

# **Отчет по лабораторной работе No.5**

**Дисциплины: Архитектура компьютера**

Нджову Нелиа

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	20
	Список литературы	21

# Список иллюстраций

3.1	Рис 1	7
3.2	Рис 2	8
3.3	Рис 3	8
3.4	Рис 4	9
3.5	Рис 5	9
3.6	Рис 6	10
3.7	Рис 7	10
3.8	Рис 8	11
3.9	Рис 9	11
3.10	Рис 10	12
3.11	Рис 11	13
3.12	Рис 12	13
3.13	Рис 13	14
3.14	Рис 14	14
3.15	Рис 15	15
3.16	Рис 16	15
3.17	Рис 17	16
3.18	Рис 18	17
3.19	Рис 19	18
3.20	Рис 20	18

## **Список таблиц**

# 1 Цель работы

Целью данной работы является приобретение практических навыков работы в Midnight Commander. Также освоите инструкции на языке ассемблера `mov` и `int`.

## 2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Основы работы с mc

Я открою Midnight Commander, выполнив команду mc в терминале(рис 1)

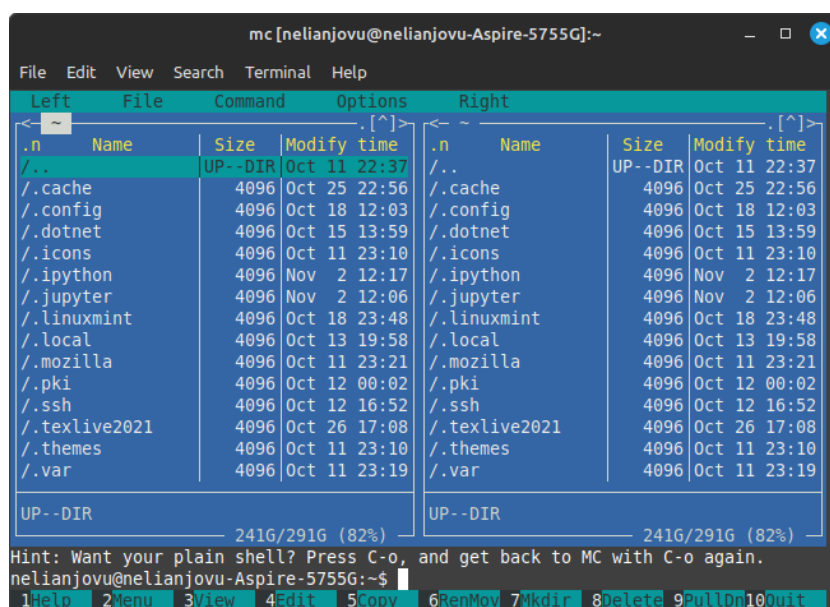


Рис. 3.1: Рис 1

Я перейду в каталог ~/work/arch-рс, который я создал в лаборатории 4 в mc Commander(рис 2)

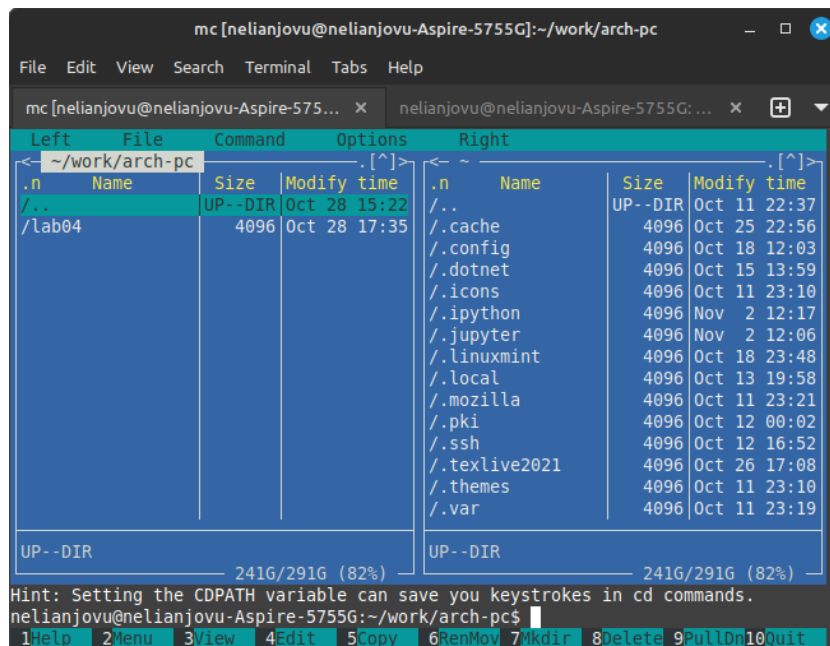


Рис. 3.2: Рис 2

Я буду использовать функциональную клавишу F7, чтобы создать папку lab05 и перейти в созданный каталог(рис 3)

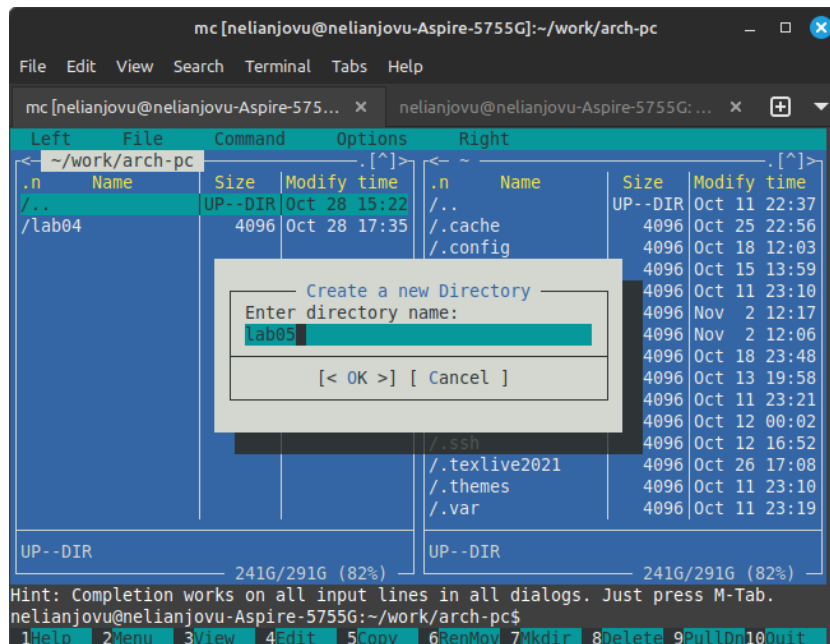


Рис. 3.3: Рис 3



В строке ввода я напишу команду touch lab5-1.asm для создания файла, в котором буду работать(рис 4)

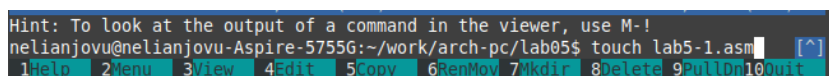


Рис. 3.4: Рис 4

## 3.2 Структура программы на языке ассемблера NASM

Я открываю созданный файл для повторного редактирования. Я введу заданный текст программы, сохраню изменения и закрою файл(рис 5)

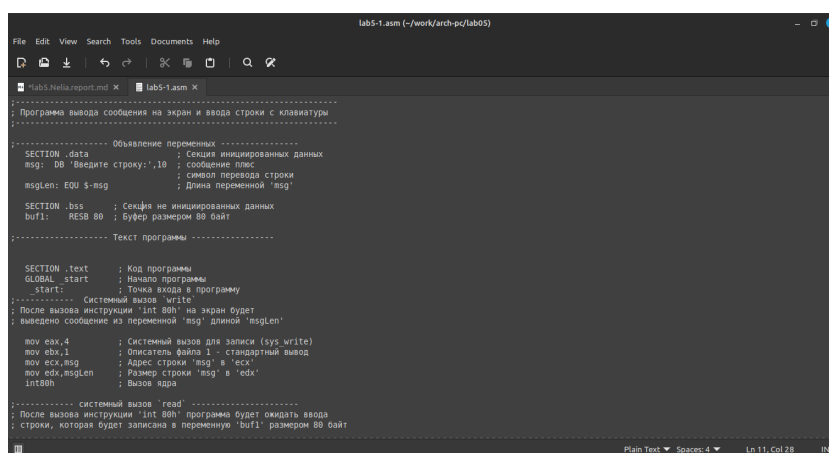


Рис. 3.5: Рис 5

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы(рис 6)

```

mc[nelianjovu@nelianjovu-Aspire-5755G]:~/work/arch-pc/lab05
File Edit View Search Terminal Help
/home/nelianjovu/work/arch-pc/lab05/lab5-1.asm 1279/2709 47%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data          ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg      ; Длина переменной 'msg'

SECTION .bss           ; Секция не инициализированных данных
buf1: RESB 80          ; Буфер размером 80 байт

;----- Текст программы -----

SECTION .text          ; Код программы
GLOBAL _start          ; Начало программы
_start:                ; Точка входа в программу
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format10Quit

```

Рис. 3.6: Рис 6

Сначала я переведу текст программы lab5-1.asm в объектный файл. Затем линкую объектный файл и запускаю полученный исполняемый файл. Программа выведет строку “Введите строку:” и ожидает ввода с клавиатуры. При появлении запроса я введу свое полное имя(рис 7)

```

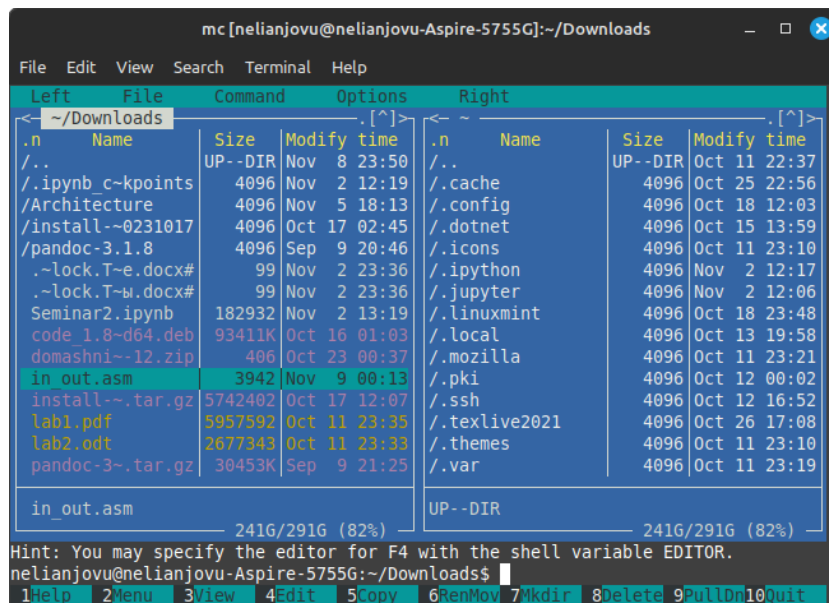
lab5-1.asm:20: error: label '_start' originally defined here
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Нелиа Нджову
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$

```

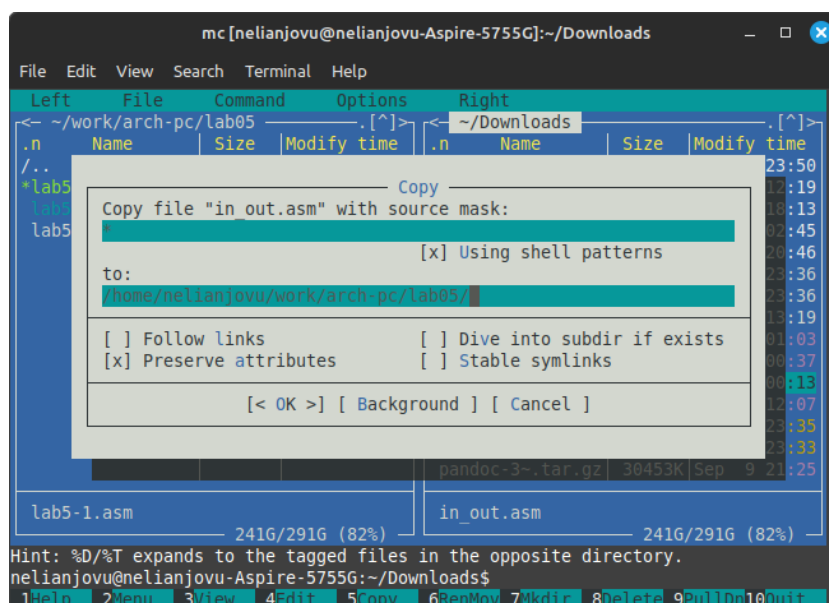
Рис. 3.7: Рис 7

### 3.3.Подключение внешнего файла in\_out.asm

Я скачаю файл in\_out.asm со страницы курса в ТУИС и сохраню его в каталоге «Downloads»(рис 8)



Функциональной клавишей F5 скопирую файл in\_out.asm из каталога Downloads в созданный каталог lab05(рис 9)



Функциональной клавишей F5 я скопирую файл lab5-1.asm в тот же каталог, но

с другим именем; для этого в появившемся окне mc я введу имя копии файла(рис 10)

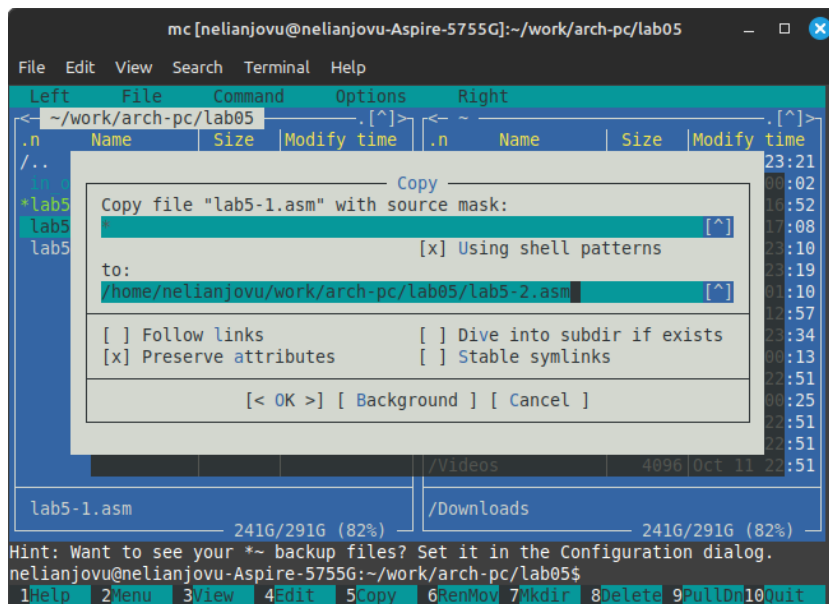
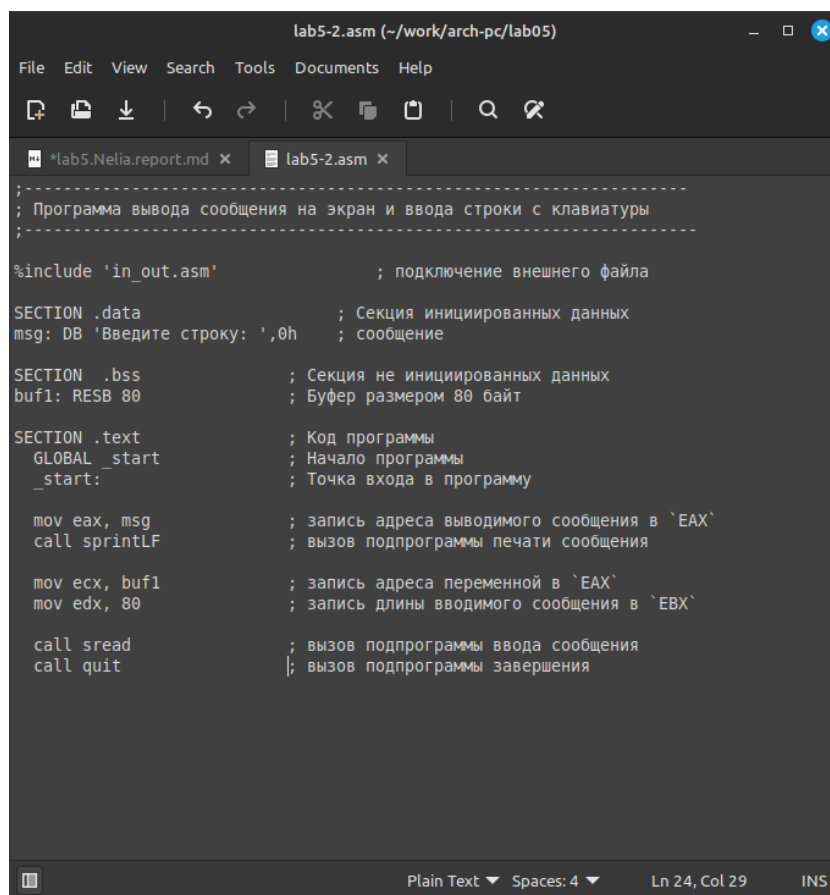


Рис. 3.10: Рис 10

Содержимое файла lab5-2.asm я изменю с помощью редактора, чтобы программа использовала подпрограммы из внешнего файла in\_out.asm(рис 11)



```
lab5-2.asm (~work/arch-pc/lab05)
File Edit View Search Tools Documents Help
*lab5.Nelia.report.md x lab5-2.asm x
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

%include 'in_out.asm'           ; подключение внешнего файла

SECTION .data                  ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h   ; сообщение

SECTION .bss                   ; Секция не инициализированных данных
buf1: RESB 80                  ; Буфер размером 80 байт

SECTION .text                  ; Код программы
GLOBAL _start                 ; Начало программы
_start:                       ; Точка входа в программу

mov eax, msg                   ; запись адреса выводимого сообщения в `EAX`
call sprintf                   ; вызов подпрограммы печати сообщения

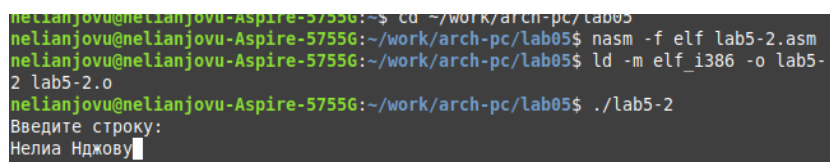
mov ecx, buf1                  ; запись адреса переменной в `EAX`
mov edx, 80                    ; запись длины вводимого сообщения в `EBX`

call sread                     ; вызов подпрограммы ввода сообщения
call quit                     ; вызов подпрограммы завершения

Plain Text Spaces: 4 Ln 24, Col 29 INS
```

Рис. 3.11: Рис 11

Я переведу текст программы lab5-2.asm в объектный файл. Затем линкую объектный файл и запускаю полученный исполняемый файл. Программа выведет строку “Введите строку:” и ожидает ввода с клавиатуры. При появлении запроса я введу свое полное имя(рис 12)

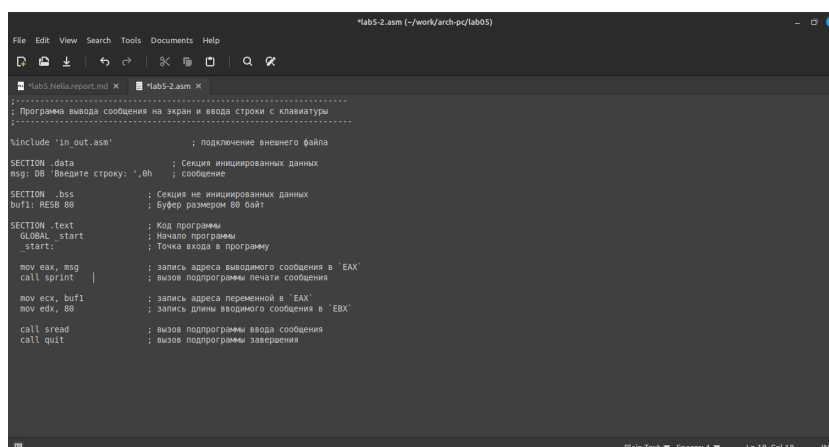


```
nelianjovu@nelianjovu-Aspire-5755G:~$ cd ~/work/arch-pc/lab05
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Нелиа Нджову
```

Рис. 3.12: Рис 12

Теперь я открою файл lab5-2.asm и поменяю место sprintf на printf. После этого я создам его исполняемый файл и запущу его, чтобы увидеть разницу(рис

13 и 14)



```
File Edit View Search Tools Documents Help
*lab5-2.asm (-/work/arch-pc/lab05)

;lab5-2.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include "in_out.asm" ; подключение внешнего файла

SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",0h ; сообщение

SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт

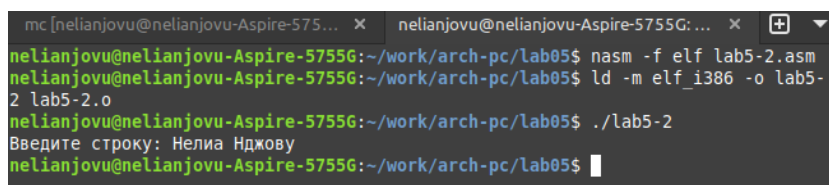
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprint ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'

call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.13: Рис 13



```
mc[nelianjovu@nelianjovu-Aspire-575... x nelianjovu@nelianjovu-Aspire-575G: ... x
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Нелиа Нджову
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab05$
```

Рис. 3.14: Рис 14

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2 заключается в том, что при запуске первого запрашивается перевод строки, а программа, запускающаяся при запуске второго, запрашивает ввод без перевода строки.

### 3.4 Задание для самостоятельной работы

1.Я создам копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5(рис 15)

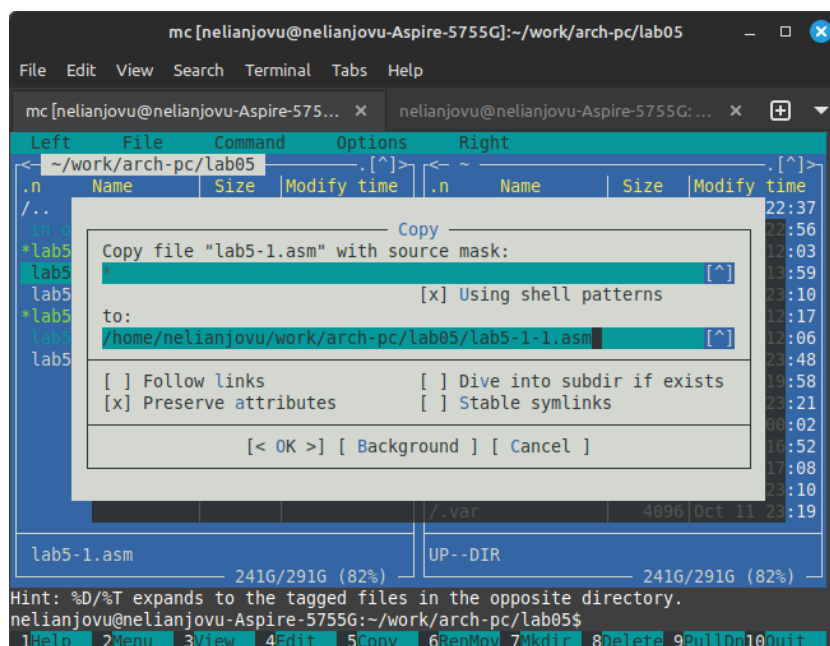


Рис. 3.15: Рис 15

Я открою созданный файл для редактирования. Я изменяю программу так, чтобы она помимо отображения подсказки и запроса на ввод отображала строку, введенную пользователем(рис 16)

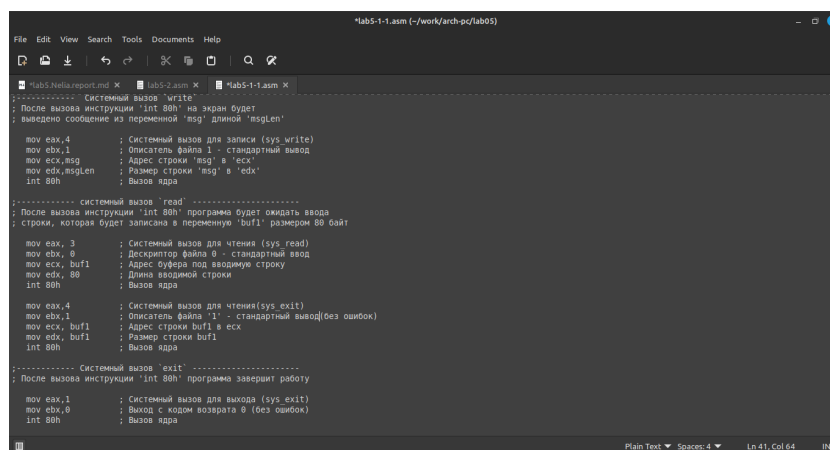
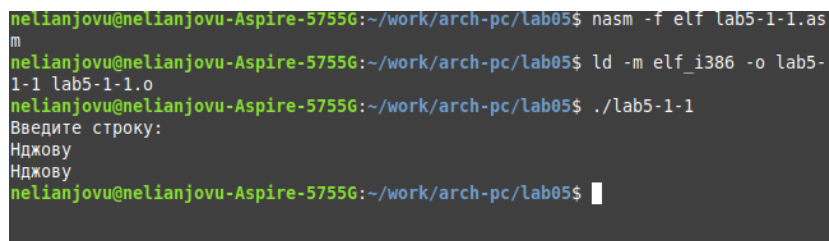


Рис. 3.16: Рис 16

2.Я переведу текст программы lab5-2.asm в объектный файл. Затем линкую объектный файл и запускаю полученный исполняемый файл.Программа выведет

строку “Введите строку:” и ожидает ввода с клавиатуры. При появлении запроса я введу свою фамилию(рис 17)



```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Нджову
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab05$
```

Рис. 3.17: Рис 17

Код программы из пункта 1:

```
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
mov eax,4 ; Системный вызов для чтения(sys_exit)
```



```

mov ebx,1 ; Описатель файла '1' - стандартный вывод(без ошибок)
mov ecx, buf1 ; Адрес строки buf1 в ecx
mov edx, buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

3.Я создам копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5(рис 18)

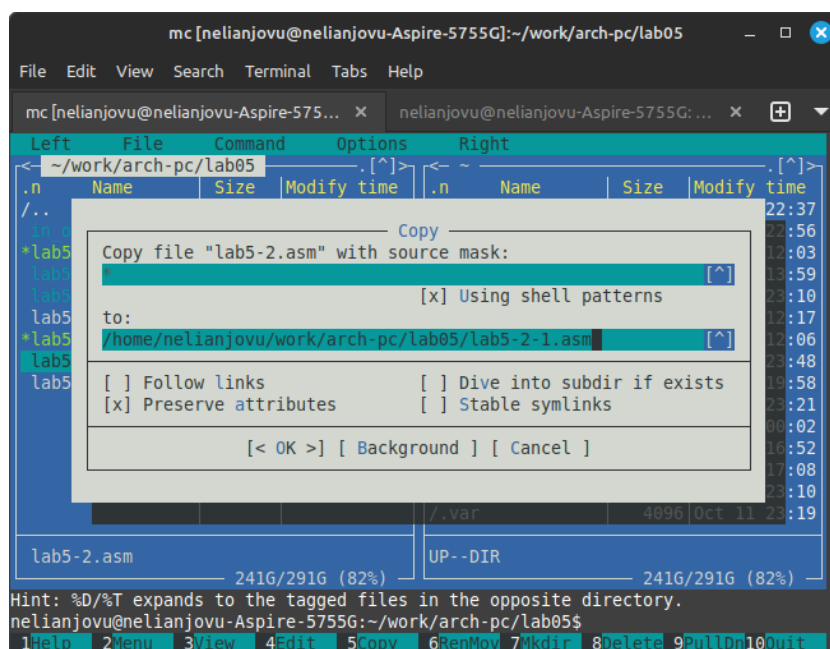


Рис. 3.18: Рис 18

Я открою созданный файл для редактирования. Я изменяю программу так, чтобы она помимо отображения подсказки и запроса на ввод отображала строку, введенную пользователем(рис 19)

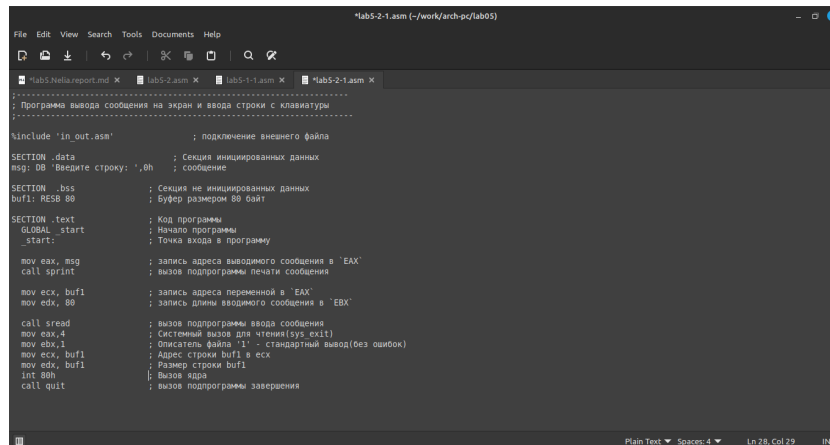


Рис. 3.19: Рис 19

4.Я создаю исполняемый файл и проверьте его работу(рис 20)

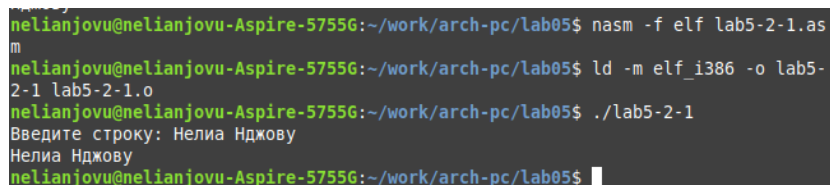


Рис. 3.20: Рис 20

Код программы из пункта 3:

```

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения

```

```
mov ecx, buf1 ; запись адреса переменной в `EAX`  
mov edx, 80 ; запись длины вводимого сообщения в `EBX`  
call sread ; вызов подпрограммы ввода сообщения  
mov eax, 4 ; Системный вызов для чтения(sys_exit)  
mov ebx, 1 ; Описатель файла '1' - стандартный вывод(без ошибок)  
mov ecx, buf1 ; Адрес строки buf1 в ecx  
mov edx, buf1 ; Размер строки buf1  
int 80h ; Вызов ядра  
call quit ; вызов подпрограммы завершения
```

## 4 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.

# Список литературы

Архитектура ЭВМ