

# Российский университет дружбы народов имени Патриса Лумумбы

## Отчёт по лабораторной работе №5

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

Выполнил: Чубаев Кирилл Евгеньевич, НММбд-04-24

### Содержание

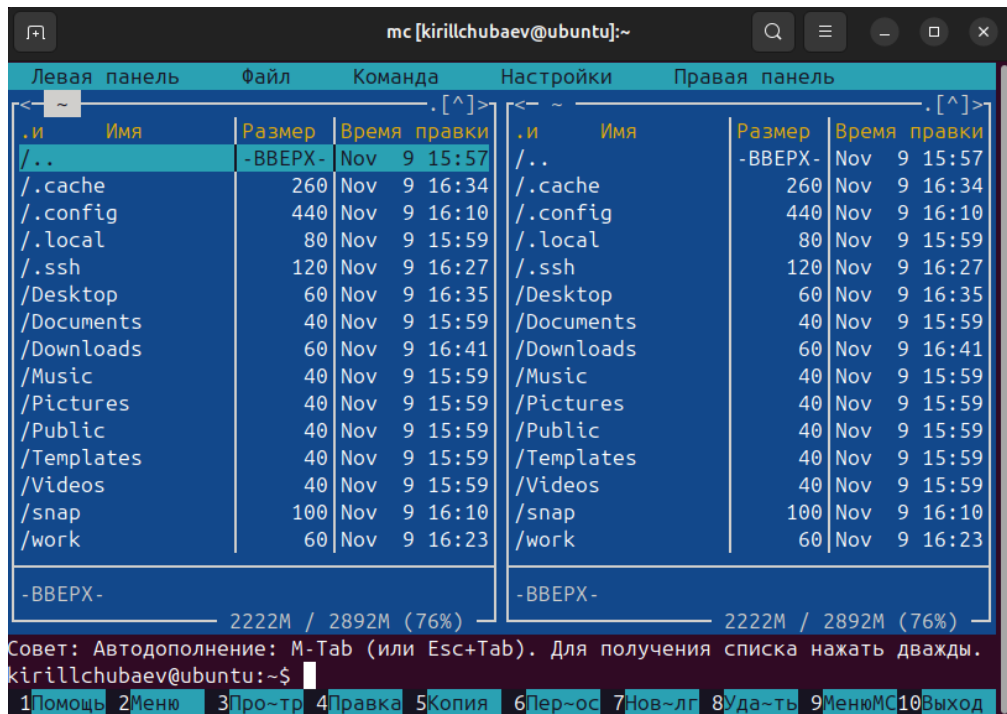
Цель работы.....	1
Ход выполнения лабораторной работы: .....	2
Выполнение самостоятельной работы: .....	8
Вывод: .....	10
Список литературы: .....	11

### Цель работы

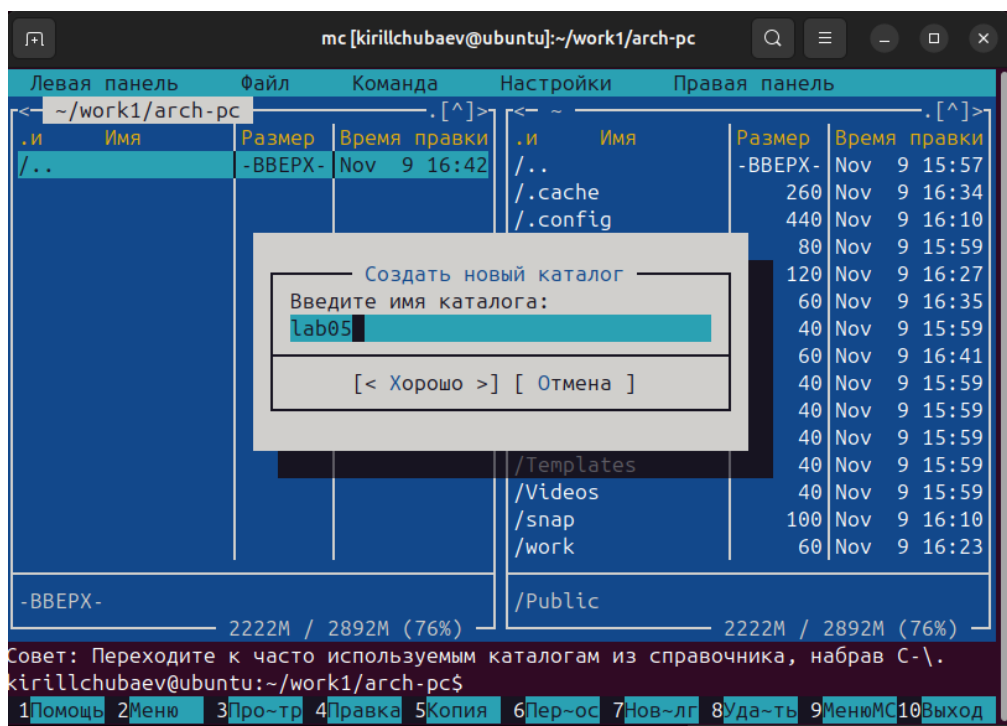
Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, а также освоение инструкций языка ассемблера mov и int.

## Ход выполнения лабораторной работы:

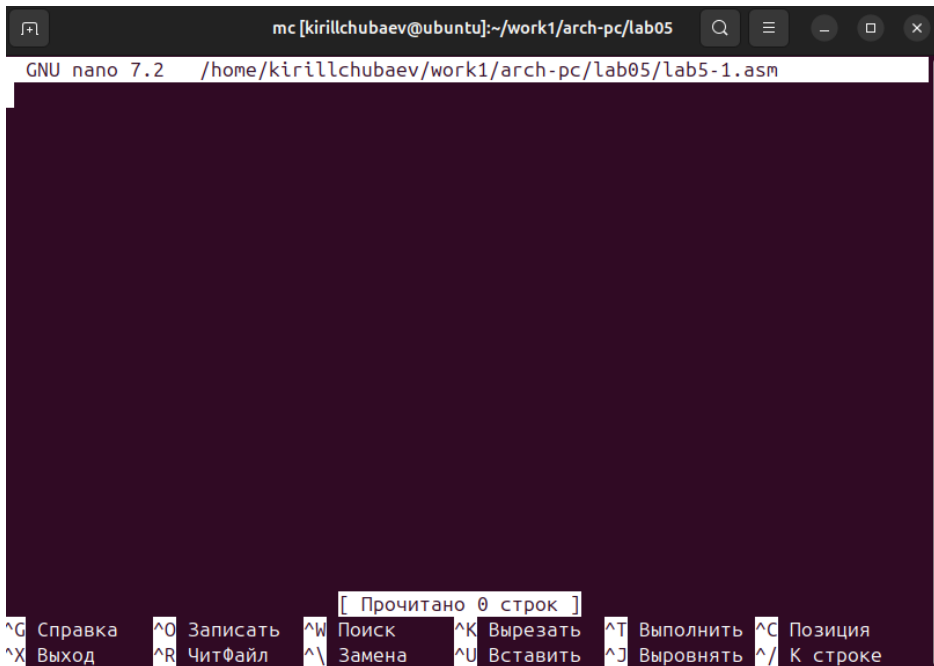
1. Я ввел команду `mc` в терминале. Открылся Midnight Commander:



2. Перешёл в каталог `~/work1/arch-pc`. С помощью клавиши `F7` я создал папку `lab05` и перешел в неё:

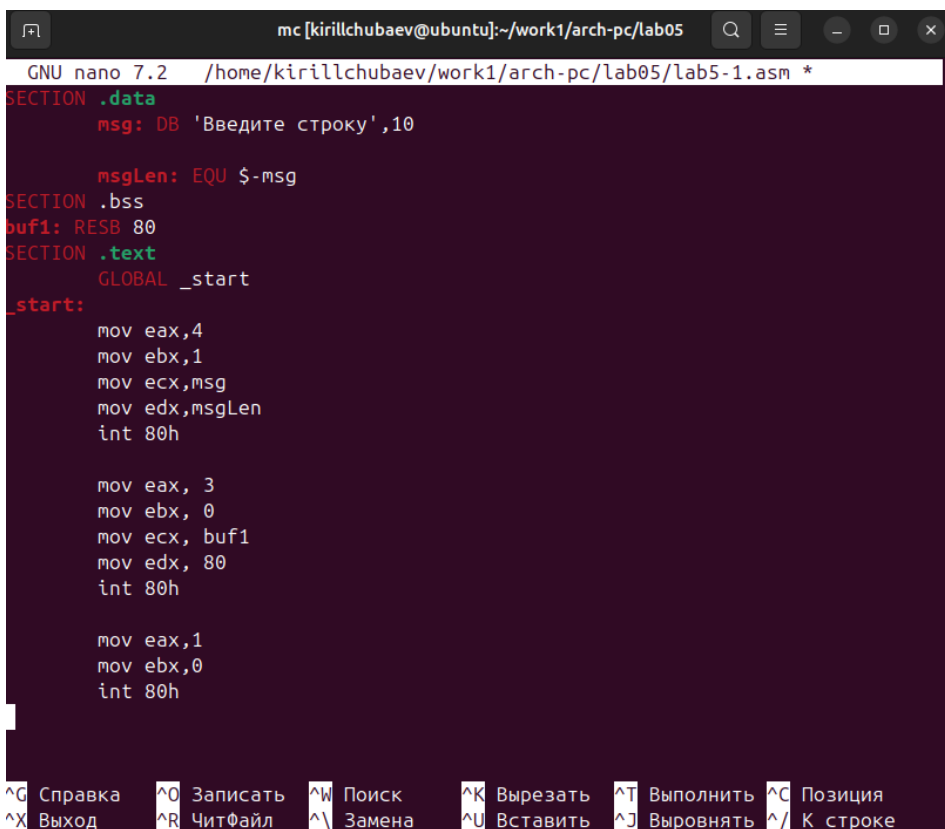


3. Далее создал файл lab5-1.asm с помощью команды touch. С помощью клавиши F4 я открыл созданный файл для редактирования во встроенном редакторе nano:



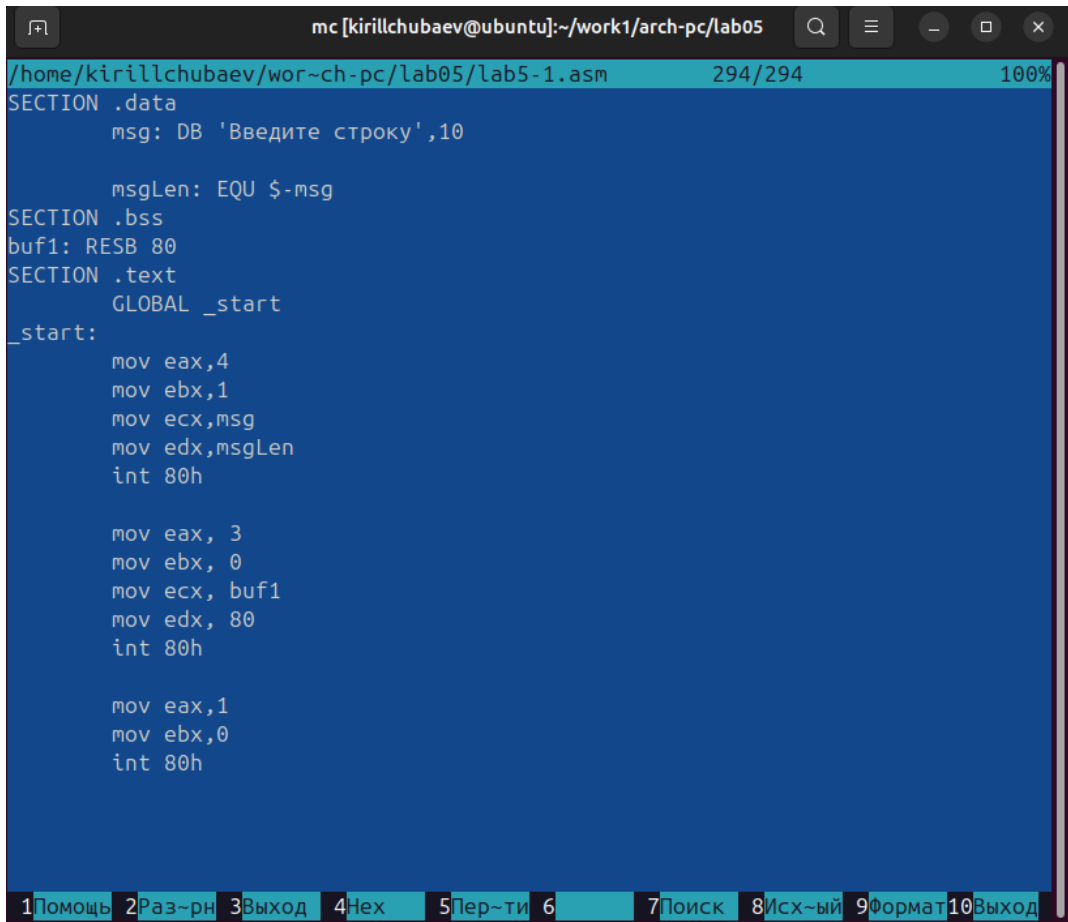
The screenshot shows the nano text editor interface. The title bar indicates the user is 'mc [kirillchubaev@ubuntu]:~/work1/arch-pc/lab05'. The editor is open to the file '/home/kirillchubaev/work1/arch-pc/lab05/lab5-1.asm'. The main editing area is empty. The status bar at the bottom shows '[ Прочитано 0 строк ]' and a list of keyboard shortcuts: ^G Справка, ^O Записать, ^W Поиск, ^K Вырезать, ^T Выполнить, ^C Позиция, ^X Выход, ^R ЧитФайл, ^\ Замена, ^U Вставить, ^J Выводить, ^\_ К строке.

4. Далее я ввёл текст программы из листинга 5.1, сохранил изменения и закрыл файл:



The screenshot shows the nano text editor with the assembly code from listing 5.1 entered. The code is as follows:  
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  
The status bar at the bottom shows the same keyboard shortcuts as in the previous screenshot.

5. С помощью клавиши F3 я просмотрел файл lab5-1.asm



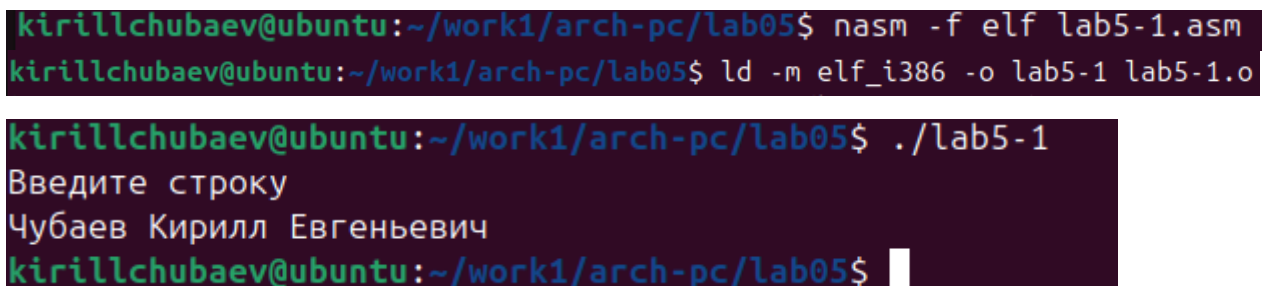
```
mc [kirillchubaev@ubuntu]:~/work1/arch-pc/lab05
/home/kirillchubaev/work1/arch-pc/lab05/lab5-1.asm 294/294 100%
SECTION .data
    msg: DB 'Введите строку',10

    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

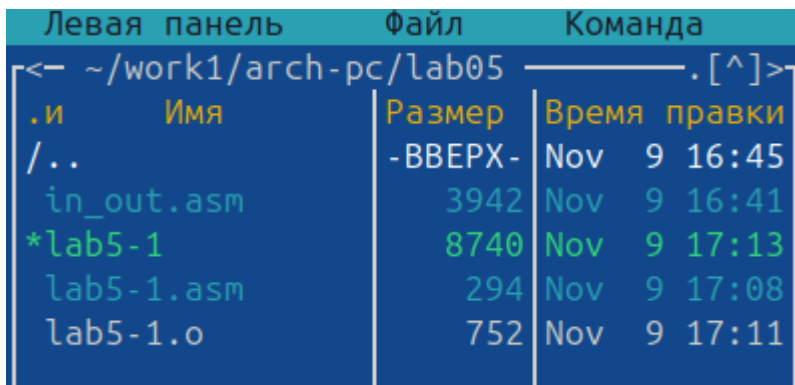
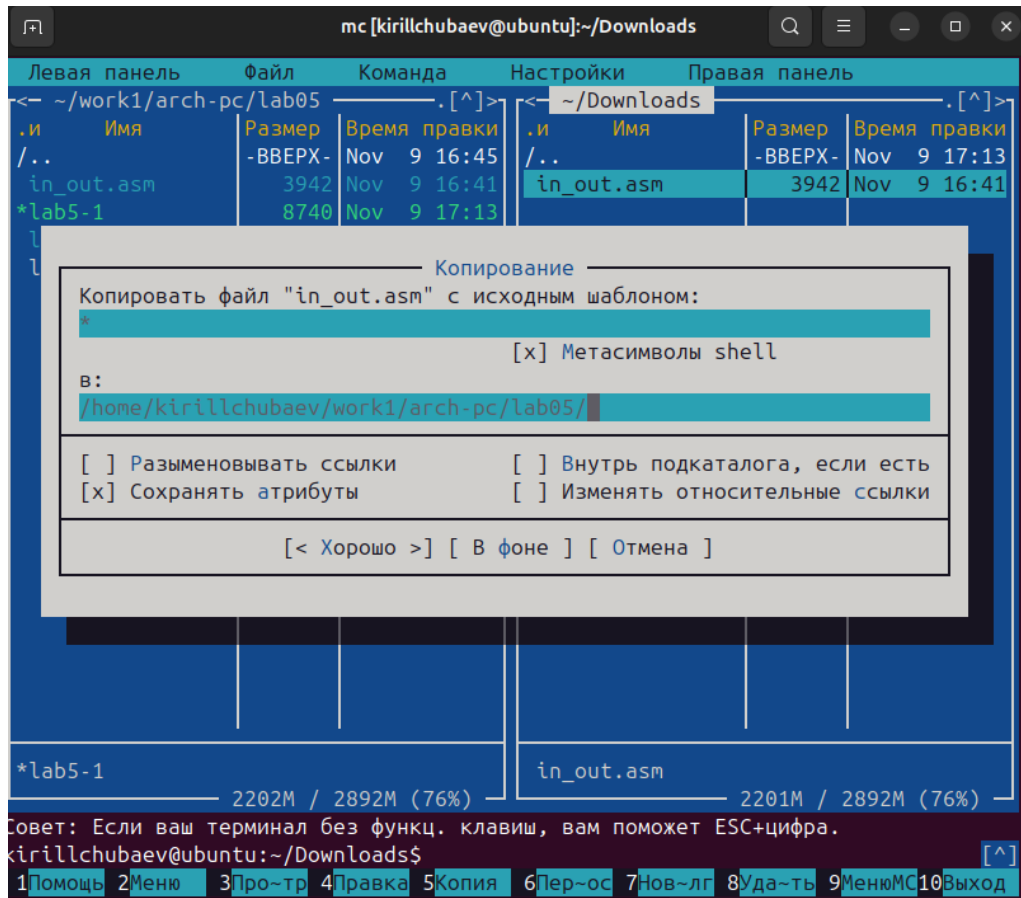
    mov eax,1
    mov ebx,0
    int 80h
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

6. Я оттранслировал текст программы в объектный файл. Далее выполнил компоновку объектного файла и запустил получившийся исполняемый файл. Программа вывела строку "Введите строку" и ожидала ввода с клавиатуры. На запрос я ввёл свои ФИО:

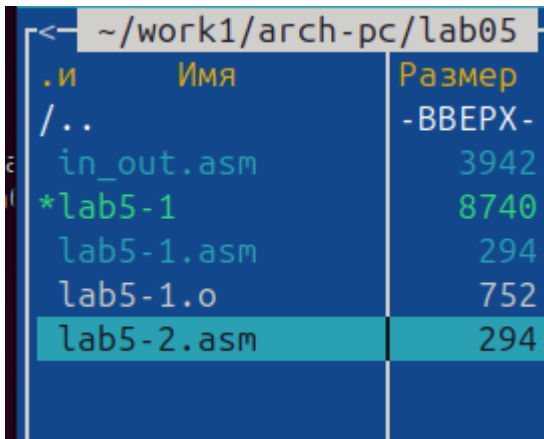


```
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ nasm -f elf lab5-1.asm
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ./lab5-1
Введите строку
Чубаев Кирилл Евгеньевич
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$
```

7. Я скачал файл in\_out.asm со страницы курса в ТУИС. Далее я скопировал загруженный файл в каталог с файлом lab5-1.asm с помощью клавиши F5:

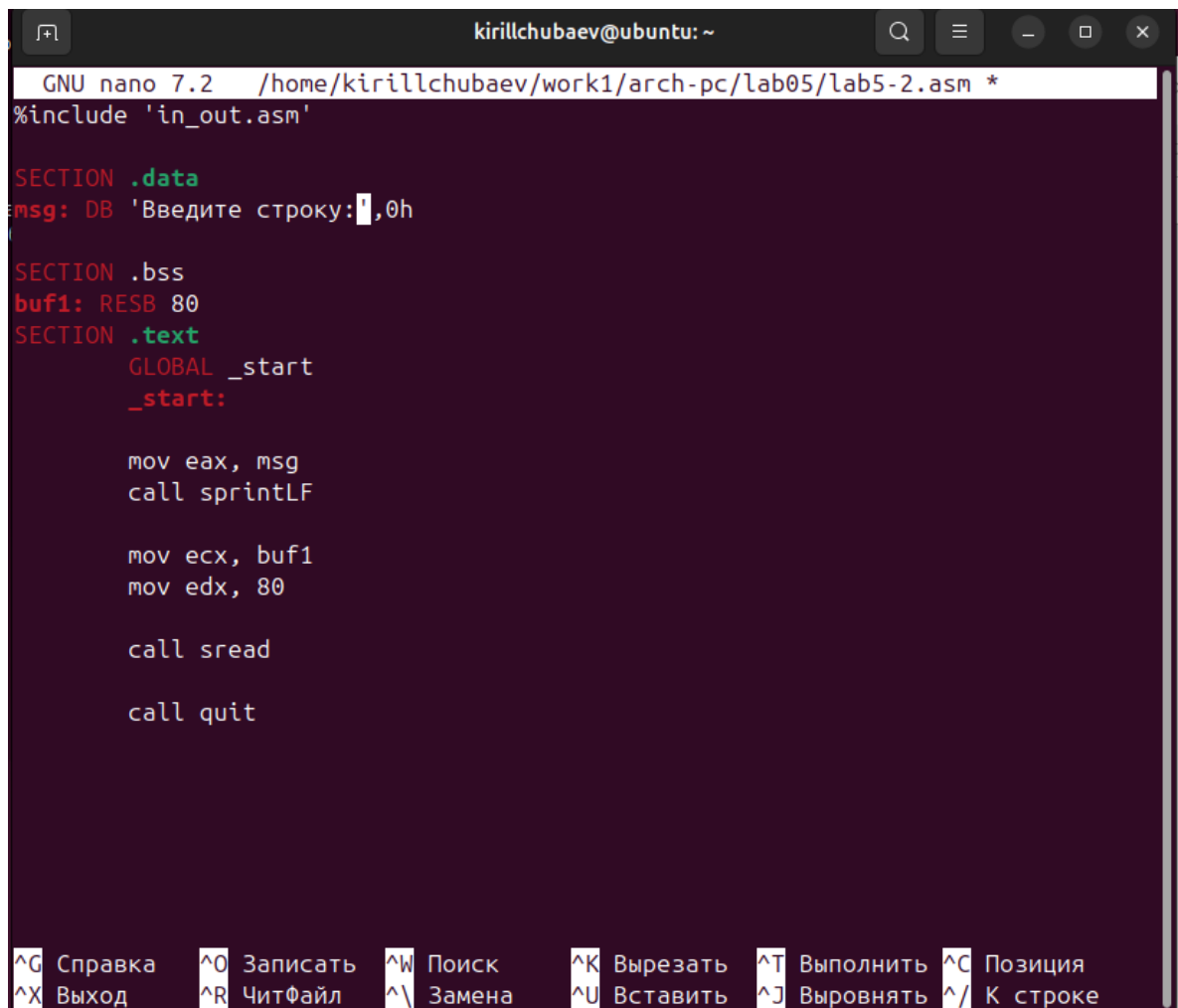


8. С помощью клавиши F6 я создал копию файла lab5-1.asm под названием lab5-2.asm:



Имя	Размер
./..	-ВВЕРХ-
in_out.asm	3942
*lab5-1	8740
lab5-1.asm	294
lab5-1.o	752
lab5-2.asm	294

9. Далее исправил текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2. Создал исполняемый файл и проверил его работу:



```
GNU nano 7.2 /home/kirillchubaev/work1/arch-pc/lab05/lab5-2.asm *
#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
```

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

```

kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ nasm -f elf lab5-2.asm
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ./lab5-2
Введите строку:
Чубаев Кирилл Евгеньевич
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$

```

10. В файле lab5-2.asm я заменил подпрограмму sprintLF на sprint. Далее создал исполняемый файл и проверил его работу.

```

#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 sprint

    mov ecx, buf1
    mov edx, 80

    call sread

    call quit

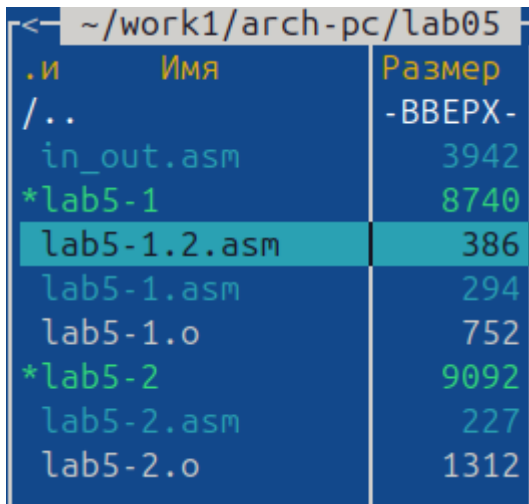
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ nasm -f elf lab5-2.asm
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ./lab5-2
Введите строку: Чубаев Кирилл Евгеньевич

```

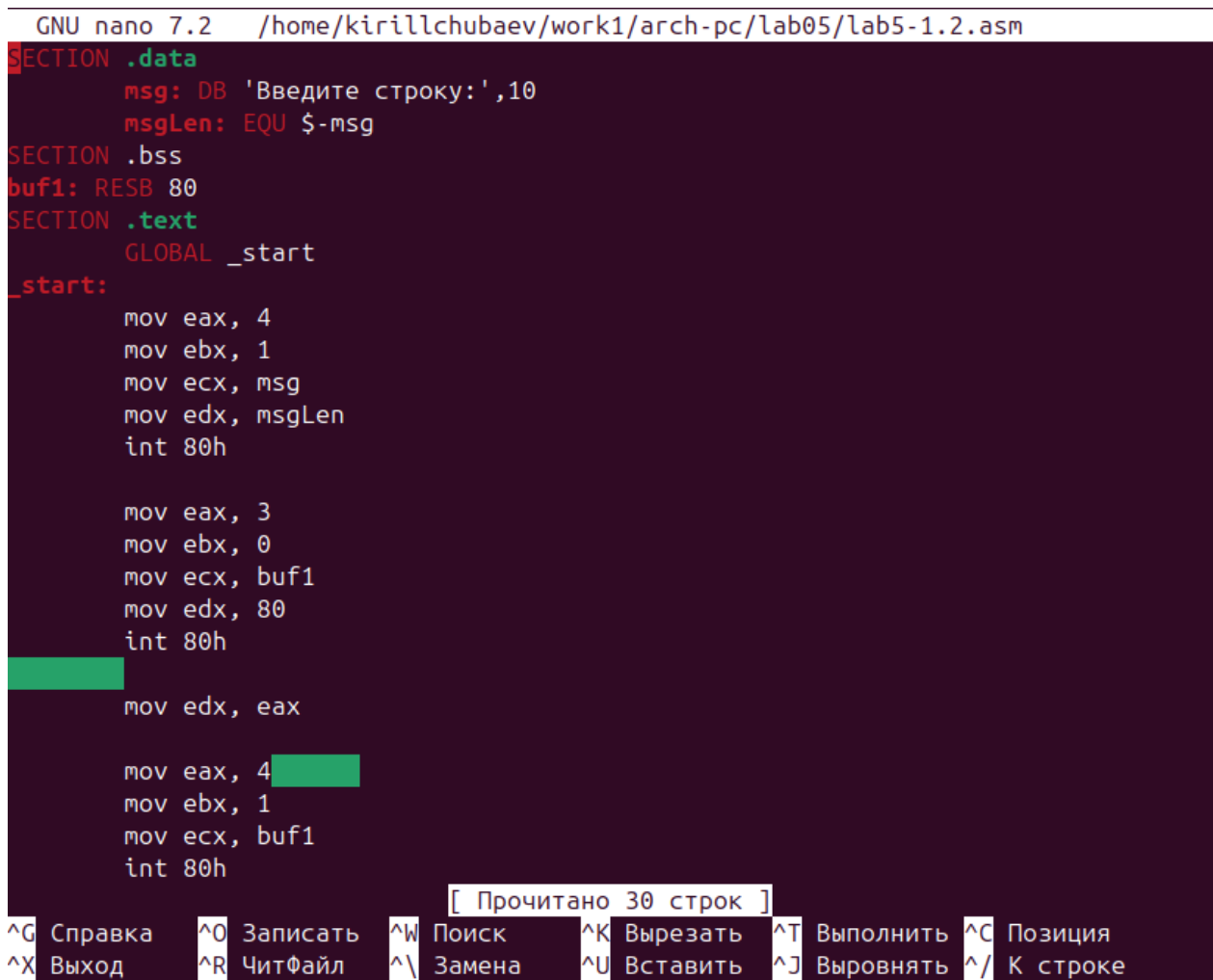
Разница выполнения программы в том, что подпрограмма sprint ожидает ответа от пользователя на той же строке, где и было выведено "Введите строку", а подпрограмма sprintLF ожидает ответа от пользователя уже на следующей строке.

## Выполнение самостоятельной работы:

1. Я создал копию файла lab5-1.asm под названием lab5-1.2.asm. Далее я внёс изменения в программу, так чтобы она работала по алгоритму, указанному в задании:



.и	Имя	Размер
	/..	-ВВЕРХ-
	in_out.asm	3942
*	lab5-1	8740
	lab5-1.2.asm	386
	lab5-1.asm	294
	lab5-1.o	752
*	lab5-2	9092
	lab5-2.asm	227
	lab5-2.o	1312



```
GNU nano 7.2 /home/kirillchubaev/work1/arch-pc/lab05/lab5-1.2.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 80h

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

    mov edx, eax

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

[ Прочитано 30 строк ]

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



2. Далее я получил исполняемый файл и проверил его работу. На приглашение ввести строку я ввёл свою фамилию. Измененная программа вывела в терминал ранее введенную мною строку:

```
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ nasm -f elf lab5-1.2.asm
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ld -m elf_i386 -o lab5-1.2 lab5-1.2.o

kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ./lab5-1.2
Введите строку:
Чубаев
Чубаев
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$
```

3. По аналогичному принципу я создал копию файла lab5-2.asm под названием lab5-2.2.asm. Далее внёс необходимые изменения в файл, чтобы он выводил введенную ранее строку в терминал:

~/work1/arch-pc/lab05		
.и	Имя	Размер
-ВВЕРХ-		
	/..	
	in_out.asm	3942
*	lab5-1	8740
*	lab5-1.2	8748
	lab5-1.2.asm	366
	lab5-1.2.o	784
	lab5-1.asm	294
	lab5-1.o	752
*	lab5-2	9092
	lab5-2.2.asm	227
	lab5-2.asm	227
	lab5-2.o	1312

```
GNU nano 7.2 /home/kirillchubaev/work1/arch-pc/lab05/lab5-2.2.asm
%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 sprint

    mov ecx, buf1
    mov edx, 80

    call sread

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

    call quit

[ Прочитано 25 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

4. Я создал исполняемый файл и проверил работу измененной программы.  
Программа работает исправно.

```
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ nasm -f elf lab5-2.2.asm
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ld -m elf_i386 -o lab5-2.2 lab5-2.2.o
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$ ./lab5-2.2
Введите строку:Чубаев
Чубаев
kirillchubaev@ubuntu:~/work1/arch-pc/lab05$
```

## Вывод:

В ходе данной лабораторной работы я приобрёл полезные практические навыки работы в Midnight Commander. Помимо этого, я также освоил инструкции языка ассемблера `mov` и `int`.

## Список литературы:

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. *Newham C.* Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. *Robbins A.* Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. *Zarrelli G.* Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. *Колдаев В. Д., Лупин С. А.* Архитектура ЭВМ. — М. : Форум, 2018.
10. *Куляс О. Л., Никитин К. А.* Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. *Новожилов О. П.* Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. *Робачевский А., Немнюгин С., Стесик О.* Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

14. *Столяров А.* Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).
15. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. *Таненбаум Э., Бос Х.* Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).