Отчет по лабораторной работе №6

*дисциплина: Архитектура компьютера*

Галацан Николай, НПИбд-01-22

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Ввожу mc, чтобы открыть *Midnight Commander*. Пользуясь клавишами **вверх** , **вниз** и **Enter** перехожу в каталог, созданный при выполнении лабораторной работы №5. С помощью функцинальной клавиши **F7** создаю папку и перехожу в созданный каталог (рис. 1).

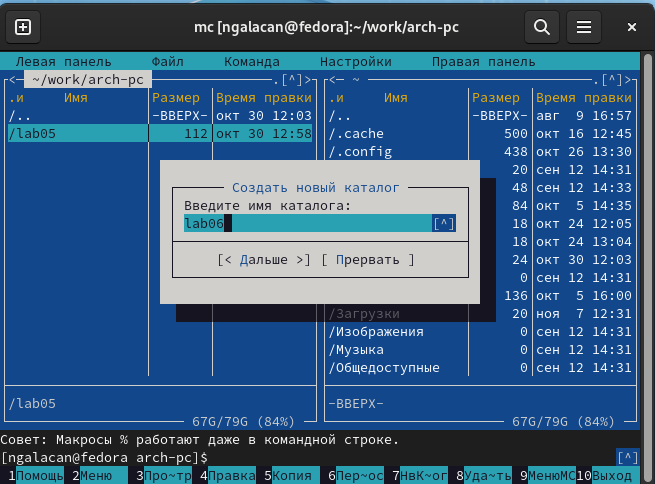


Рис. 1: Создание каталога в mc

Пользуясь строкой ввода и командой touch создаю файл lab6-1.asm (рис. 2).

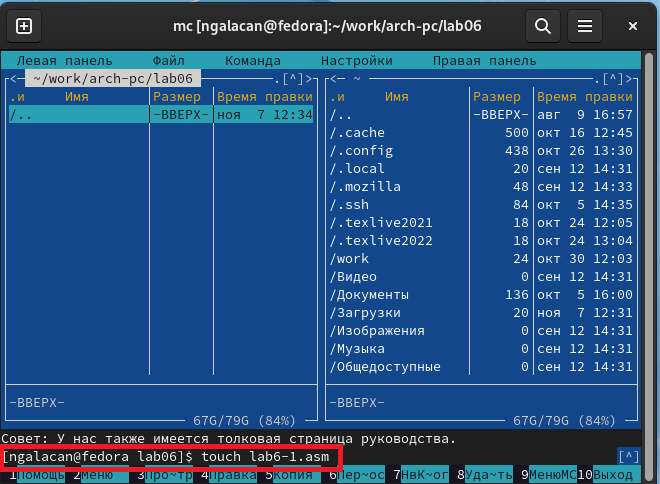


Рис. 2: Создание файла в mc

С помощью функциональной клавиши **F4** открываю файл для редактирования во встроенном редакторе (mcedit) и ввожу текст программы (рис. 3).

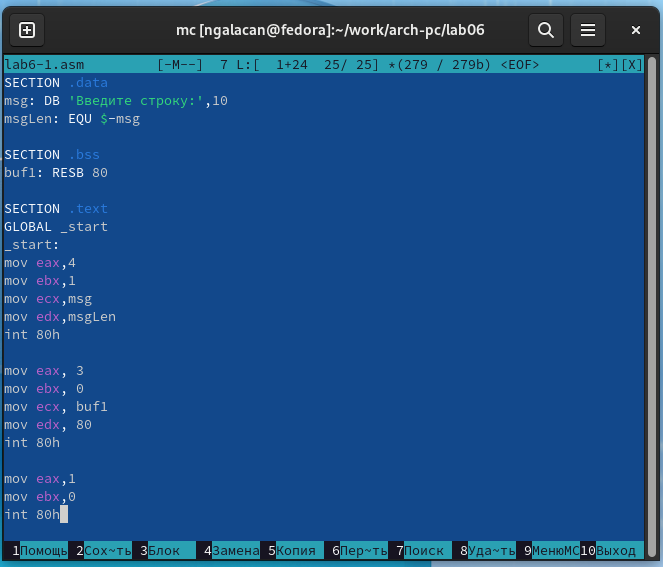


Рис. 3: Редактирование файла программы

Нажимаю **F2** для сохранения изменений, **F10** - для выхода из редактора. С помощью **F3** открываю файл для просмотра и убеждаюсь, что текст программы сохранен (рис. 4).

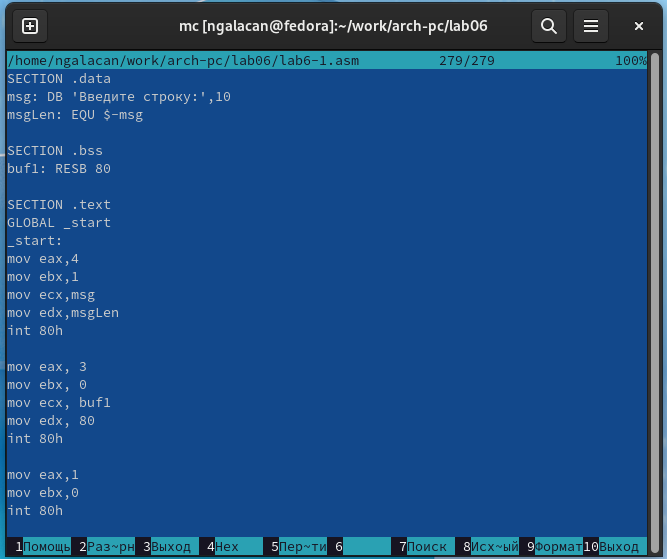


Рис. 4: Просмотр файла программы в mc

Для скрытия и отображения панелей использую комбинацию клавиш **Ctrl+o**. Транслирую текст программы lab6-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. Ввожу ФИО (рис. 5).

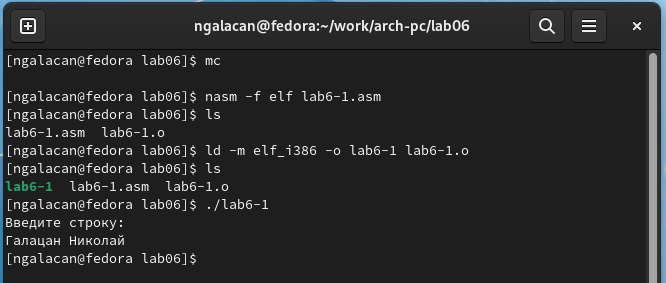


Рис. 5: Трансляция, компоновка, запуск исполняемого файла lab 6-1

Со страницы курса скачиваю файл in\_out.asm. Во второй панели перехожу в папку “Загрузки”. Копирую файл в рабочую папку с помощью **F5** (рис. 6).

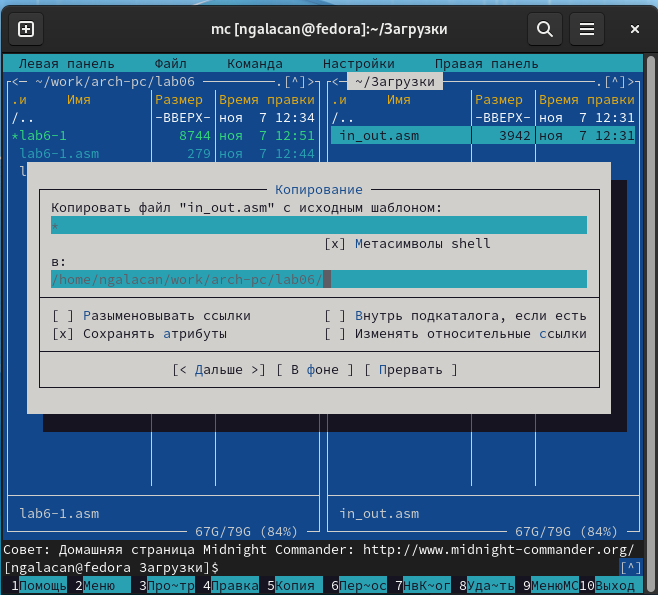


Рис. 6: Копирование файла in\_out.asm

С помощью функциональной клавиши **F6** создаю копию файла lab6-1.asm с именем lab6-2.asm (рис. 7).

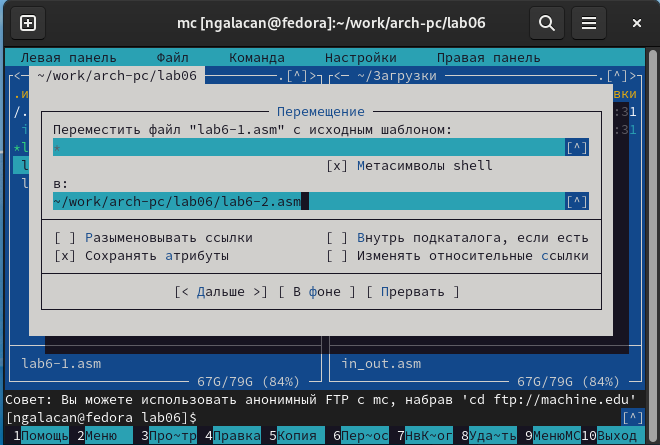


Рис. 7: Создание копии файла lab6-1.asm

Обнаруживаю, что файл был не скопирован, а только переименован, так как **F6** отвечает за перемещение файлов. Для того, чтобы сохранить исходный файл, создаю его копию с именем lab6-1.asm с помощью **F5** (рис. 8).

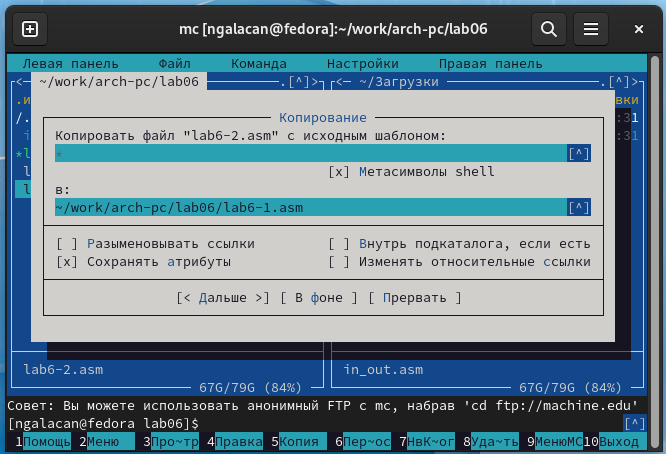


Рис. 8: Копирование файла под другим именем

Исправляю текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 6.2 (рис. 9). Создаю исполняемый файл и проверяю его работу (рис. 10).

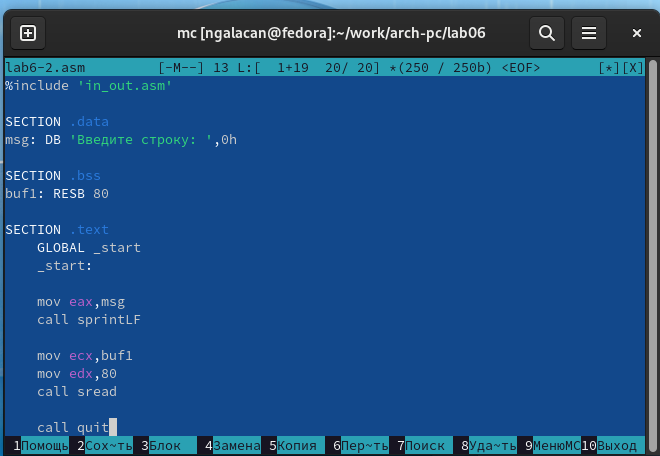


Рис. 9: Редактирование текста программы lab6-2.asm

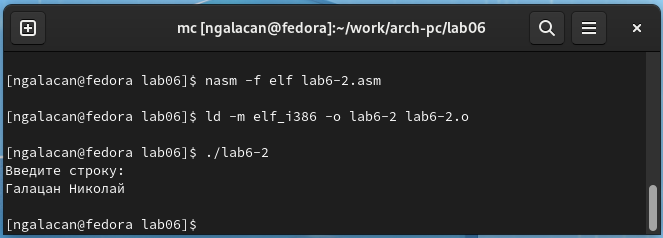


Рис. 10: Трансляция, компоновка, запуск исполняемого файла lab6-2

Программы работают аналогично, но преимуществами второй являются удобство и читаемость.

В файле lab6-2.asm заменяю подпрограмму sprintLF на sprint. Создаю исполняемый файл и запускаю (рис. 11).

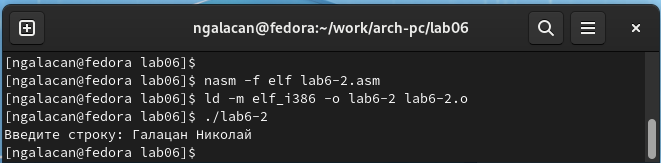


Рис. 11: Трансляция, компоновка, запуск исполняемого файла lab6-2 после замены подпрограмм

Разница в том, что после выведения приглашения ввести строку программа не производит перенос строки. За это отвечает sprintLF, которая была заменена.

# 3 Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab6-1.asm под новым именем lab6-1-2.asm (рис. 12).

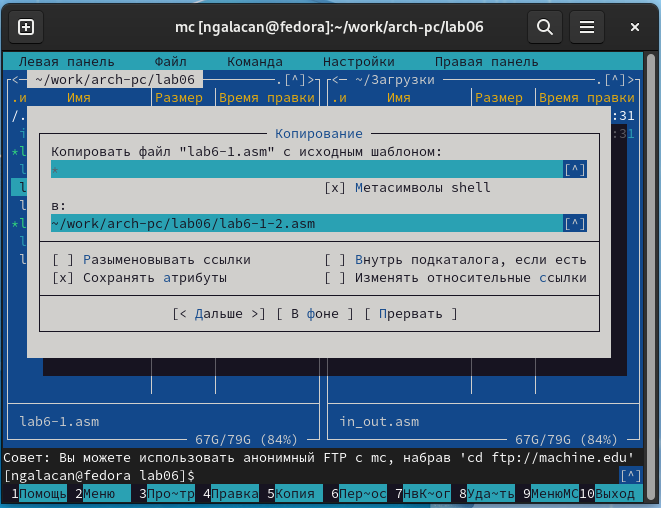


Рис. 12: Копирование файла под новым именем в mc

Вношу изменения в программу, чтобы она работала по следующему алгоритму:

* вывести приглашение типа “Введите строку:”;
* ввести строку с клавиатуры;
* вывести введённую строку на экран.

Для этого перед завершением программы прописываю вывод введенной строки:

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

Измененная программа lab6-1-1.asm выглядит следующим образом:

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

1. Создаю исполняемый файл и запускаю (рис. 13).

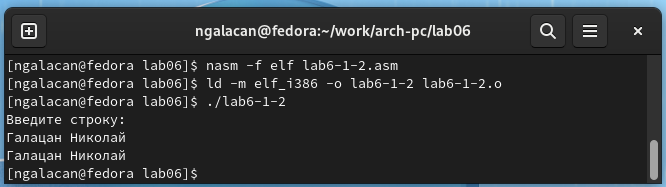


Рис. 13: Запуск исполняемого файла lab6-1-2

Программа запускается успешно и выводит ранее введенную строку.

1. Создаю копию файла lab6-2.asm под новым именем lab6-2-2.asm (рис. 14).

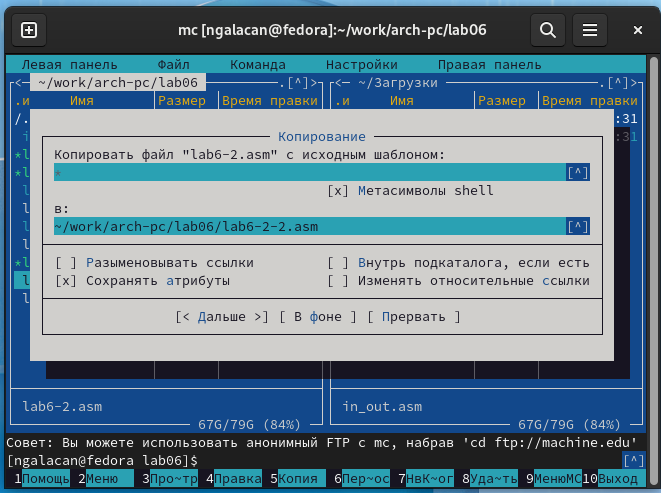


Рис. 14: Копирование файла под новым именем в mc

Вношу изменения в программу с использованием in\_out.asm, чтобы она работала по следующему алгоритму:

* вывести приглашение типа “Введите строку:”;
* ввести строку с клавиатуры;
* вывести введённую строку на экран.

Для этого перед call quit прописываю вывод введенной строки:

mov eax,buf1  
call sprintLF

Измененная программа lab6-2-2.asm выглядит следующим образом:

%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 sprintLF  
  
 mov ecx,buf1  
 mov edx,80  
 call sread  
  
 mov eax,buf1  
 call sprintLF  
  
 call quit

1. Создаю исполняемый файл и запускаю (рис. 15).

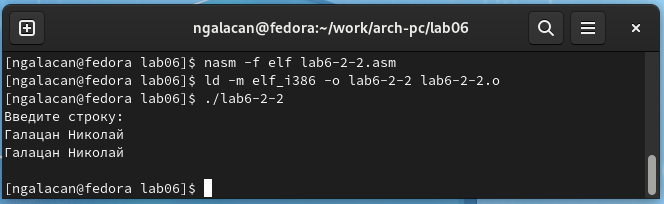


Рис. 15: Запуск исполняемого файла lab6-2-2

Программа работает аналогично и выводит ранее введенную строку, а так же перенос строки, так как я использовал подпрограмму sprintLF.

# 4 Выводы

На данной лабораторной работы были приобретены практические навыки работы в *Midnight Commander*. Изучена структура программы на языке ассемблера NASM, освоены инструкции языка ассемблера mov и int.