

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Новичков Максим Алексеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Порядок выполнения лабораторной работы	6
3.2	Задание для самостоятельной работы	11
4	Выводы	15

Список иллюстраций

3.1	мс	6
3.2	Создаем каталог	6
3.3	touch	7
3.4	Открываем файл, заполняем	7
3.5	Открываем файл и убеждаемся, что файл содержит текст программы	8
3.6	Проверка	8
3.7	Копируем файл	9
3.8	Создаем копию файла	9
3.9	Редактируем файл	10
3.10	Смотрим, как работает программа и сравниваем с прошлой . . .	11
3.11	Создаем копию файла lab5-1.asm	12
3.12	Редактируем файл	13
3.13	Проверяем правильность	13
3.14	Редактируем файл	14
3.15	Проверяем правильность программы	14

1 Цель работы

Освоить инструкции языка ассемблера mov. Приобрести знания использования Midnight Commander.

2 Задание

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

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

Открываем Mid. Commander (рис. 3.1)

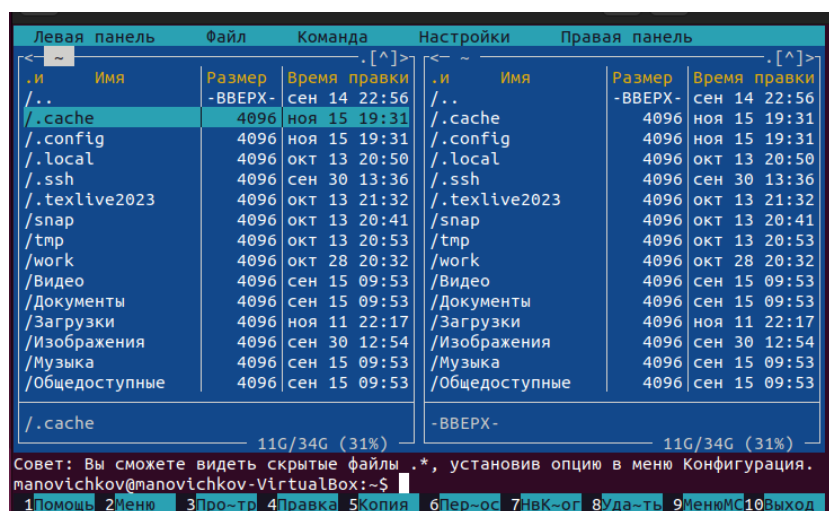


Рис. 3.1: mc

Создаем каталог lab05 (рис. 3.2)

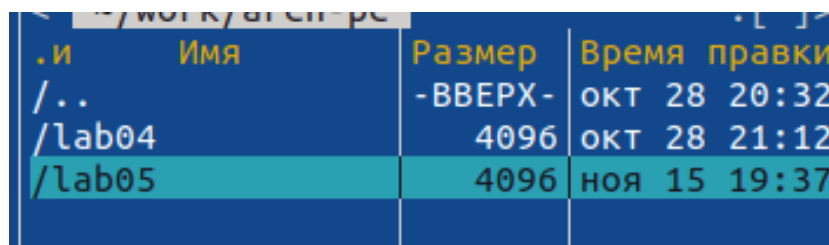


Рис. 3.2: Создаем каталог

Создаем файл lab5-1.asm (рис. 3.3)

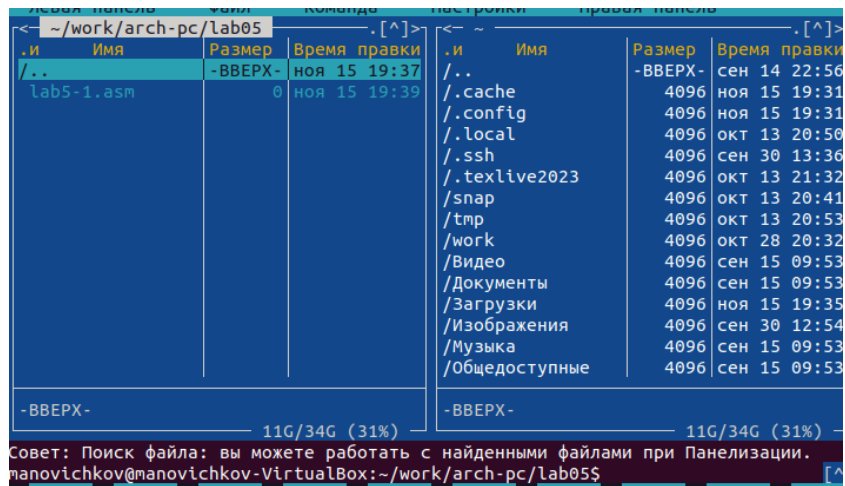


Рис. 3.3: touch

Открываем файл для редактирования и заполняем его по листингу (рис. 3.4)

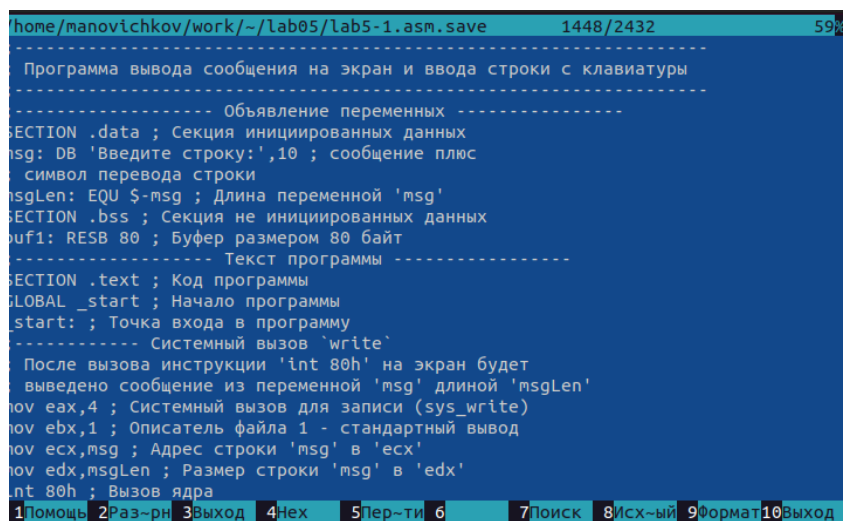


Рис. 3.4: Открывем файл, заполняем

Открывем файл и просматриваем (рис. 3.5)

```
manovichkov@manovichkov-VirtualBox: ~...
manovichkov@manovichkov-VirtualBox:~$ nasm -f elf lab5-1.asm
nasm: fatal: unable to open input file `lab5-1.asm' No such file
or directory
manovichkov@manovichkov-VirtualBox:~$ cd
cd                                cd-fix-profile    cd-it8
cd-create-profile  cd-iccdump
manovichkov@manovichkov-VirtualBox:~$ cd ~/work/arch-pc/lab05
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ nasm -f
elf lab5-1.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ld -m el
f_i386 -o lab5-1 lab5-1.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Новичков Максим Алексеевич
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$
```

Рис. 3.5: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем файл (рис. 3.6)

```
manovichkov@manovichkov-VirtualBox: ~...
manovichkov@manovichkov-VirtualBox:~$ nasm -f elf lab5-1.asm
nasm: fatal: unable to open input file `lab5-1.asm' No such file
or directory
manovichkov@manovichkov-VirtualBox:~$ cd
cd                                cd-fix-profile    cd-it8
cd-create-profile  cd-iccdump
manovichkov@manovichkov-VirtualBox:~$ cd ~/work/arch-pc/lab05
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ nasm -f
elf lab5-1.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ld -m el
f_i386 -o lab5-1 lab5-1.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Новичков Максим Алексеевич
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$
```

Рис. 3.6: Проверка

Копируем файл в нужную директорию (рис. 3.7)

Имя	Размер	Время правк
./..	ВВЕРХ-	ноя 15 19:3
in_out.asm	3942	ноя 15 22:2
*lab5-1	8744	ноя 15 21:4
lab5-1.asm	2432	ноя 15 19:4
lab5-1.o	752	ноя 15 21:4
lab5-2.asm	2432	ноя 15 19:4

Рис. 3.7: Копируем файл

Создаем копию файла (рис. ??)

Создаем копию файла

Рис. 3.8: Создаем копию файла

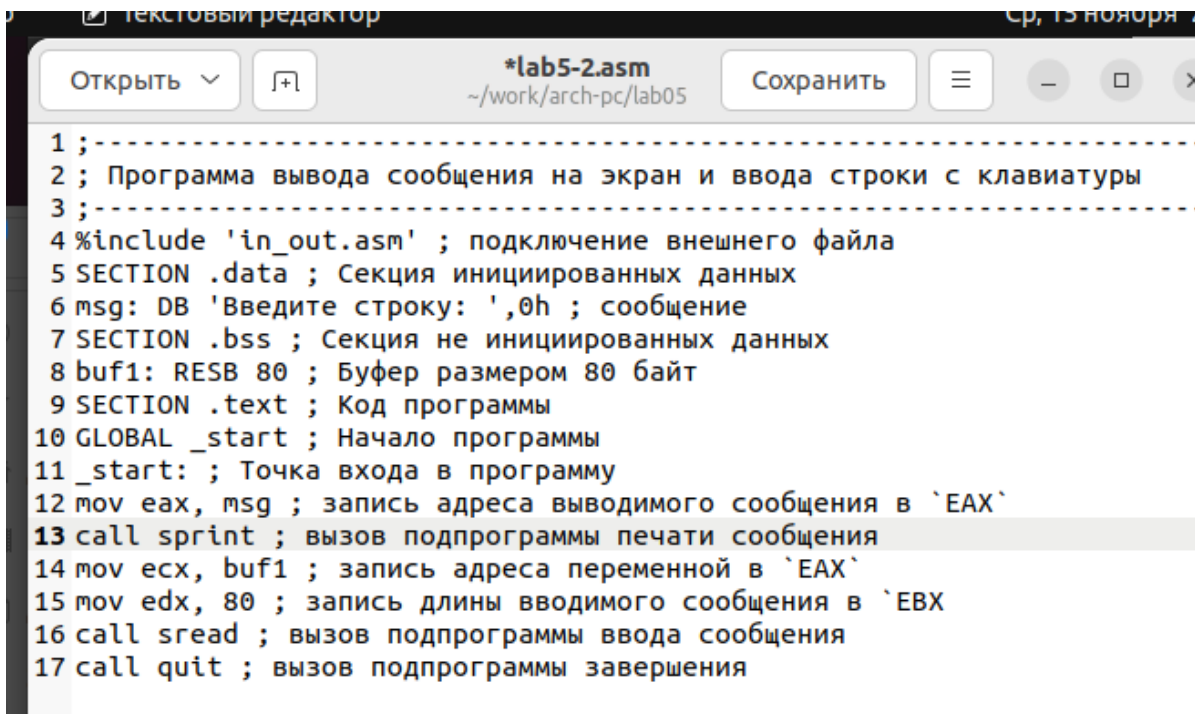
Проверяем созданный файл (рис. ??)

```

1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data ; Секция инициированных данных
6 msg: DB 'Введите строку: ',0h ; сообщение
7 SECTION .bss ; Секция не инициированных данных
8 buf1: RESB 80 ; Буфер размером 80 байт
9 SECTION .text ; Код программы
0 GLOBAL _start ; Начало программы
1 _start: ; Точка входа в программу
2 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
3 call sprintf ; вызов подпрограммы печати сообщения
4 mov ecx, buf1 ; запись адреса переменной в `EAX`
5 mov edx, 80 ; запись длины вводимого сообщения в `EBX`

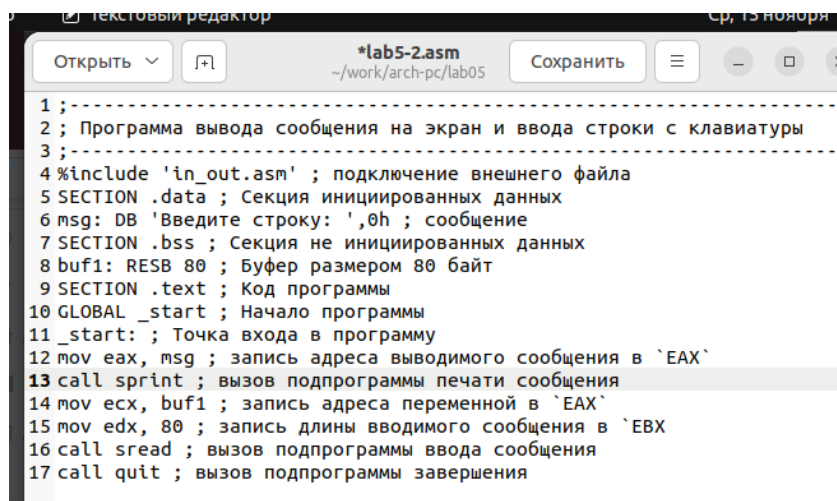
```

Открываем новый файл и заполняем его (рис. ??)



```
1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data ; Секция инициированных данных
6 msg: DB 'Введите строку: ',0h ; сообщение
7 SECTION .bss ; Секция не инициированных данных
8 buf1: RESB 80 ; Буфер размером 80 байт
9 SECTION .text ; Код программы
10 GLOBAL _start ; Начало программы
11 _start: ; Точка входа в программу
12 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
13 call sprint ; вызов подпрограммы печати сообщения
14 mov ecx, buf1 ; запись адреса переменной в `EAX`
15 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
16 call sread ; вызов подпрограммы ввода сообщения
17 call quit ; вызов подпрограммы завершения
```

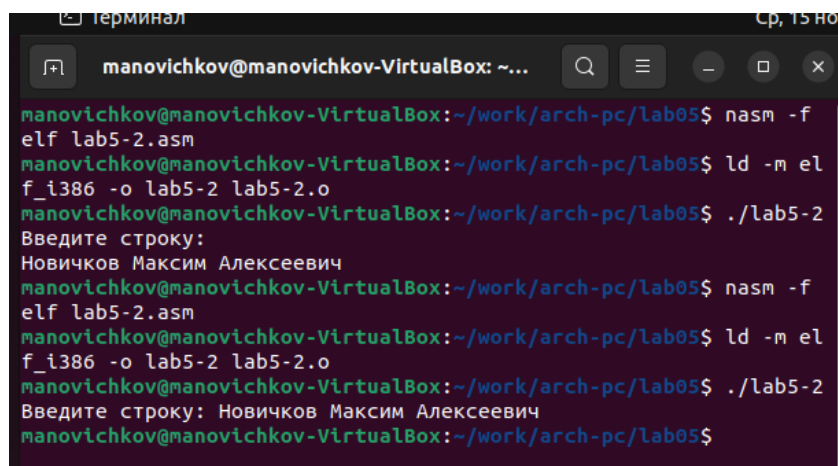
Открываем файл для редактирования и меняем sprintf на sprint(рис. 3.9)



```
1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data ; Секция инициированных данных
6 msg: DB 'Введите строку: ',0h ; сообщение
7 SECTION .bss ; Секция не инициированных данных
8 buf1: RESB 80 ; Буфер размером 80 байт
9 SECTION .text ; Код программы
10 GLOBAL _start ; Начало программы
11 _start: ; Точка входа в программу
12 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
13 call sprint ; вызов подпрограммы печати сообщения
14 mov ecx, buf1 ; запись адреса переменной в `EAX`
15 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
16 call sread ; вызов подпрограммы ввода сообщения
17 call quit ; вызов подпрограммы завершения
```

Рис. 3.9: Редактируем файл

Транслируем и запускаем файл(рис. 3.10)

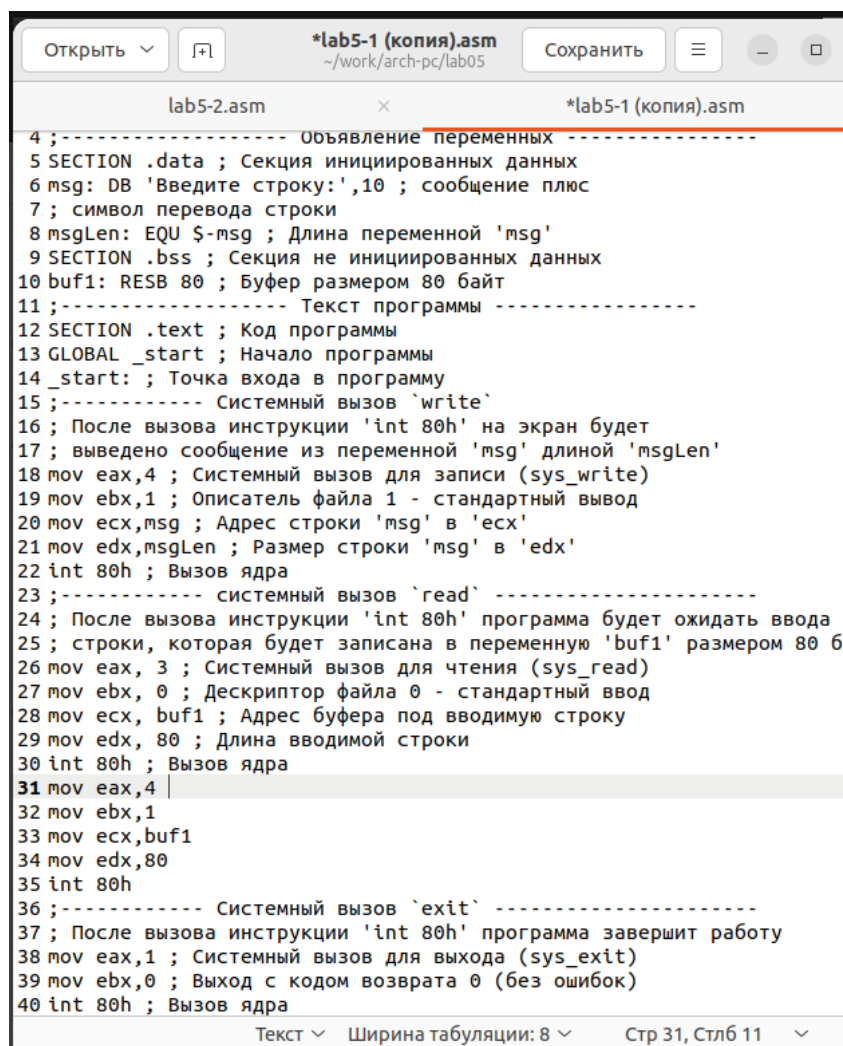
A screenshot of a terminal window titled "терминал" (terminal). The window shows a series of commands and their outputs in a dark-themed environment. The user is working in a directory ~/work/arch-pc/lab05. The commands executed are: nasm -f elf lab5-2.asm, ld -m elf_i386 -o lab5-2 lab5-2.o, and ./lab5-2. The output of the ./lab5-2 command is "Введите строку:" (Enter a line:), followed by the input "Новичков Максим Алексеевич". This sequence of commands and outputs is repeated twice in the screenshot.

```
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Новичков Максим Алексеевич
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Новичков Максим Алексеевич
manovichkov@manovichkov-VirtualBox: ~/work/arch-pc/lab05$
```

Рис. 3.10: Смотрим, как работает программа и сравниваем с прошлой

3.2 Задание для самостоятельной работы

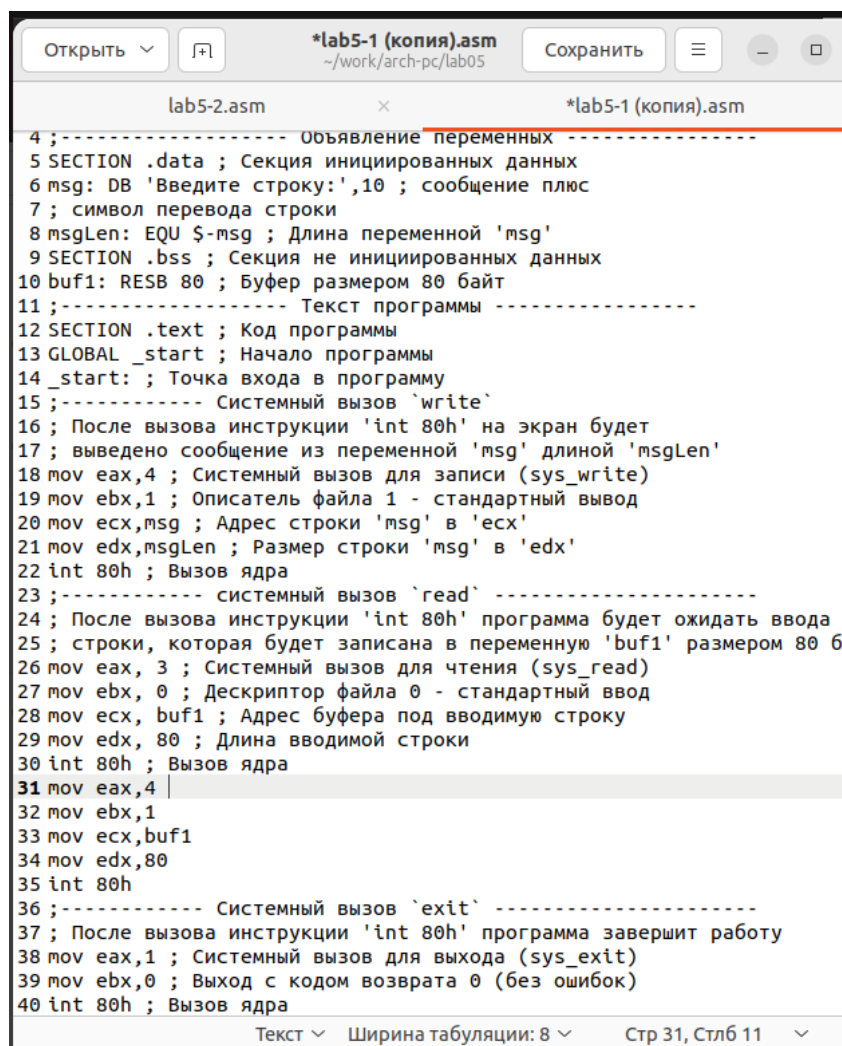
Создаем копию файла lab5-1.asm (рис. 3.11)



```
4 ;----- Объявление переменных -----
5 SECTION .data ; Секция инициированных данных
6 msg: DB 'Введите строку:',10 ; сообщение плюс
7 ; символ перевода строки
8 msgLen: EQU $-msg ; Длина переменной 'msg'
9 SECTION .bss ; Секция не инициированных данных
10 buf1: RESB 80 ; Буфер размером 80 байт
11 ;----- Текст программы -----
12 SECTION .text ; Код программы
13 GLOBAL _start ; Начало программы
14 _start: ; Точка входа в программу
15 ;----- Системный вызов `write` -----
16 ; После вызова инструкции 'int 80h' на экран будет
17 ; выведено сообщение из переменной 'msg' длиной 'msgLen'
18 mov eax,4 ; Системный вызов для записи (sys_write)
19 mov ebx,1 ; Описатель файла 1 - стандартный вывод
20 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
21 mov edx,msgLen ; Размер строки 'msg' в 'edx'
22 int 80h ; Вызов ядра
23 ;----- системный вызов `read` -----
24 ; После вызова инструкции 'int 80h' программа будет ожидать ввода
25 ; строки, которая будет записана в переменную 'buf1' размером 80 б
26 mov eax, 3 ; Системный вызов для чтения (sys_read)
27 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
28 mov ecx, buf1 ; Адрес буфера под вводимую строку
29 mov edx, 80 ; Длина вводимой строки
30 int 80h ; Вызов ядра
31 mov eax,4
32 mov ebx,1
33 mov ecx,buf1
34 mov edx,80
35 int 80h
36 ;----- Системный вызов `exit` -----
37 ; После вызова инструкции 'int 80h' программа завершит работу
38 mov eax,1 ; Системный вызов для выхода (sys_exit)
39 mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
40 int 80h ; Вызов ядра
```

Рис. 3.11: Создаем копию файла lab5-1.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 3.12)



```
4 ;----- Объявление переменных -----
5 SECTION .data ; Секция инициализированных данных
6 msg: DB 'Введите строку:',10 ; сообщение плюс
7 ; символ перевода строки
8 msgLen: EQU $-msg ; Длина переменной 'msg'
9 SECTION .bss ; Секция не инициализированных данных
10 buf1: RESB 80 ; Буфер размером 80 байт
11 ;----- Текст программы -----
12 SECTION .text ; Код программы
13 GLOBAL _start ; Начало программы
14 _start: ; Точка входа в программу
15 ;----- Системный вызов `write` -----
16 ; После вызова инструкции 'int 80h' на экран будет
17 ; выведено сообщение из переменной 'msg' длиной 'msgLen'
18 mov eax,4 ; Системный вызов для записи (sys_write)
19 mov ebx,1 ; Описатель файла 1 - стандартный вывод
20 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
21 mov edx,msgLen ; Размер строки 'msg' в 'edx'
22 int 80h ; Вызов ядра
23 ;----- системный вызов `read` -----
24 ; После вызова инструкции 'int 80h' программа будет ожидать ввода
25 ; строки, которая будет записана в переменную 'buf1' размером 80 б
26 mov eax, 3 ; Системный вызов для чтения (sys_read)
27 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
28 mov ecx, buf1 ; Адрес буфера под вводимую строку
29 mov edx, 80 ; Длина вводимой строки
30 int 80h ; Вызов ядра
31 mov eax,4
32 mov ebx,1
33 mov ecx,buf1
34 mov edx,80
35 int 80h
36 ;----- Системный вызов `exit` -----
37 ; После вызова инструкции 'int 80h' программа завершит работу
38 mov eax,1 ; Системный вызов для выхода (sys_exit)
39 mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
40 int 80h ; Вызов ядра
```

Рис. 3.12: Редактируем файл

Транслируем файл и запускаем программу (рис. 3.13)

Проверяем правильность

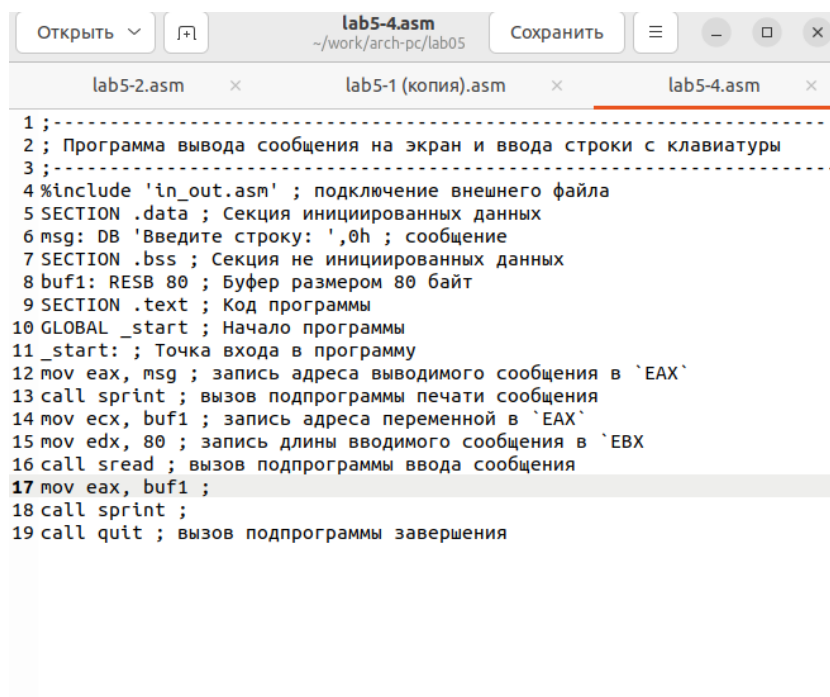
Рис. 3.13: Проверяем правильность

Создаем копию файла lab5-2.asm (рис. ??)

![Создаем копию файла lab5-2.asm](image/Снимок экрана от 2023-11-15 23-17-17.png){#fig:015 width=70%}

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

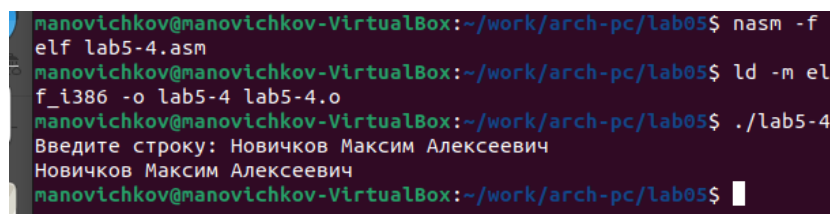
(рис. 3.14)



```
1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data ; Секция инициализированных данных
6 msg: DB 'Введите строку: ',0h ; сообщение
7 SECTION .bss ; Секция не инициализированных данных
8 buf1: RESB 80 ; Буфер размером 80 байт
9 SECTION .text ; Код программы
10 GLOBAL _start ; Начало программы
11 _start: ; Точка входа в программу
12 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
13 call sprint ; вызов подпрограммы печати сообщения
14 mov ecx, buf1 ; запись адреса переменной в `EAX`
15 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
16 call sread ; вызов подпрограммы ввода сообщения
17 mov eax, buf1 ;
18 call sprint ;
19 call quit ; вызов подпрограммы завершения
```

Рис. 3.14: Редактируем файл

Транслируем файл и запускаем (рис. 3.9)



```
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ nasm -f
elf lab5-4.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ld -m el
f_i386 -o lab5-4 lab5-4.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-4
Введите строку: Новичков Максим Алексеевич
Новичков Максим Алексеевич
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab05$
```

Рис. 3.15: Проверяем правильность программы

4 Выводы

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