Отчет по лабораторной работе No.5

Дисциплины: Архитектура компьютера

Нджову Нелиа

Содержание

# 1 Цель работы

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

# 2 Задание

1. Основы работы с mc
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)

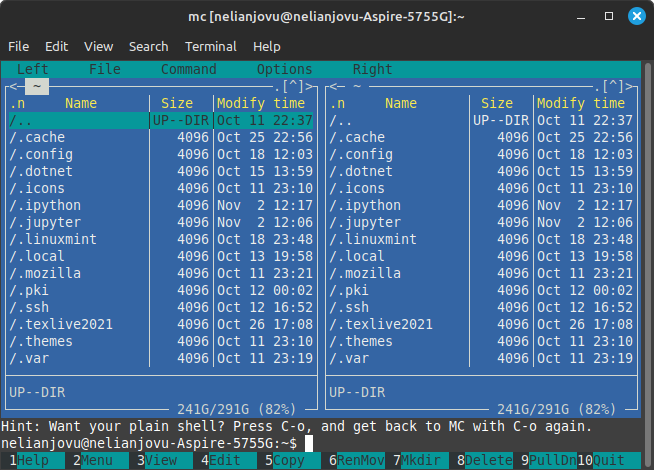


Рис 1

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

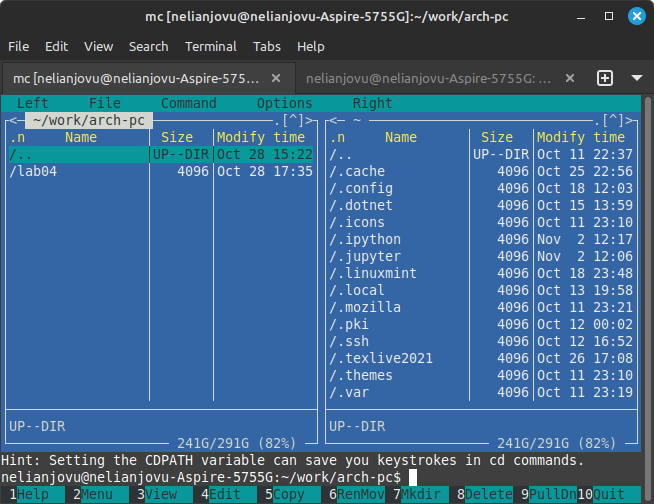


Рис 2

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

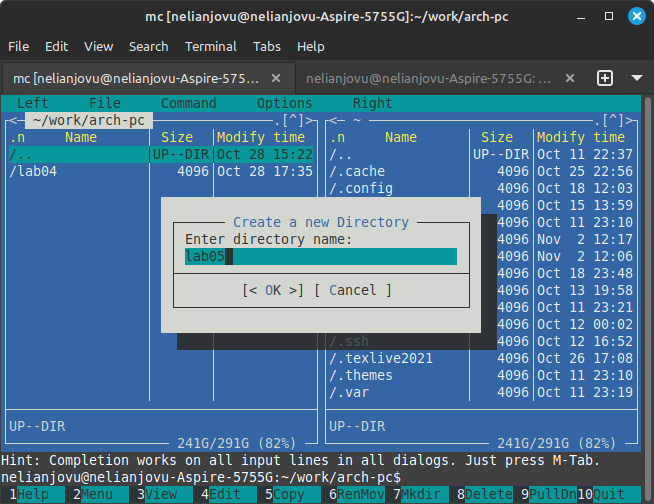


Рис 3

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

Рис 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)

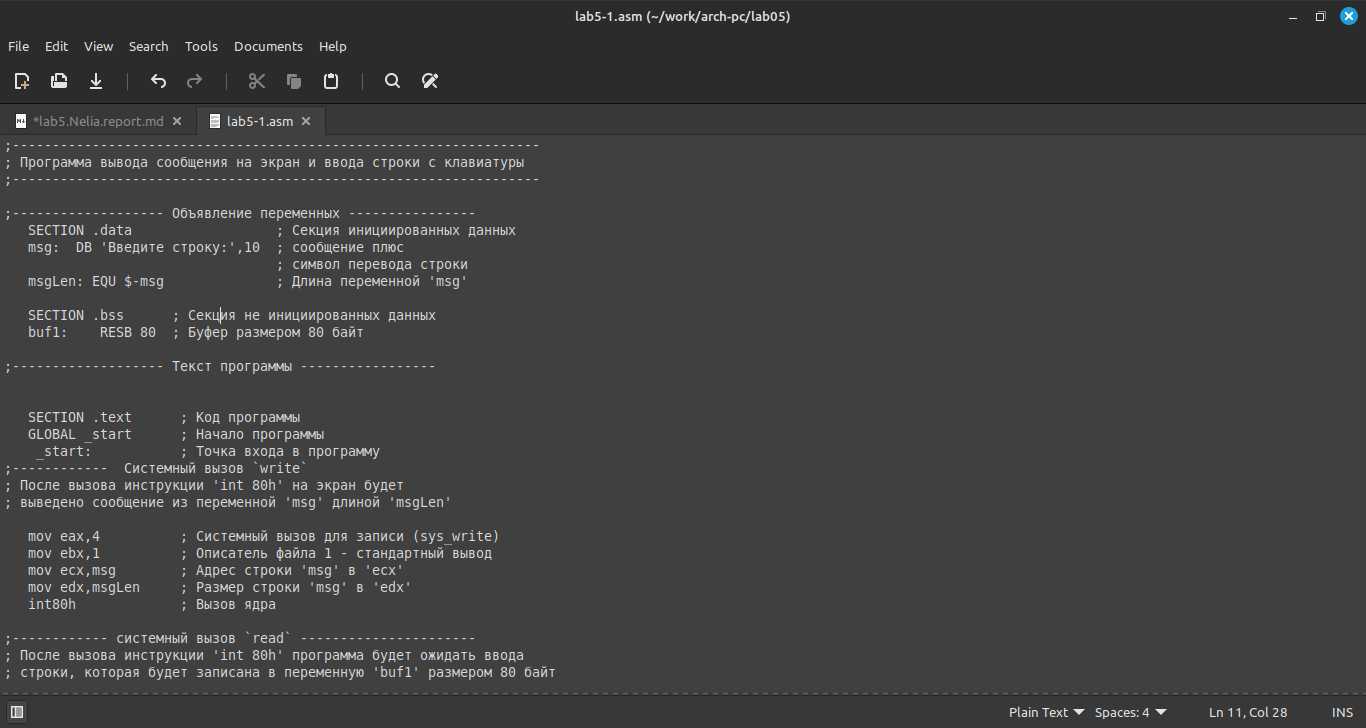


Рис 5

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

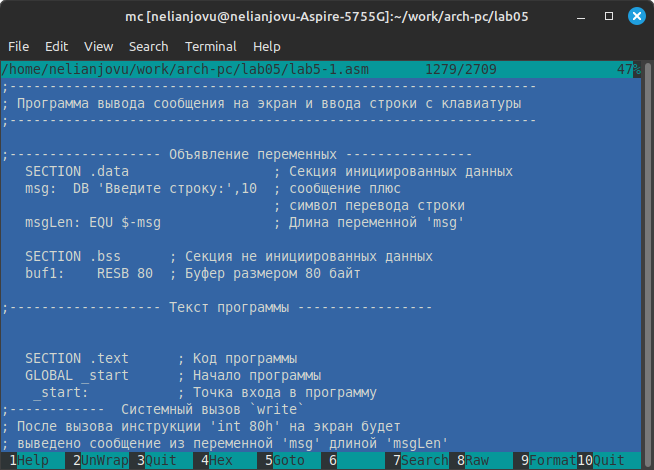


Рис 6

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)

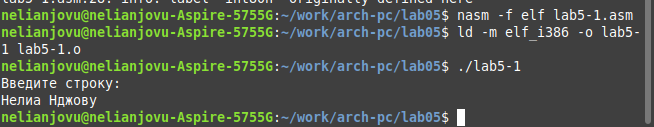


Рис 7

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)

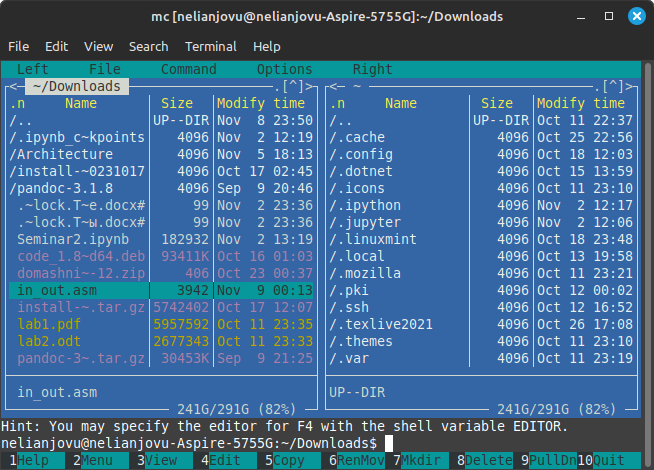


Рис 8

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

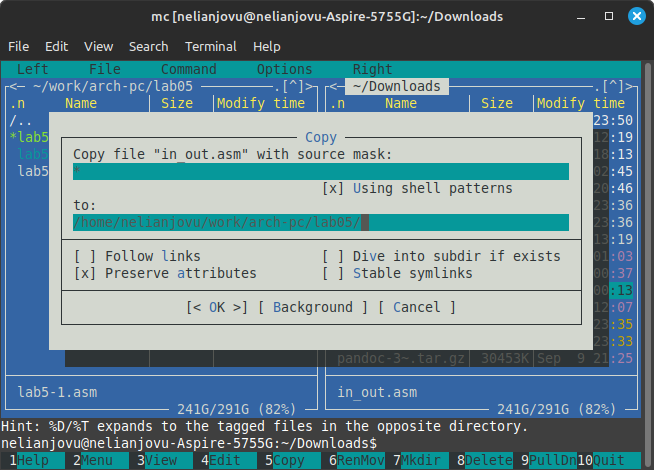


Рис 9

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)

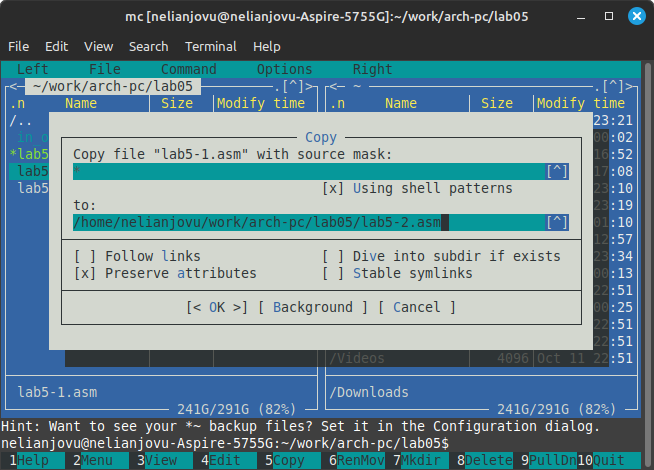


Рис 10

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

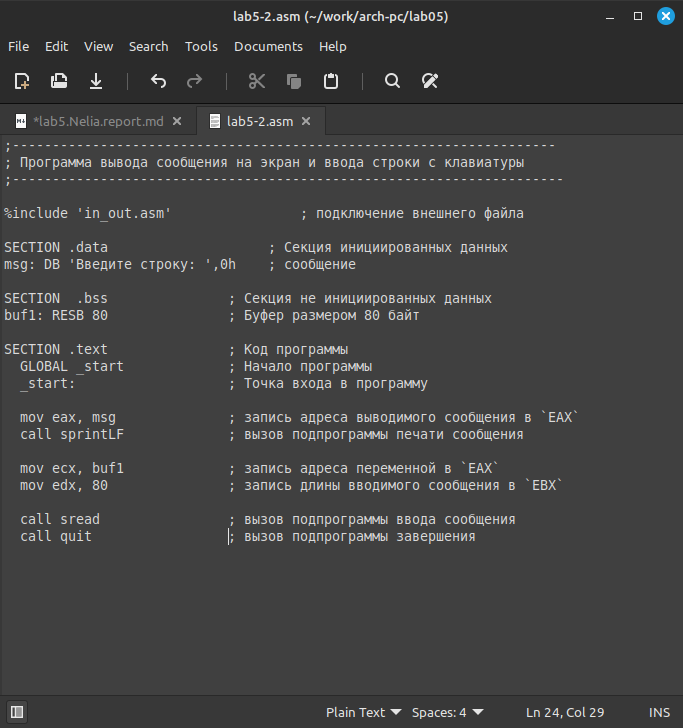


Рис 11

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)

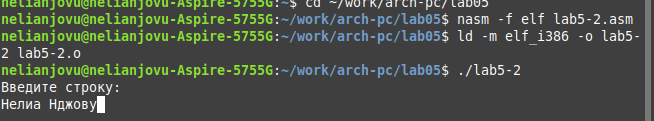


Рис 12

Теперь я открою файл lab5-2.asm и поменяю место sprintLF на sprint. После этого я создам его исполняемый файл и запущу его, чтобы увидеть разницу(рис 13 и 14)

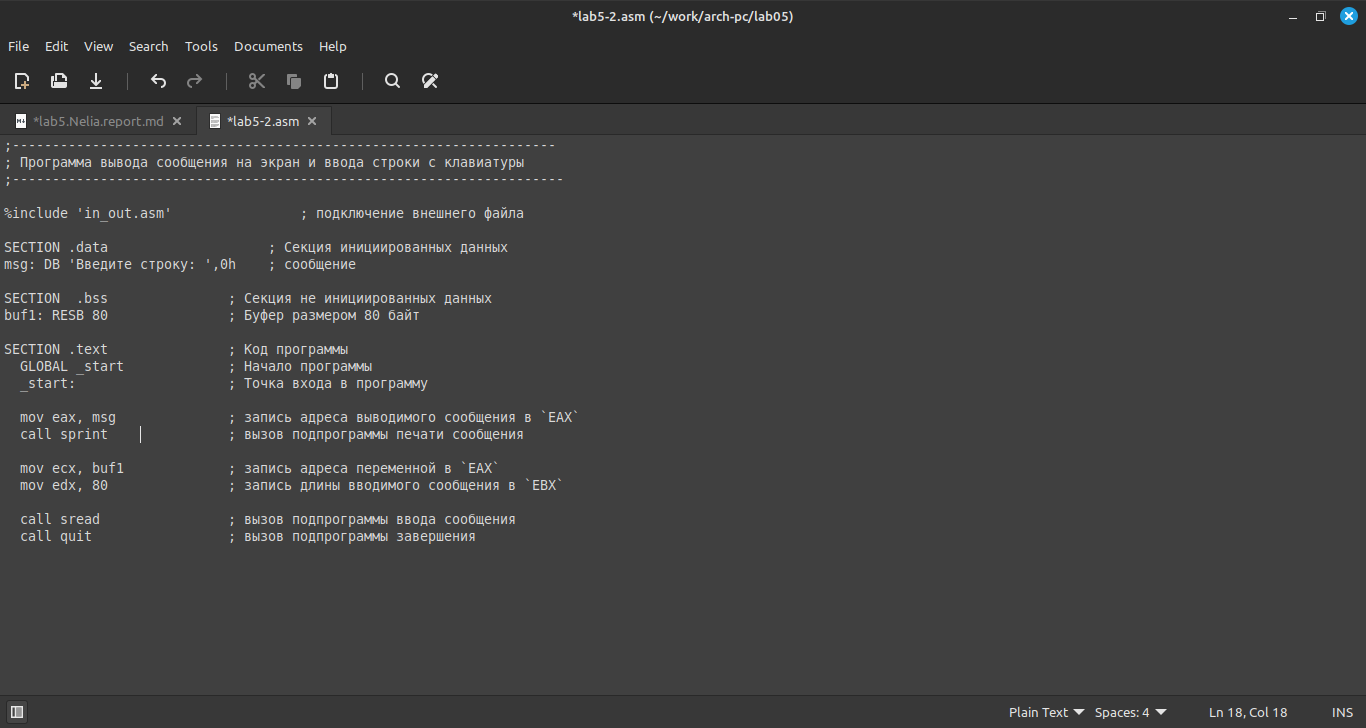


Рис 13

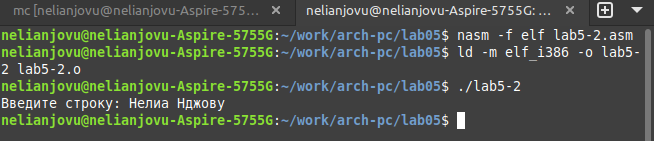


Рис 14

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)

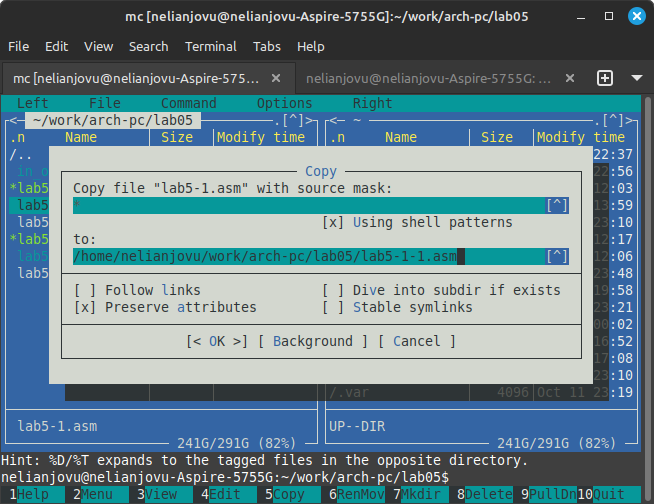


Рис 15

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)

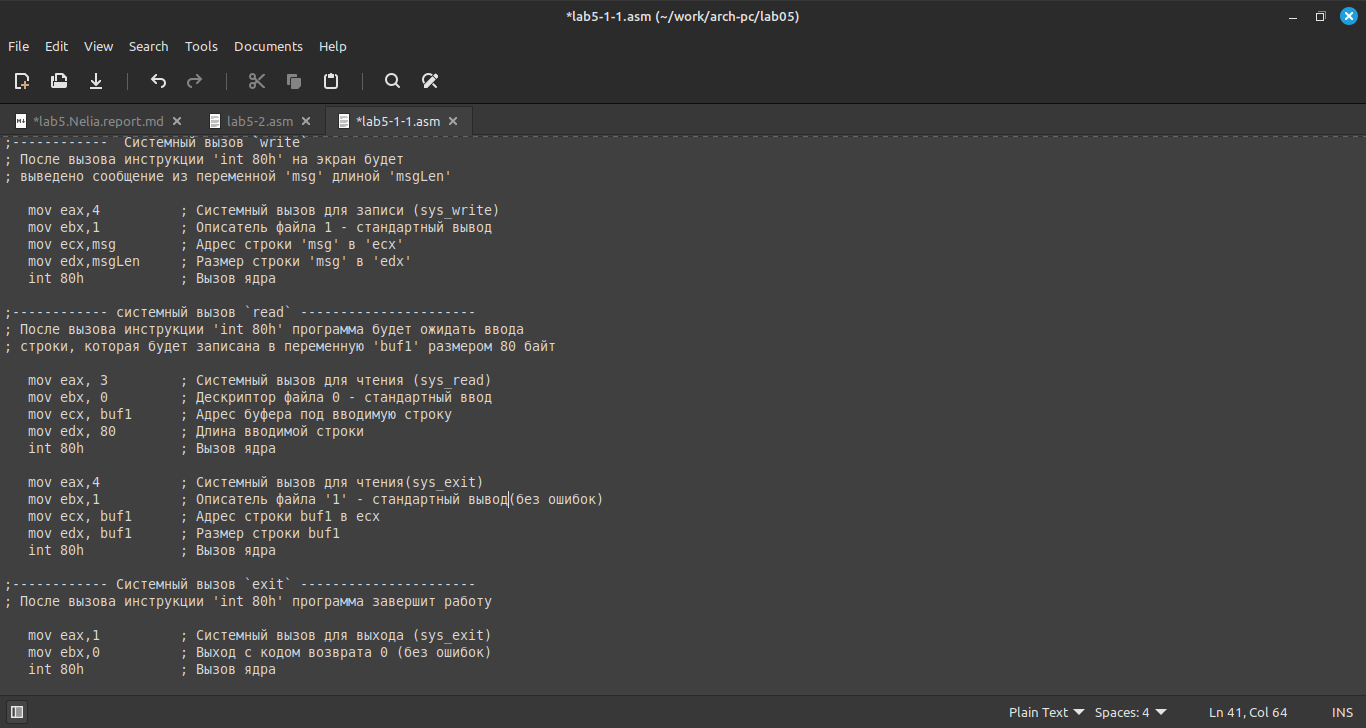


Рис 16

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 фамиилию(рис 17)

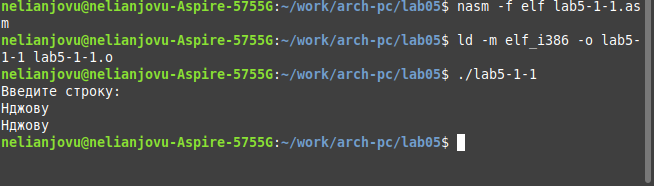


Рис 17

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 есх  
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)

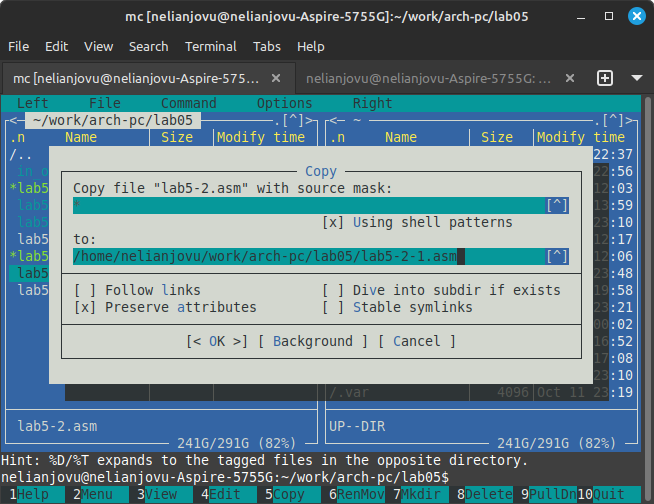


Рис 18

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)

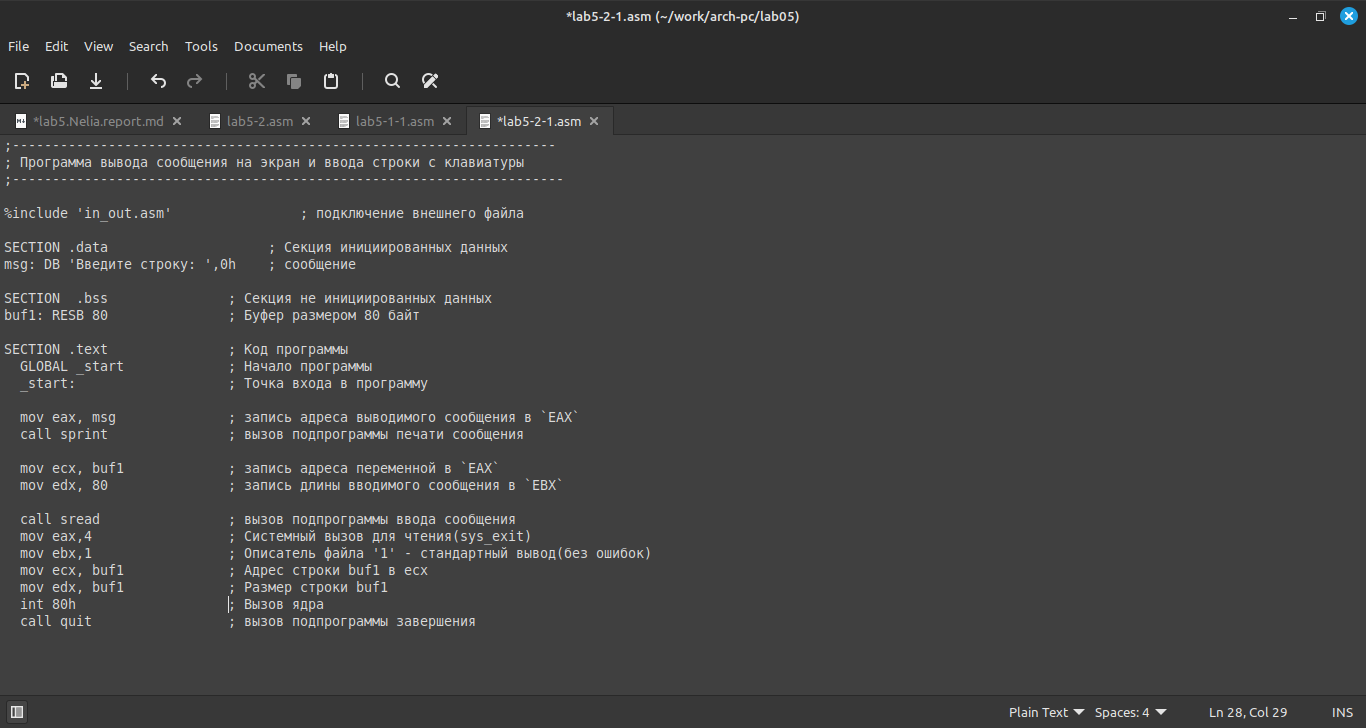


Рис 19

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

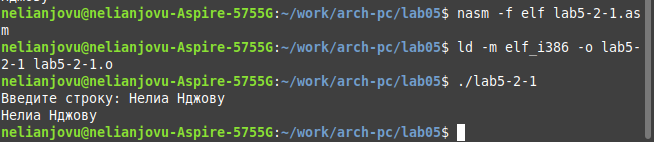


Рис 20

Код программы из пункта 3:

%include 'in\_out.asm' ; подключение внешнего файла  
SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку: ',0h ; сообщение  
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, 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 есх  
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