

Отчёта по лабораторной работе №5

Основы работы с Midnight Commander

Гурбанов Сарча

Содержание

1	<u>Цель работы</u>	4
2	<u>Задание</u>	5
3	<u>Выполнение лабораторной работы</u>	6
1.	<u>Порядок выполнения лабораторной работы</u>	6
2.	<u>Задание для самостоятельной работы</u>	15
4	<u>Выводы</u>	19

Список иллюстраций

1.	<u>Вводим в консоль команду <code>mc</code></u>	6
2.	<u>Переходим в каталог</u>	7
3.	<u>Создаем каталог функциональной клавишей F7</u>	8
4.	<u>Воспользуемся командой <code>touch</code></u>	9
5.	<u>Открываем файл функциональной клавишей, заполняем и сохраняем</u>	10
6.	<u>Открываем файл и убеждаемся, что файл содержит текст программы</u>	11
7.	<u>Проверяем, как работает данная программа</u>	11
8.	<u>Скачиваем файл</u>	12
9.	<u>Копируем скаченный файл</u>	12
10.	<u>Создаем копию файла клавишей F6</u>	13
11.	<u>Проверяем скопировался ли файл</u>	13
12.	<u>Открываем и заполняем файл</u>	14
13.	<u>Смотрим, как сработала программа</u>	14
14.	<u>Редактируем файл</u>	14
15.	<u>Смотрим, как сработала программа и сравниваем с прошлой</u>	15
16.	<u>Создаем копию файла <code>lab5-1.asm</code></u>	15
17.	<u>Редактируем файл</u>	16
18.	<u>Проверяем правильность написания программы</u>	16
19.	<u>Создаем копию файла <code>lab5-2.asm</code></u>	17
20.	<u>Редактируем файл</u>	17
21.	<u>Проверяем правильность написания программы</u>	18

1 Цель работы

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

2 Задание

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

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

Открываем Midnight Commander (рис. 3.1).

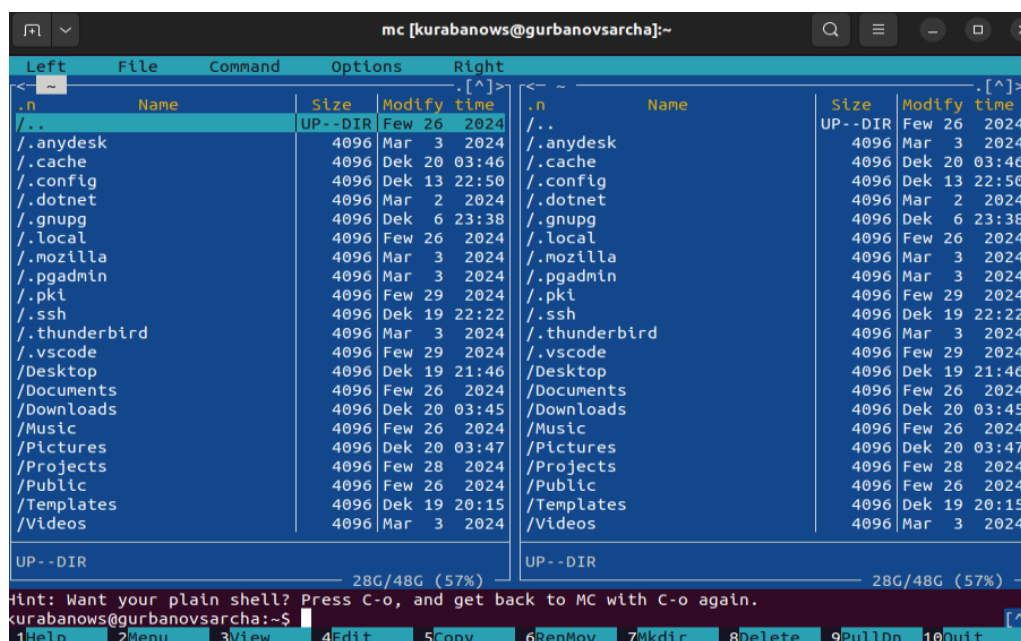


Рис. 3.1: Вводим в консоль команду mc

Переходим в каталог, созданный при выполнении 4 ЛБ (рис. 3.2).

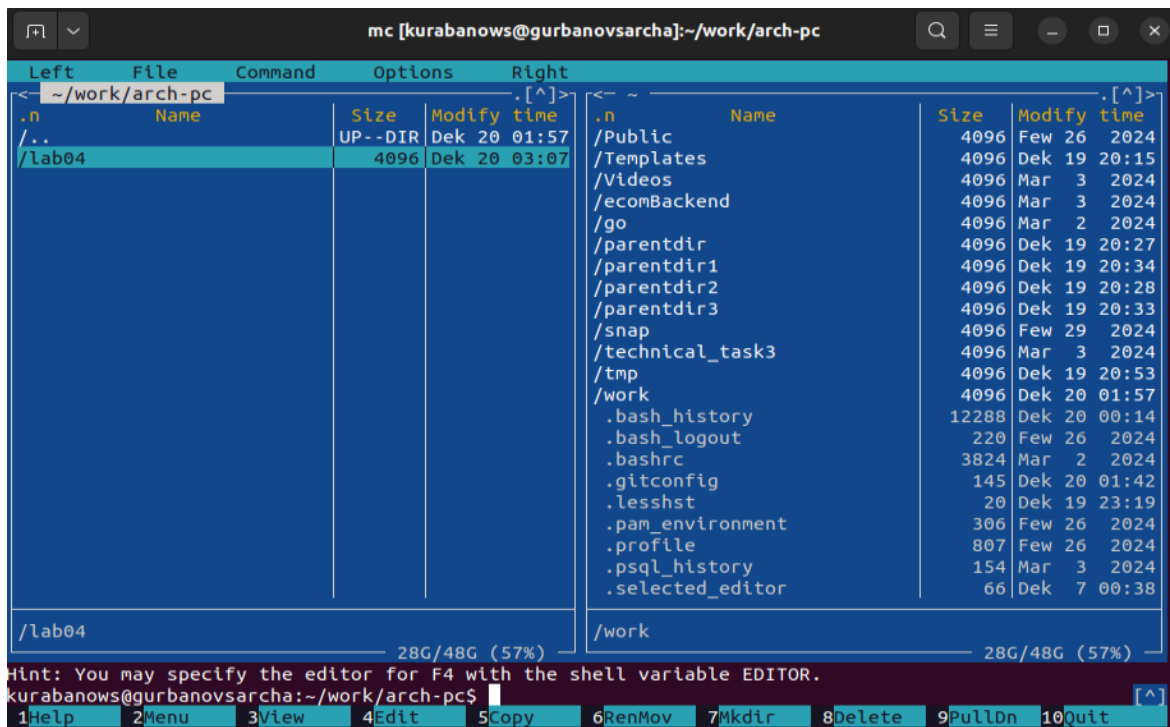


Рис. 3.2: Переходим в каталог

Создаем каталог lab05 (рис. 3.3).

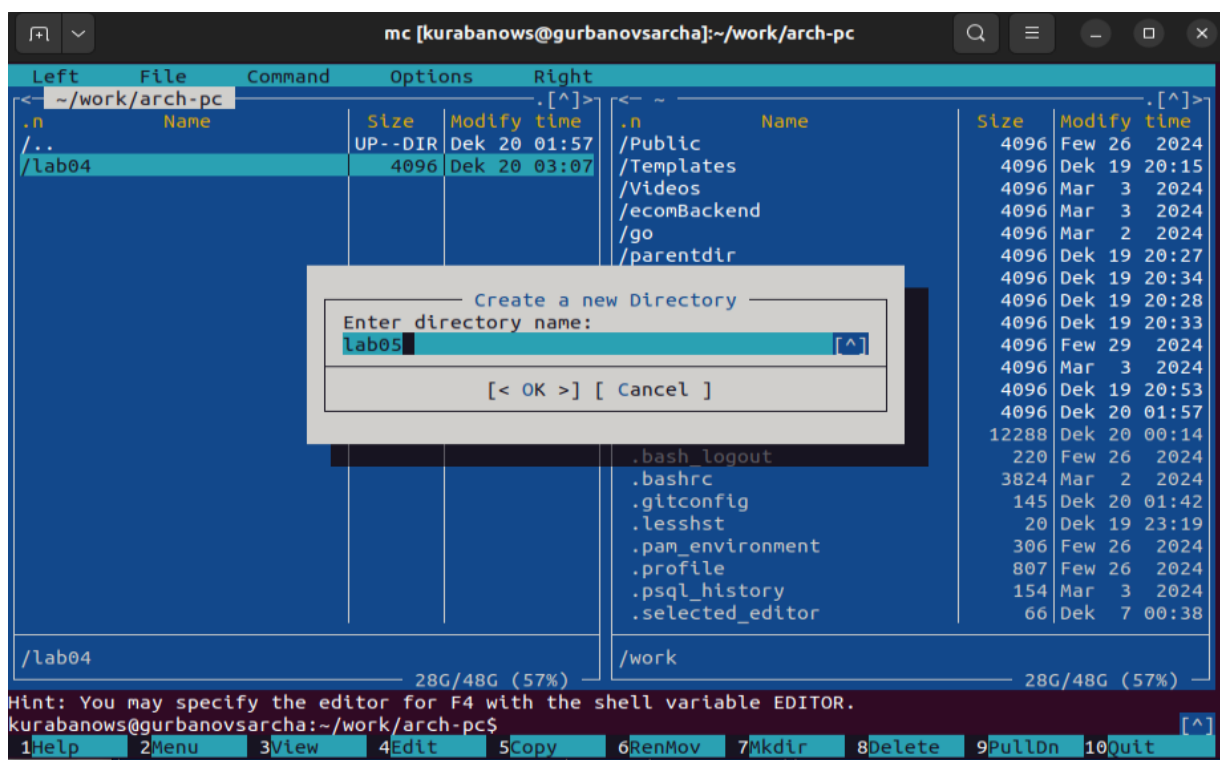


Рис. 3.3: Создаем каталог функциональной клавишей F7

Создаем файл lab5-1.asm (рис. 3.4).

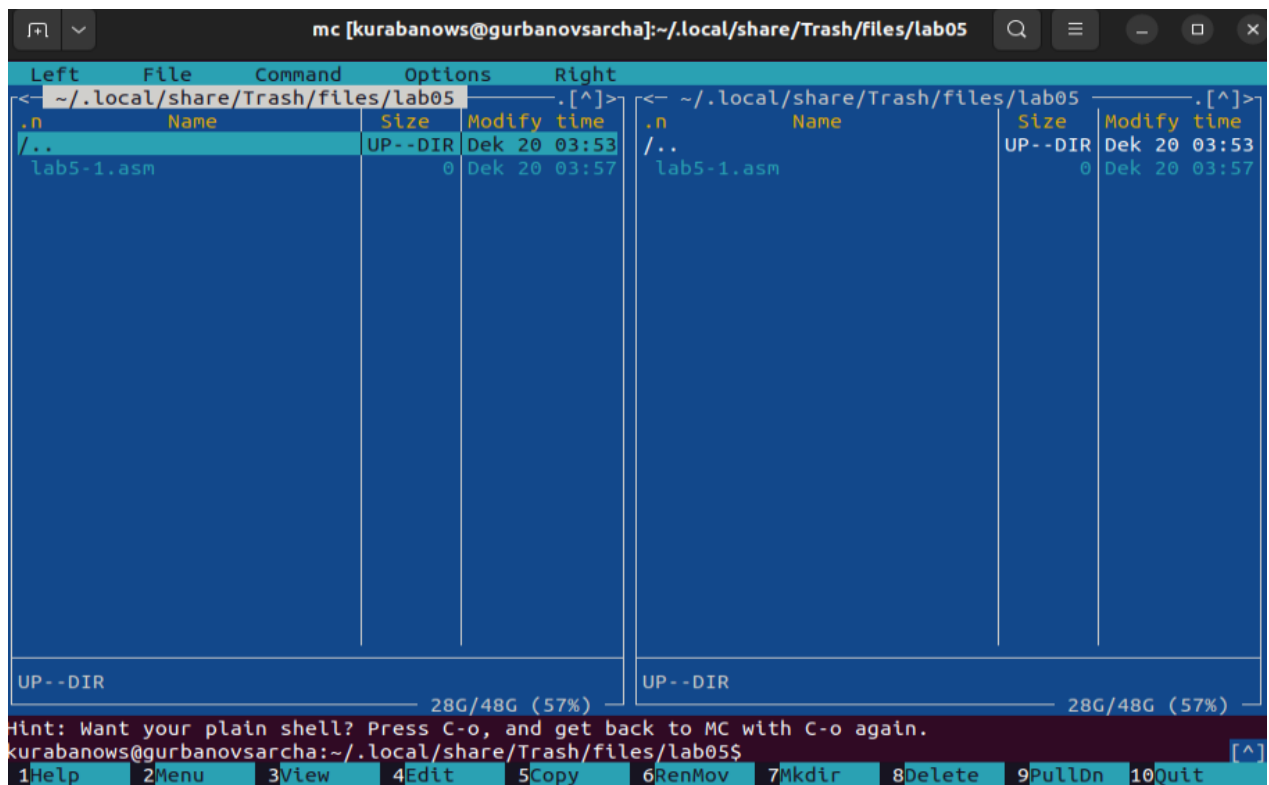


Рис. 3.4: Воспользуемся командой touch

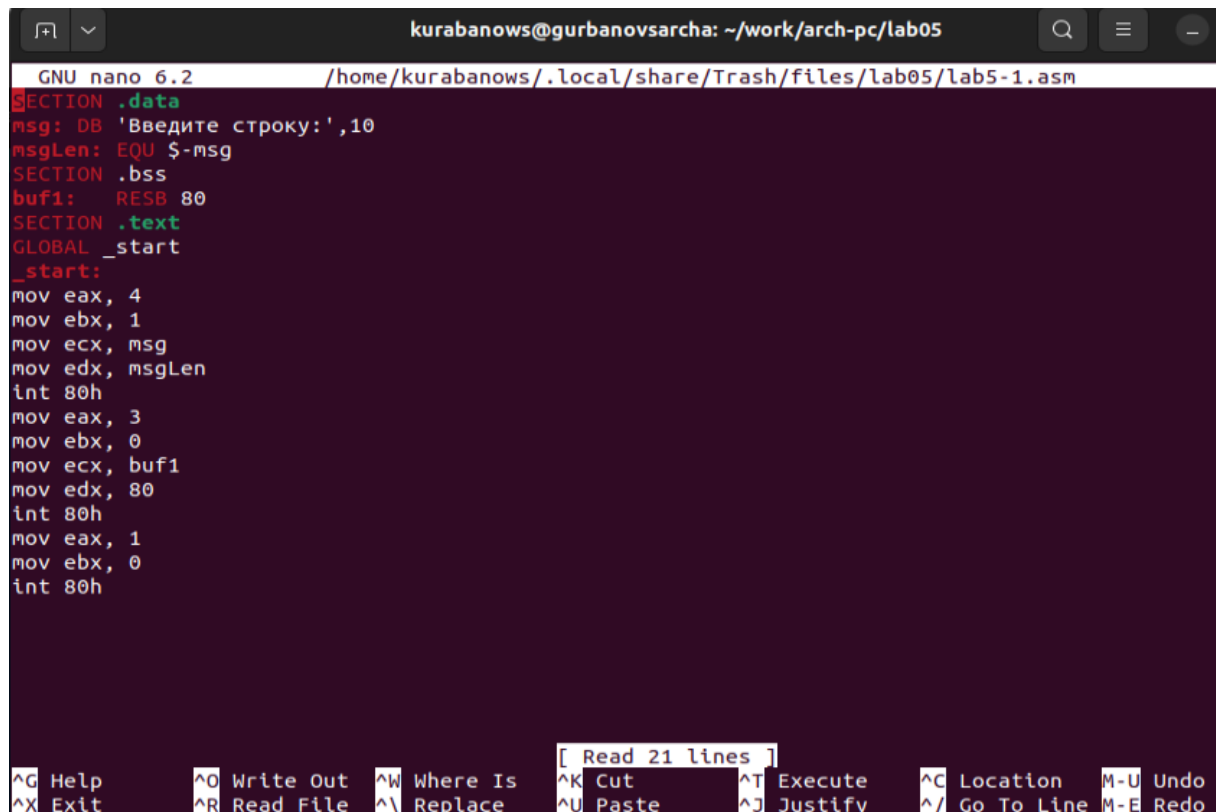
Открываем файл для редактирования и заполняем его по листингу (рис. 3.5).

```
mc [kurabanows@gurbanovsarcha]:~/local/share/Trash/files/lab05
/home/kurabanows/.l~s/lab05/lab5-1.asm [-M--] 6 L:[ 1+20 21/ 21] *(280 / 281b) 0104 0x068 [*][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
```

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

Рис. 3.5: Открываем файл функциональной клавишей, заполняем и сохраняем

Открываем файл для просмотра (рис. [3.6](#)).

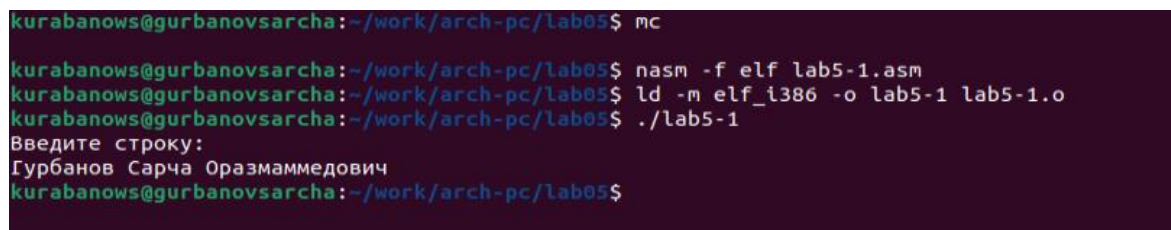


```
GNU nano 6.2 /home/kurabanows/.local/share/Trash/files/lab05/lab5-1.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, 4
mov ebx, 1
mov ecx, msg
mov edx, msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax, 1
mov ebx, 0
int 80h

[ Read 21 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

Рис. 3.6: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем исполняемый файл (рис. [3.7](#)).



```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nc
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Гурбанов Сарча Оразмаммедович
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$
```

Рис. 3.7: Проверяем, как работает данная программа

Скачиваем файл со страницы курса (рис. [3.8](#)).

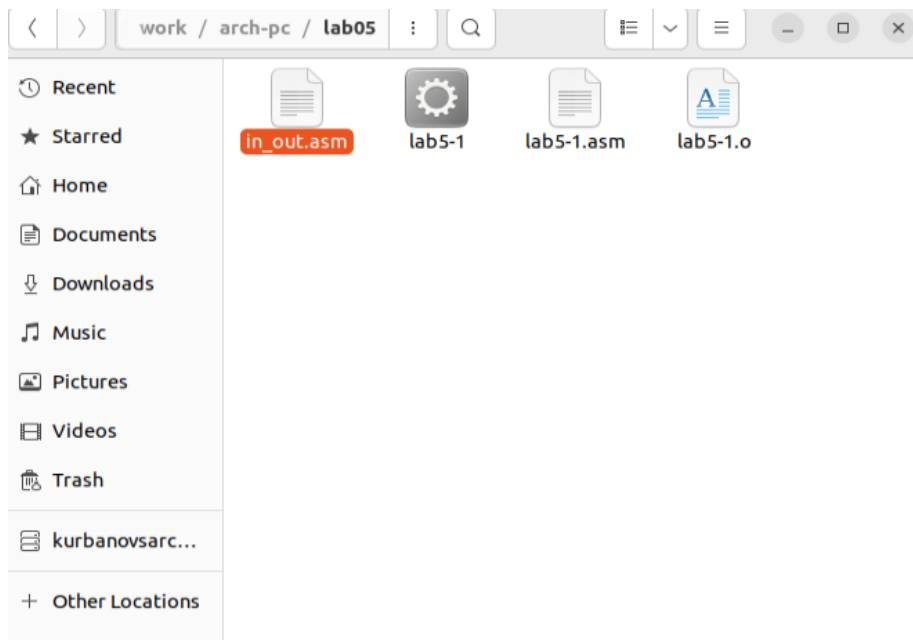


Рис. 3.8: Скачиваем файл

Копируем файл в нужную директорию (рис. 3.9).

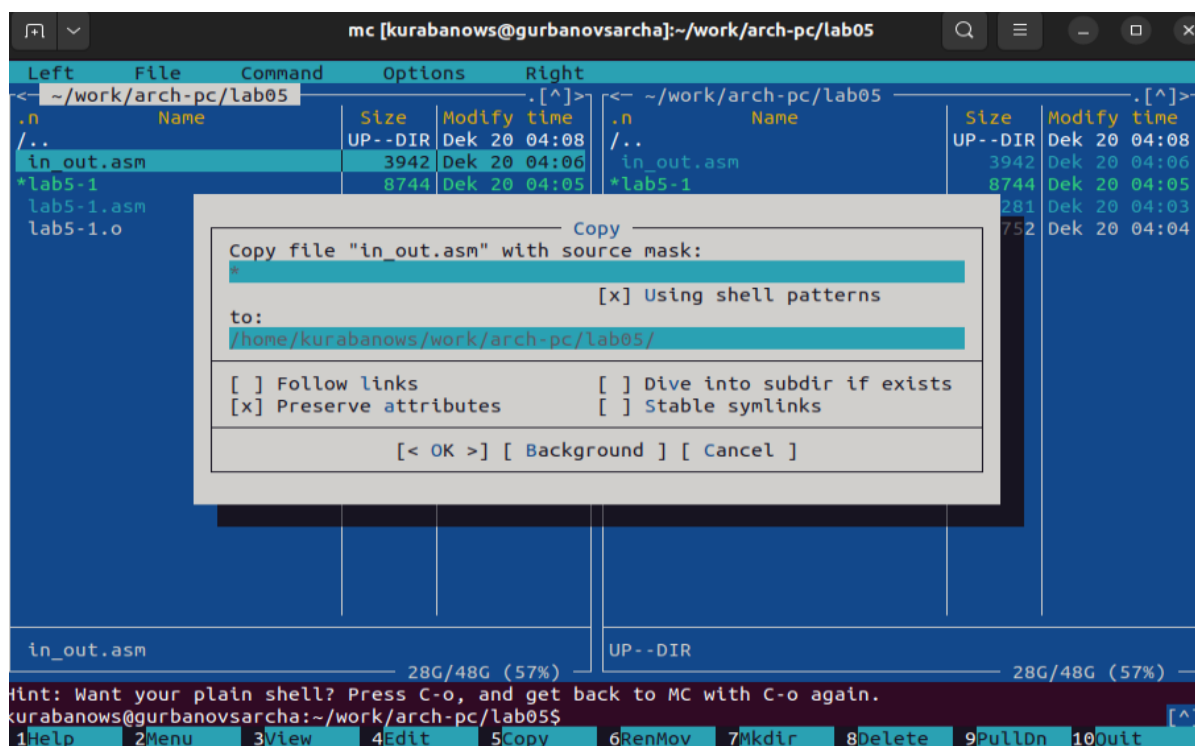


Рис. 3.9: Копируем скаченный файл

Создаем копию файла lab5-1.asm (рис. 3.10).

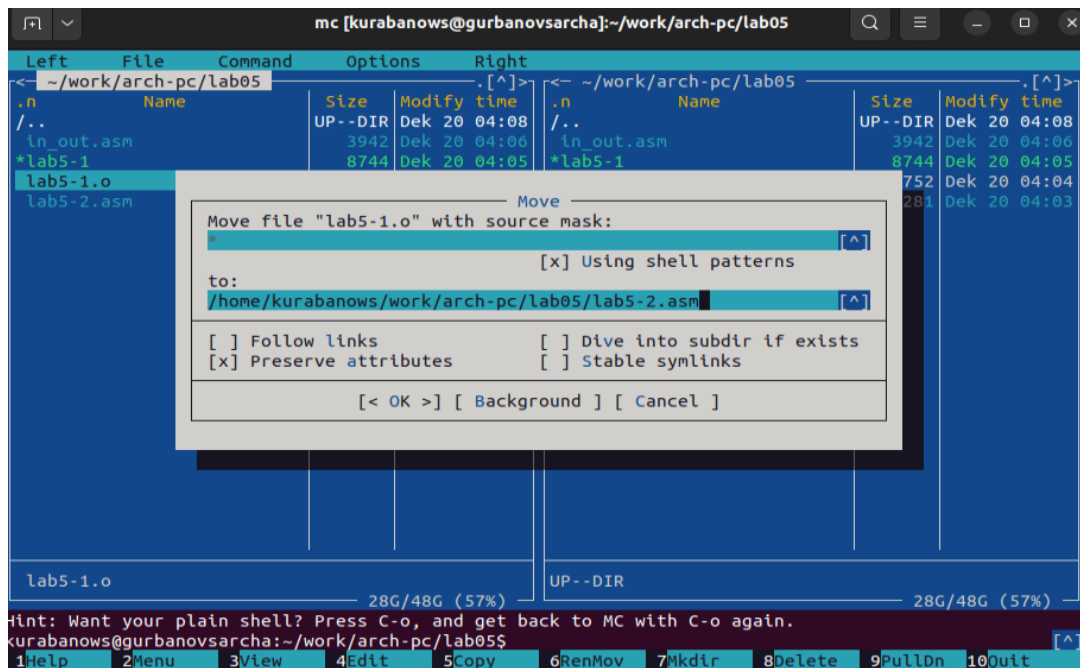


Рис. 3.10: Создаем копию файла клавишей F6

Проверяем созданный файл (рис. 3.11).

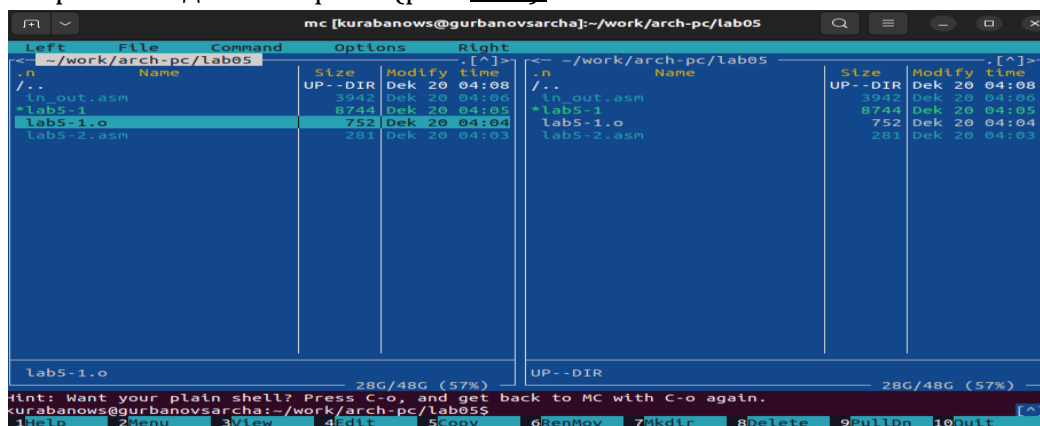
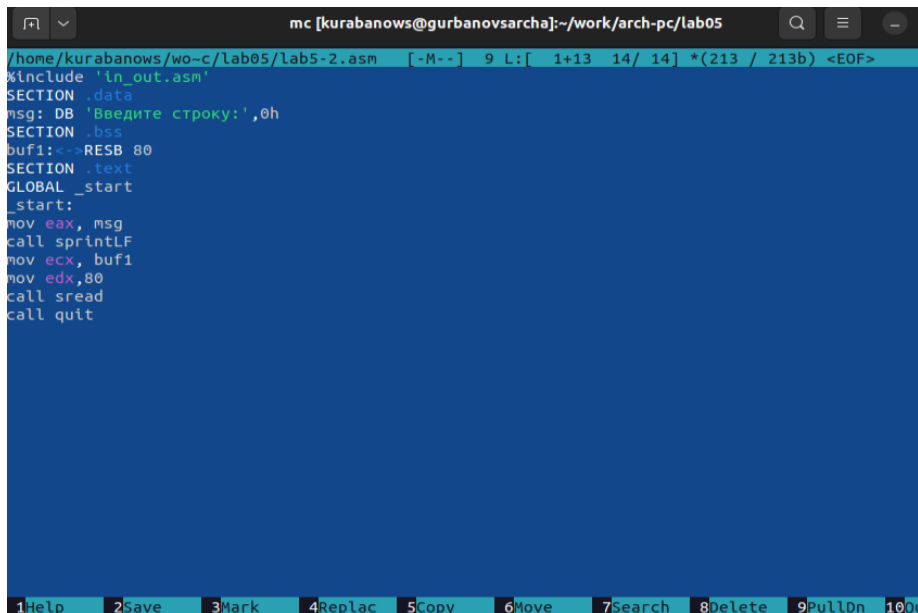


Рис. 3.11: Проверяем скопировался ли файл

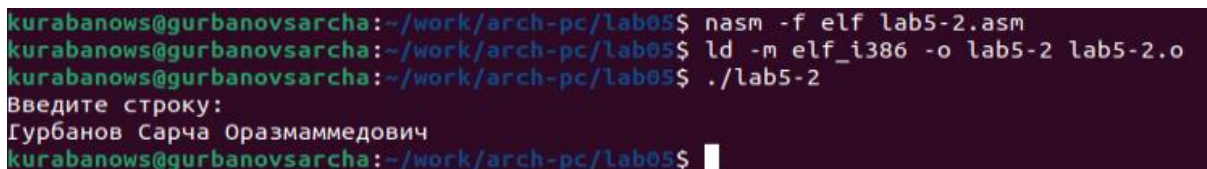
Открываем новый файл и заполняем его в соответствии с листингом (рис. 3.12).



```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab05
/home/kurabanows/work/arch-pc/lab05/lab5-2.asm [-M--] 9 L: [ 1+13 14/ 14] *(213 / 213b) <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
```

Рис. 3.12: Открываем и заполняем файл

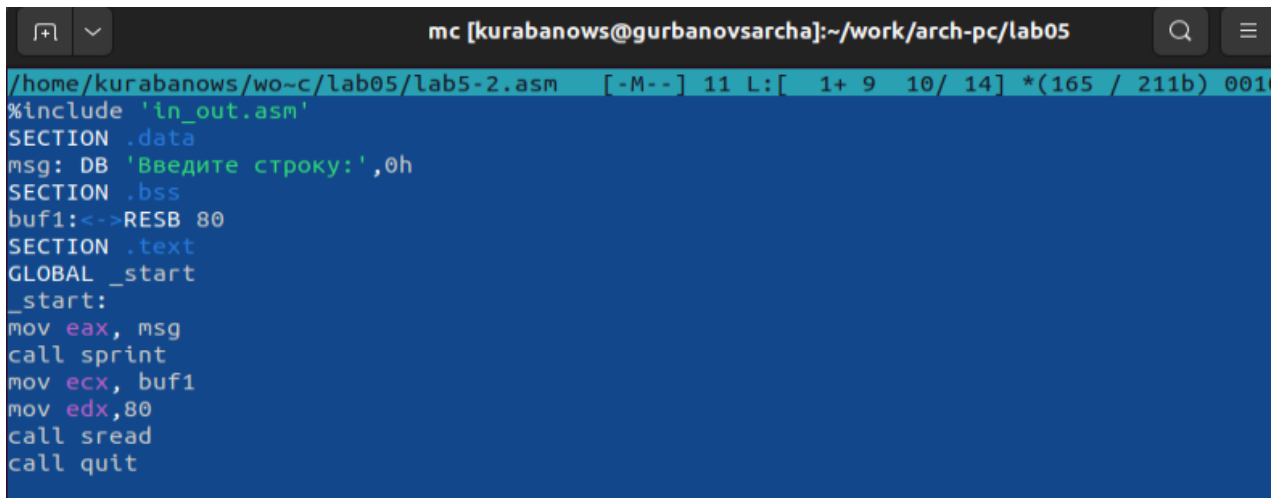
Транслируем и запускаем новый файл (рис. 3.13).



```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Гурбанов Сарча Оразмаммедович
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$
```

Рис. 3.13: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем sprintf на printf (рис. 3.14).



```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab05
/home/kurabanows/work/arch-pc/lab05/lab5-2.asm [-M--] 11 L: [ 1+ 9 10/ 14] *(165 / 211b) 001
#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 printf
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 3.14: Редактируем файл

Транслируем и запускаем файл(рис. 3.15).

```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Гурбанов Сарча Оразмаммедович
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$
```

Рис. 3.15: Смотрим, как сработала программа и сравниваем с прошлой

Таким образом можем понять, что команда `sprint` выводит текст в той же строке, а `sprintLF` переносит на новую строку.

3.2 Задание для самостоятельной работы

Создаем копию файла `lab5-1.asm` и называем его так же (рис. 3.16).

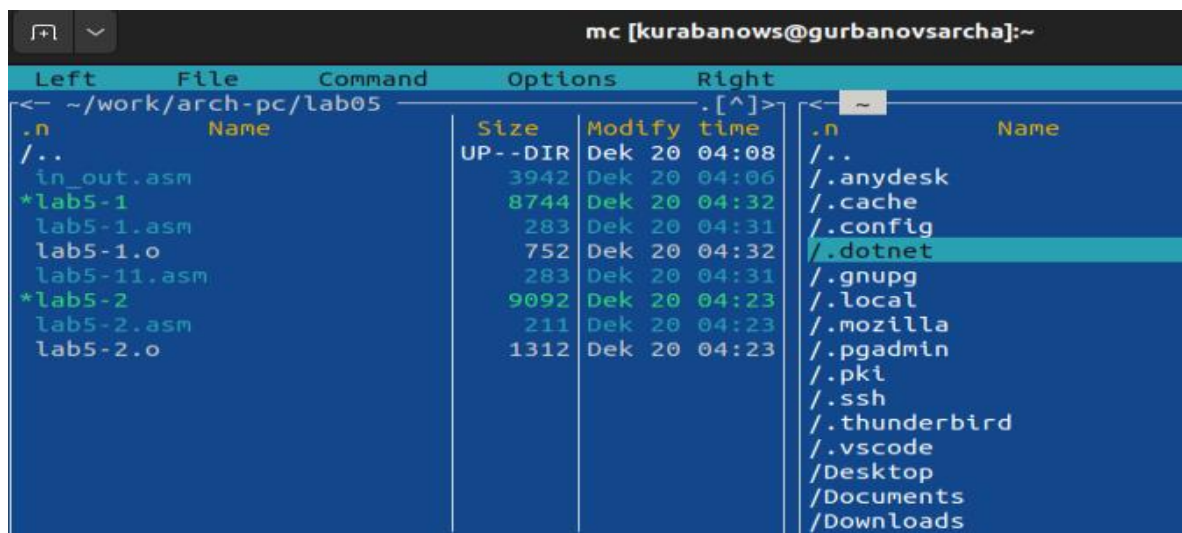
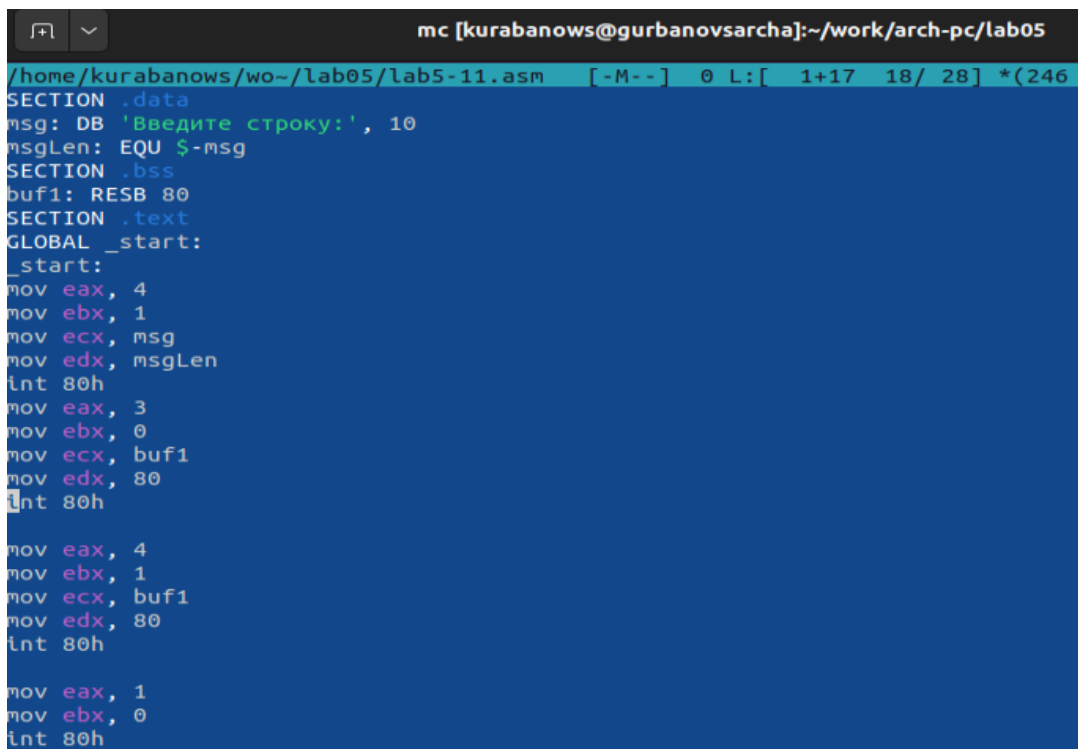


Рис. 3.16: Создаем копию файла `lab5-1.asm`

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 3.17).

A screenshot of a text editor window titled 'mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab05'. The editor shows the contents of a file named 'lab5-11.asm'. The code is written in assembly language and includes sections for data, bss, and text. It defines a message 'Введите строку:', sets up a buffer, and contains two identical blocks of assembly code that initialize registers and perform some operations. The code is as follows:

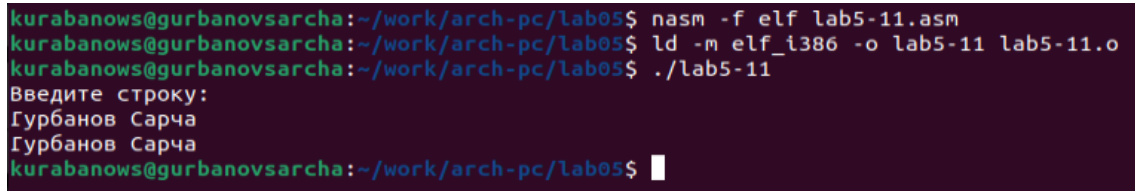
```
SECTION .data
msg: DB 'Введите строку:', 10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start:
_start:
mov eax, 4
mov ebx, 1
mov ecx, msg
mov edx, msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

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

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

Рис. 3.17: Редактируем файл

Транслируем файл и запускаем программу (рис. [3.18](#)).

A screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf lab5-11.asm', then 'ld -m elf_i386 -o lab5-11 lab5-11.o', and finally './lab5-11'. The program prompts 'Введите строку:' and the user enters 'Гурбанов Сарча' twice. The terminal output is as follows:

```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nasm -f elf lab5-11.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-11 lab5-11.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ./lab5-11
Введите строку:
Гурбанов Сарча
Гурбанов Сарча
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$
```

Рис. 3.18: Проверяем правильность написания программы

Создаем копию файла lab5-2.asm и называем его так же (рис. [3.19](#)).

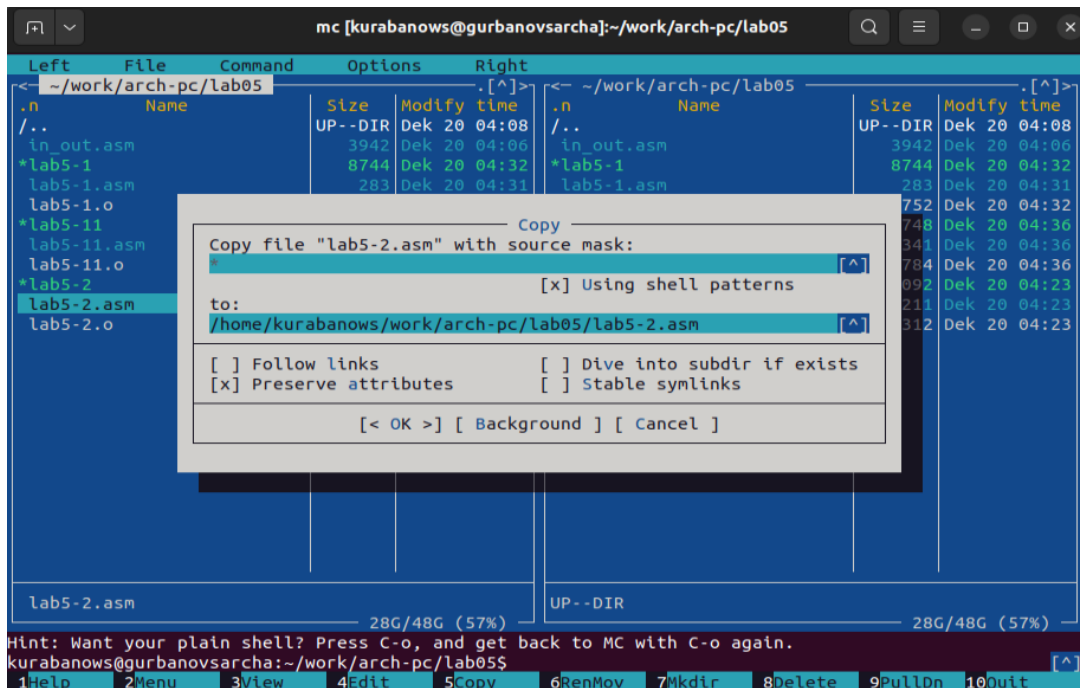


Рис. 3.19: Создаем копию файла lab5-2.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль (рис. 3.20).

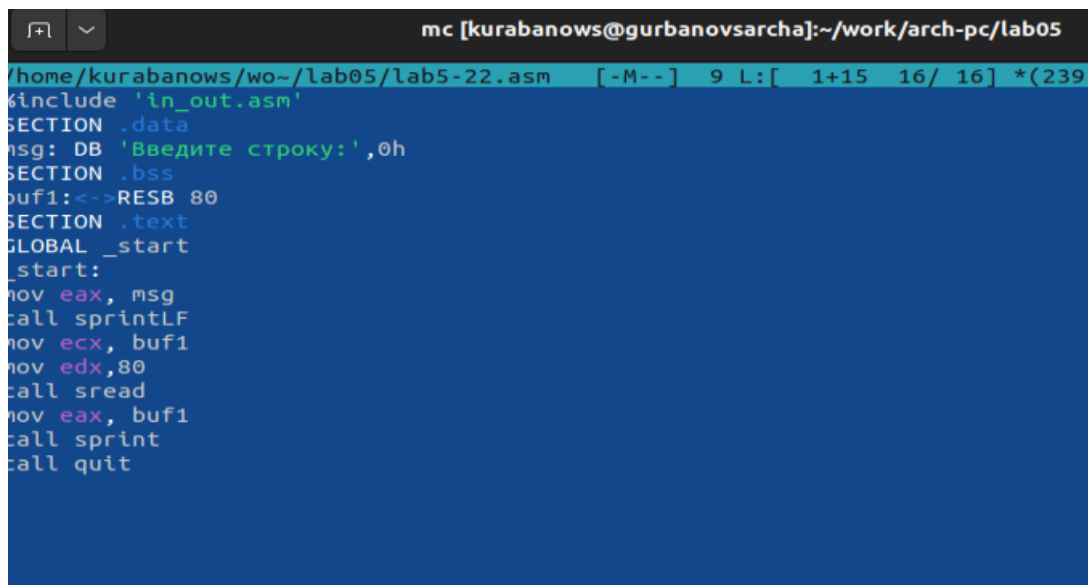


Рис. 3.20: Редактируем файл

Транслируем файл и запускаем программу (рис. 3.21).

```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ ./lab5-22
Введите строку:
Гурбанов Сарча
Гурбанов Сарча
kurabanows@gurbanovsarcha:~/work/arch-pc/lab05$ █
```

Рис. 3.21: Проверяем правильность написания программы

4 Выводы

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