

```

1 SECTION .data
2 msg: DB 'Введите строку:',10
3
4 msgLen: EQU $-msg
5
6 SECTION .bss
7 buf1: RESB 80
8
9 SECTION .text
10 GLOBAL _start
11 _start:
12
13     mov eax,4
14     mov ebx,1
15     mov ecx,msg
16     mov edx,msgLen
17     int 80h
18
19     mov eax, 3
20     mov ebx, 0
21     mov ecx, buf1
22     mov edx, 80
23     int 80h
24
25     mov eax, 4
26     mov ebx, 1
27     mov ecx, buf1
28     int 80h
29
30     mov eax,1
31     mov ebx,0
32     int 80h

```

Рис. 3.2: Изменение файла

2. После этого транслируем текст нашей программы в объектный файл, выполняем компоновку и запускаем. Программа просит ввести строку и выводит её. (рис. 3.3)

```

kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.1.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1.1 lab6-1.1.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-1.1
Введите строку:
Сячинова Ксения
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $

```

Рис. 3.3: Выполнение программы

3. Аналогично создаём копию файла lab6-2.asm (рис. 3.4)

lab6-1.1.asm	415	ОКТ 13 12:37	
lab6-1.1.o	784	ОКТ 13 12:43	
lab6-1.asm	349	ОКТ 10 16:51	
lab6-1.o	784	ОКТ 13 12:41	
*lab6-2	9092	ОКТ 10 16:40	*
lab6-2.2.asm	258	ОКТ 10 16:40	
lab6-2.asm	258	ОКТ 10 16:40	
lab6-2.o	1312	ОКТ 10 16:40	

Рис. 3.4: Копирование файла

Изменяем файл, чтобы он работал по такому же алгоритму, при этом используем подпрограмму из внешнего файла in_out.asm.(рис. 3.5)

```
lab6-2.2.asm [----] 17 L: [ 2+18 20/ 23]
SECTION .data
msg: DB 'Введите строку:',0h

SECTION .bss
buf1: <-> RESB 80

SECTION .text
GLOBAL _start
_start:
....
    mov eax, msg
    call sprintf

....
    mov ecx, buf1
    mov edx, 80
    call sread

....
    mov eax, buf1
    call sprint

....
    call quit
```

Рис. 3.5: Изменение команды

4. Компилируем и запускаем файл.(рис. 3.6)

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:Сячинова Ксения Ивановна
kisyachinova1@dk6n58 ~/work/arch-pc/lab06 $
```

Рис. 3.6: Результат команды

4 Выводы

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