

Отчет по лабораторной работе №5

Жукова София Викторовна

Содержание

цель работы	5
Выполнение лабораторной работы	6
Выводы	17

Список иллюстраций

1	Введем команду ms	6
2	Переходим в каталог	7
3	Создадим папку	7
4	Переходим в каталог	8
5	Создаем файл lab5-1.asm	8
6	Перепишем код	9
7	Проверим как запиался код	9
8	Проверяем, как работает данная программа	10
9	Скачиваем	10
10	Копируем	11
11	Создаем копию	11
12	Смотрим, как сработала программа	12
13	Редактируем файл	13
14	Смотрим, как сработала программа и сравниваем с прошлой	13
15	Создаем копию файла lab5-1.asm	14
16	Редактируем файл	15
17	Проверяем правильность написания программы	15
18	Проверка	16

Список таблиц

цель работы

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

Выполнение лабораторной работы

Откроем Midnight Commander (рис. [-@fig:001]).

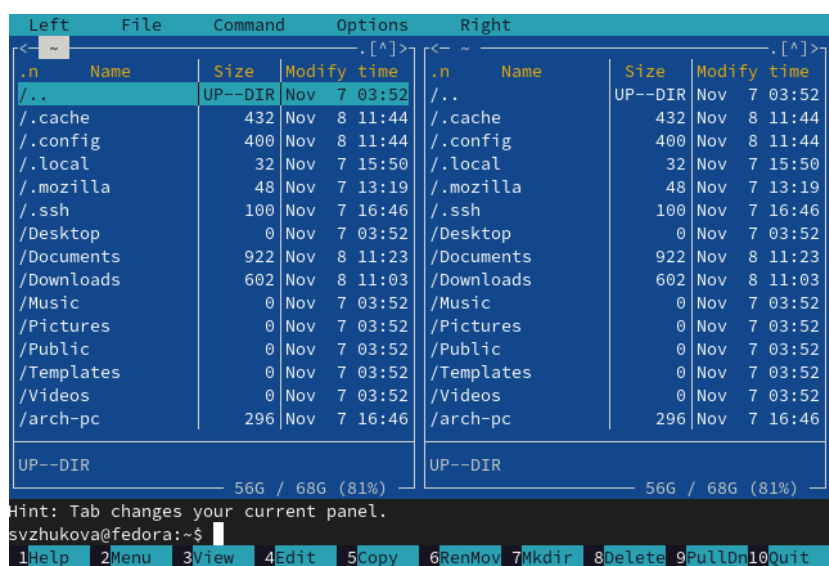


Рис. 1: Введем команду mc

Пользуясь клавишами \uparrow , \downarrow и Enter перейдем в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 (рис. [-@fig:002]).

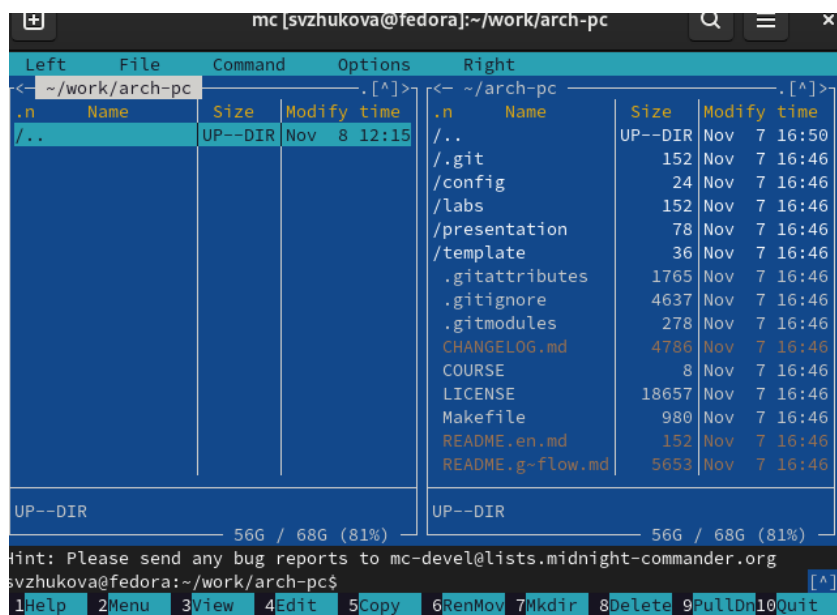


Рис. 2: Переходим в каталог

С помощью функциональной клавиши F7 создадим папку lab05 (рис. [fig:003]).

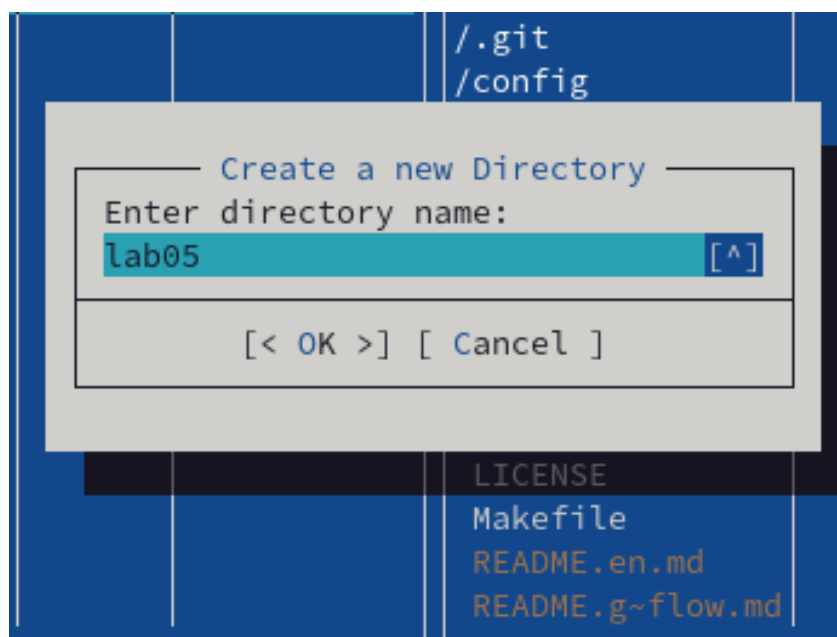


Рис. 3: Создадим папку

Перейдем в созданный каталог (рис. [-@fig:004]).

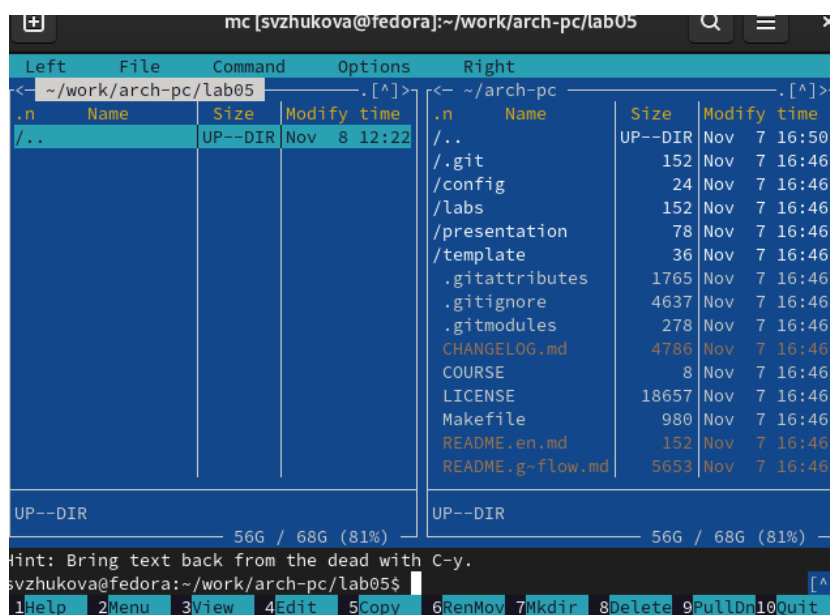


Рис. 4: Переходим в каталог

Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm (рис. [-@fig:005]).

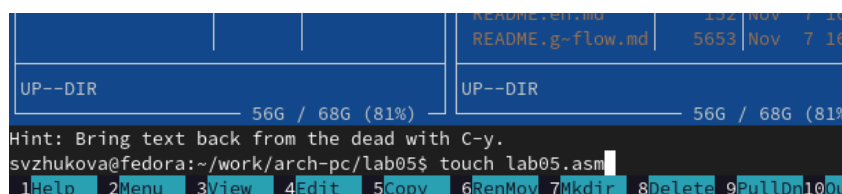
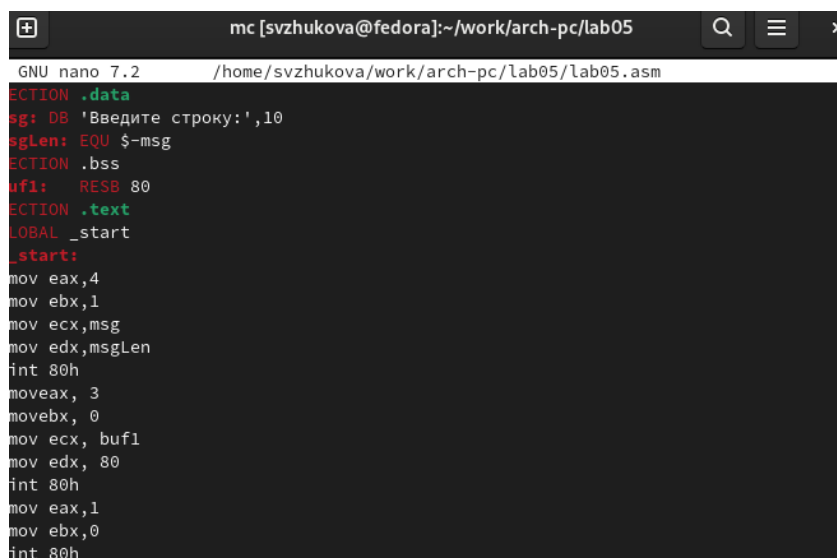


Рис. 5: Создаем файл lab5-1.asm

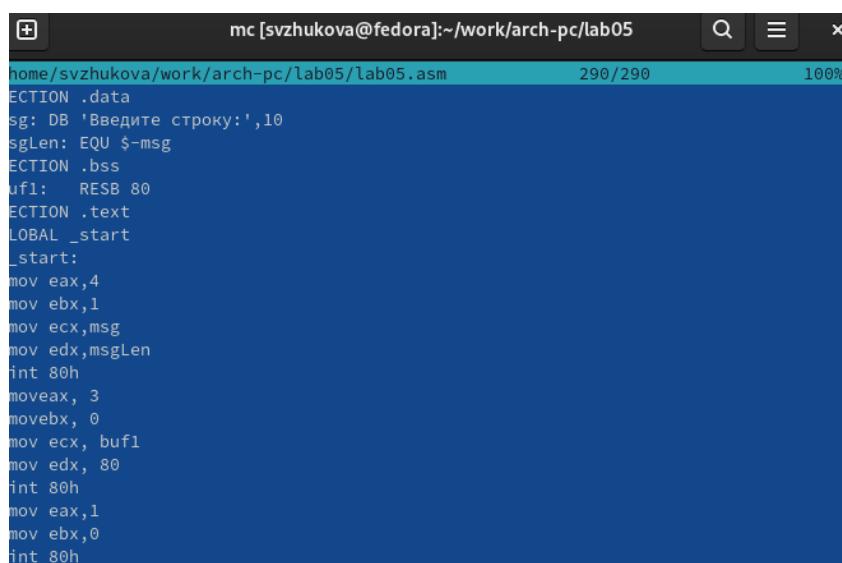
С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе. Введем текст программы из листинга, сохраним изменения и закроем файл (рис. [-@fig:006]).

A terminal window with a dark background. The title bar shows 'mc [svzhukova@fedora]:~/work/arch-pc/lab05'. The editor is GNU nano 7.2, editing /home/svzhukova/work/arch-pc/lab05/lab05.asm. 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
moveax, 3
movebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 6: Перепишем код

С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы. (рис. [-@fig:007]).

A terminal window with a dark background. The title bar shows 'mc [svzhukova@fedora]:~/work/arch-pc/lab05'. The editor is GNU nano 7.2, editing /home/svzhukova/work/arch-pc/lab05/lab05.asm. The code is the same as in Figure 6. A search bar at the top right shows '290/290' and '100%'. The code is as follows:

```
home/svzhukova/work/arch-pc/lab05/lab05.asm 290/290 100%
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
moveax, 3
movebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 7: Проверим как запиался код

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

(рис. [-@fig:008]).

```
svzhukova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
svzhukova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
svzhukova@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку
Жукова София Виктровна
svzhukova@fedora:~/work/arch-pc/lab05$
```

Рис. 8: Проверяем, как работает данная программа

Скачиваем файл со страницы курса (рис. [-@fig:009]).

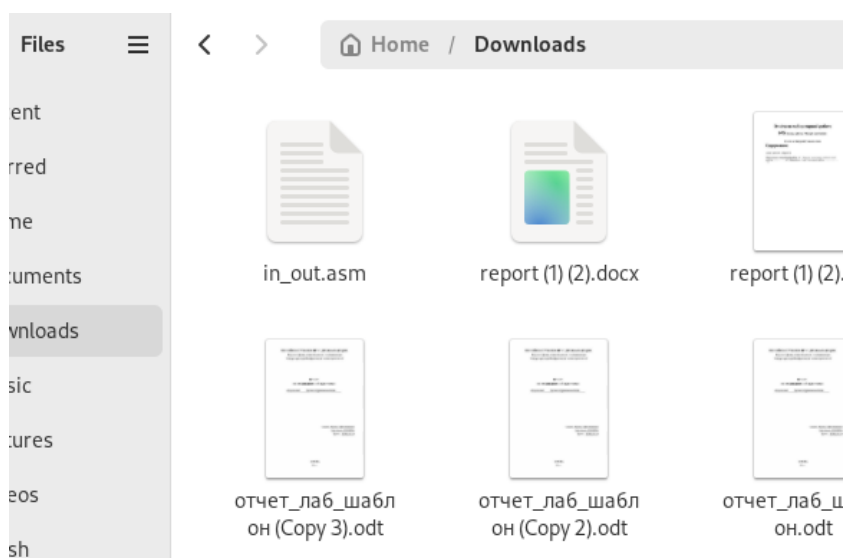


Рис. 9: Скачиваем

Копируем файл в нужную директорию (рис. [-@fig:010]).

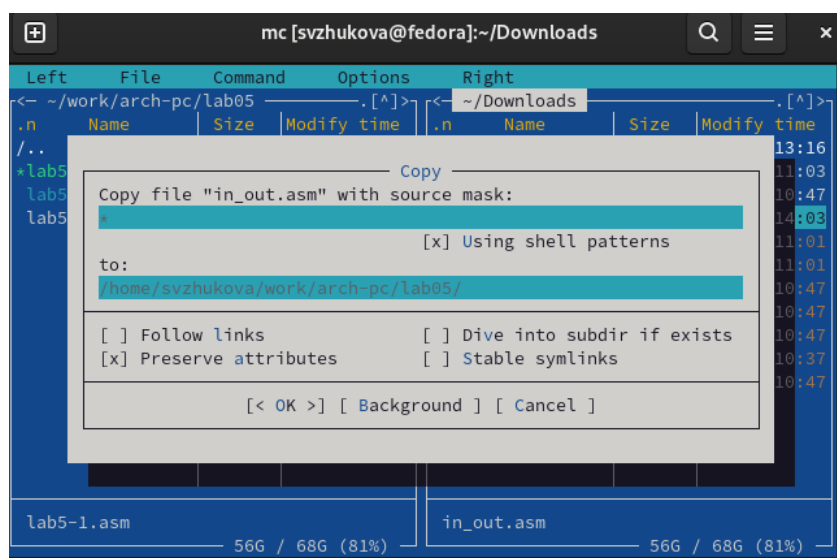


Рис. 10: Копируем

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm. Создаем копию файла клавишей F6 (рис. [-@fig:011]).

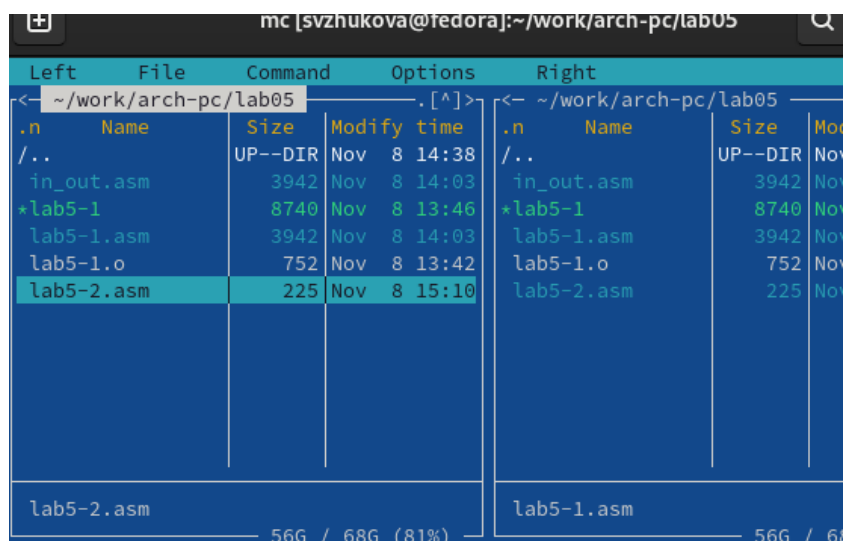
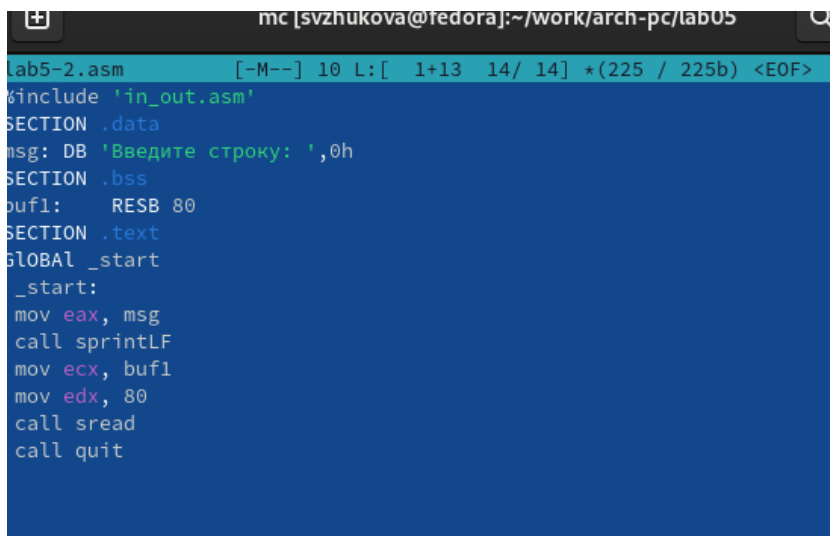


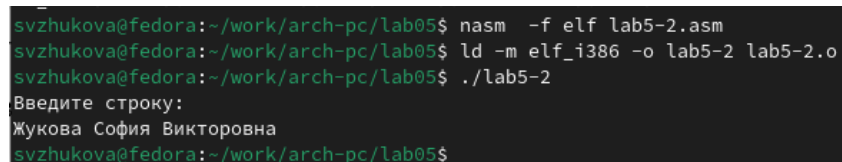
Рис. 11: Создаем копию

Открываем новый файл и заполняем его в соответствии с листингом (рис. [-@fig:012]).

A screenshot of a text editor window titled 'mc [svzhukova@fedora]:~/work/arch-pc/lab05'. The editor shows the contents of 'lab5-2.asm'. The code includes a header file, defines a data section with a message, a bss section for a buffer, and a text section with the main logic. The logic involves printing a message and reading input from the user.

```
lab5-2.asm [-M--] 10 L: [ 1+13 14/ 14] *(225 / 225b) <EOF>
#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 sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit
```

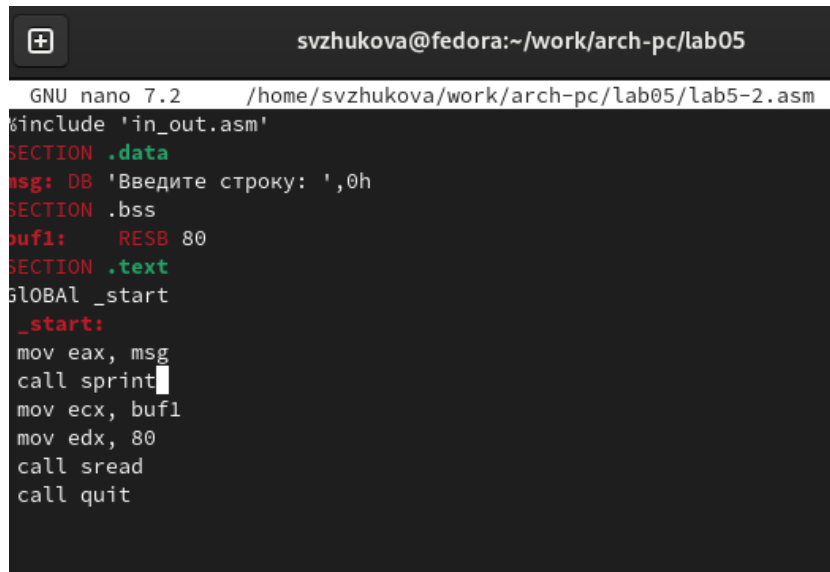
Транслируем и запускаем новый файл (рис. [-@fig:013]).

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf lab5-2.asm' to compile the assembly file into an object file, then 'ld -m elf_i386 -o lab5-2 lab5-2.o' to link it into an executable. Finally, they run './lab5-2' which prompts for input. The user enters 'Жукова София Викторовна' and the program ends.

```
svzhukova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
svzhukova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
svzhukova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Жукова София Викторовна
svzhukova@fedora:~/work/arch-pc/lab05$
```

Рис. 12: Смотрим, как сработала программа

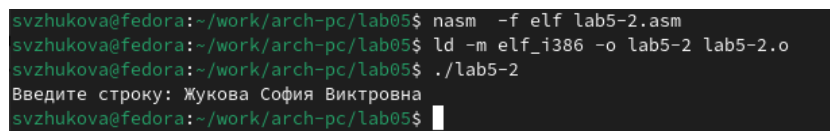
Снова открываем файл для редактирования и меняем sprintf на sprint (рис. [-@fig:014]).



```
svzhukova@fedora:~/work/arch-pc/lab05
GNU nano 7.2 /home/svzhukova/work/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 sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 13: Редактируем файл

Транслируем и запускаем файл (рис. [-@fig:015]).



```
svzhukova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
svzhukova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
svzhukova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Жукова София Викторовна
svzhukova@fedora:~/work/arch-pc/lab05$
```

Рис. 14: Смотрим, как сработала программа и сравниваем с прошлой

Можем сделать вывод, что команда `sprint` выводит текст в той же строке, а `sprintLF` переносит на новую строку.

Задание для самостоятельной работы

Создадим копию файла `lab5-1.asm`. (рис. [-@fig:016]).

Left	File	Command	Options	Right
~ /work/arch-pc/lab05				~ /work/arch-pc/lab05
.n	Name	Size	Modify time	.n Name Size Modify
/..		UP--DIR	Nov 8 14:38	/.. UP--DIR Nov 8 14:38
in_out.asm		3942	Nov 8 14:03	in_out.asm 3942 Nov 8 14:03
*lab5-1		8740	Nov 8 13:46	*lab5-1 8740 Nov 8 13:46
lab5-1.asm		3942	Nov 8 14:03	lab5-1.asm 3942 Nov 8 14:03
lab5-1.o		752	Nov 8 13:42	lab5-1.o 752 Nov 8 13:42
lab5-11.asm		322	Nov 9 10:50	lab5-11.asm 322 Nov 9 10:50
*lab5-2		9092	Nov 8 15:19	*lab5-2 9092 Nov 8 15:19
lab5-2.asm		224	Nov 8 15:19	lab5-2.asm 224 Nov 8 15:19
lab5-2.o		1312	Nov 8 15:19	lab5-2.o 1312 Nov 8 15:19
lab5-11.asm				UP--DIR
56G / 68G (81%)				56G / 68G
Hint: Key frequently visited ftp sites in the hotlist: type C-\. vzhukova@fedora:~/work/arch-pc/lab05\$				

Рис. 15: Создаем копию файла lab5-1.asm

Внесем изменения в программу (без использования внешнего файла in_out.asm), так чтобы она выводила введённую строку на экран. (рис. [-@fig:017]).

```

lab5-11.asm [----] 0 L: [ 1+ 0 1/ 29] *(0 / 350b) 0083 0x053 [
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,80
int 80h

mov eax,1
mov ebx,0
int 80h

```

Рис. 16: Редактируем файл

Транслируем файл и запускаем программу. На приглашение ввести строку введем свою фамилию. (рис. [-@fig:018]).

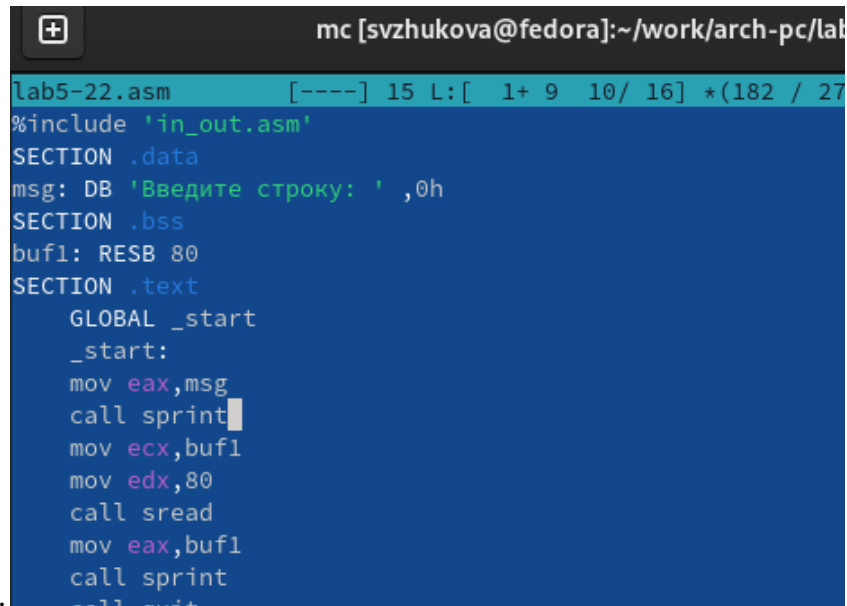
```

avzhukova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-11.asm
avzhukova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-11 lab5-11.o
avzhukova@fedora:~/work/arch-pc/lab05$ ./lab5-11
Введите строку
Кукова София Виктровна
Кукова София Виктровна
avzhukova@fedora:~/work/arch-pc/lab05$

```

Рис. 17: Проверяем правильность написания программы

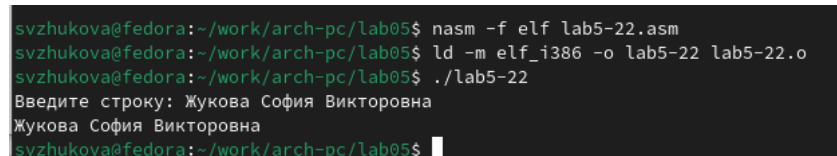
Создадим копию файла lab5-2.asm. Исправим текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она выводила введен-



```
lab5-22.asm      [----] 15 L: [ 1+ 9 10/ 16] *(182 / 27)
#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,buf1
call sprint
```

ную строку на экран (рис. [-@fig:019]).

Транслируем программу и проверяем как она работает (рис. [-@fig:020]).



```
svzhukova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
svzhukova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
svzhukova@fedora:~/work/arch-pc/lab05$ ./lab5-22
Введите строку: Жукова София Викторовна
Жукова София Викторовна
svzhukova@fedora:~/work/arch-pc/lab05$
```

Рис. 18: Проверка

Выводы

Мы приобрели практические навыки работы в Midnight Commander. Освоили инструкции языка ассемблера `mov` и `int`.