

Отчет по лабораторной работе №5

Дисциплина: Архитектура компьютера

Маньковская Дарья Станиславовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка 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 Теоретическое введение

Midnight Commander (или просто `mc`) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. `mc` является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (`SECTION .text`), секция инициированных (известных во время компиляции) данных (`SECTION .data`) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (`SECTION .bss`). Для объявления инициированных данных в секции `.data` используются директивы `DB`, `DW`, `DD`, `DQ` и `DT`, которые резервируют память и указывают, какие значения должны храниться в этой памяти: - `DB` (define byte) — определяет переменную размером в 1 байт; - `DW` (define word) — определяет переменную размеров в 2 байта

(слово); - DD (define double word) — определяет переменную размером в 4 байта (двойное слово); - DQ (define quad word) — определяет переменную размером в 8 байт (учетверённое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

Здесь операнд dst — приёмник, а src — источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

```
int n
```

Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

4 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc (рис. 1).

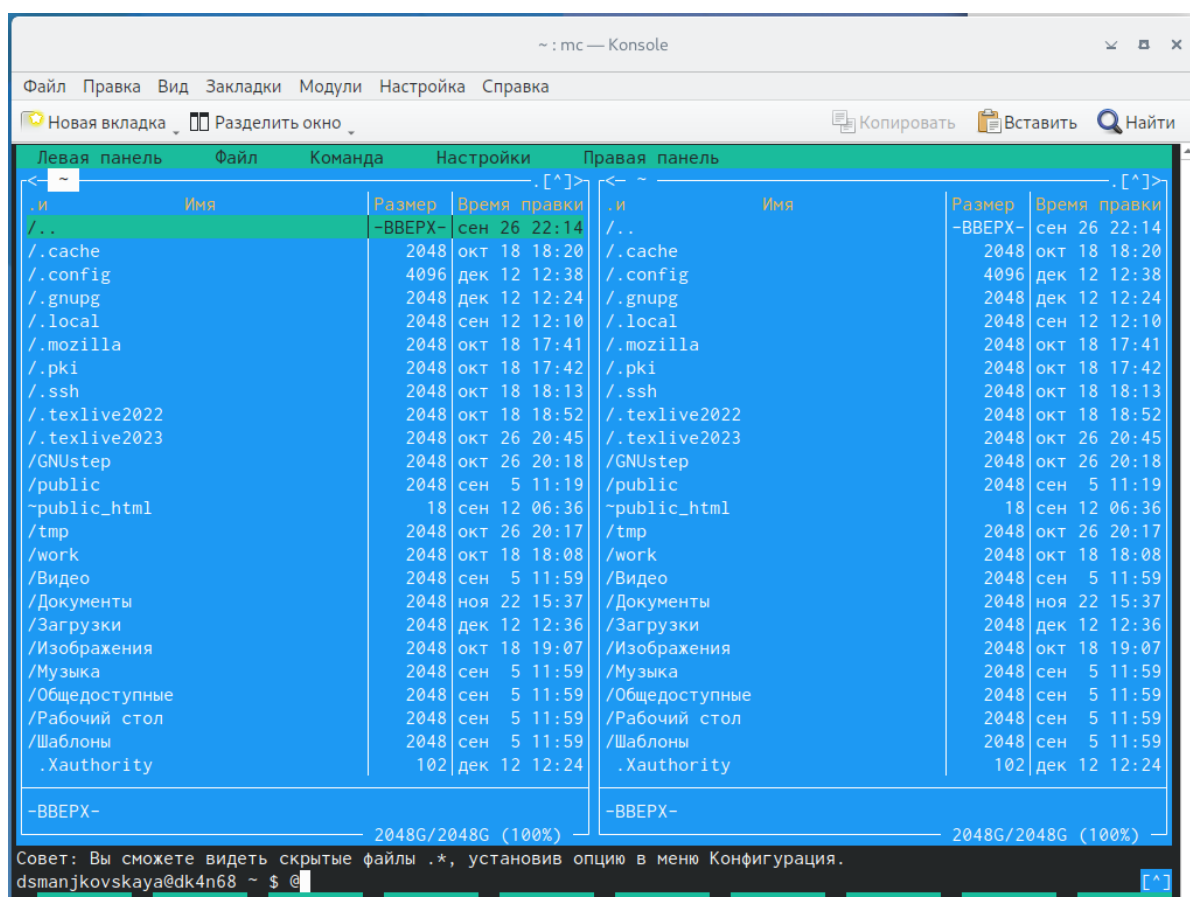


Рис. 1: Открытый mc

Перехожу в каталог ~/work/study/2022-2023/Архитектура Компьютера/arch-рс, используя файловый менеджер mc. Перехожу в каталог lab05 (рис. 2).

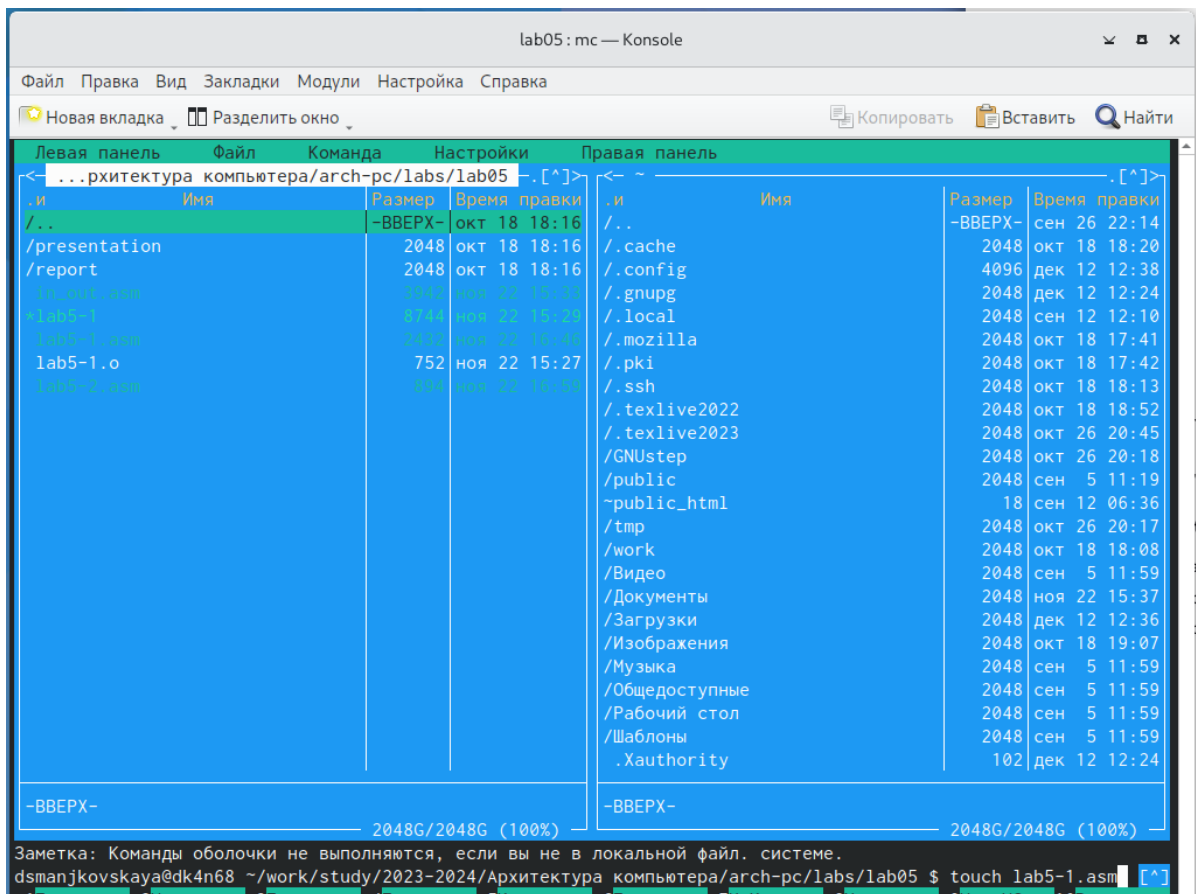


Рис. 2: Перемещение между директориями

В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать. С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano (рис. 3).

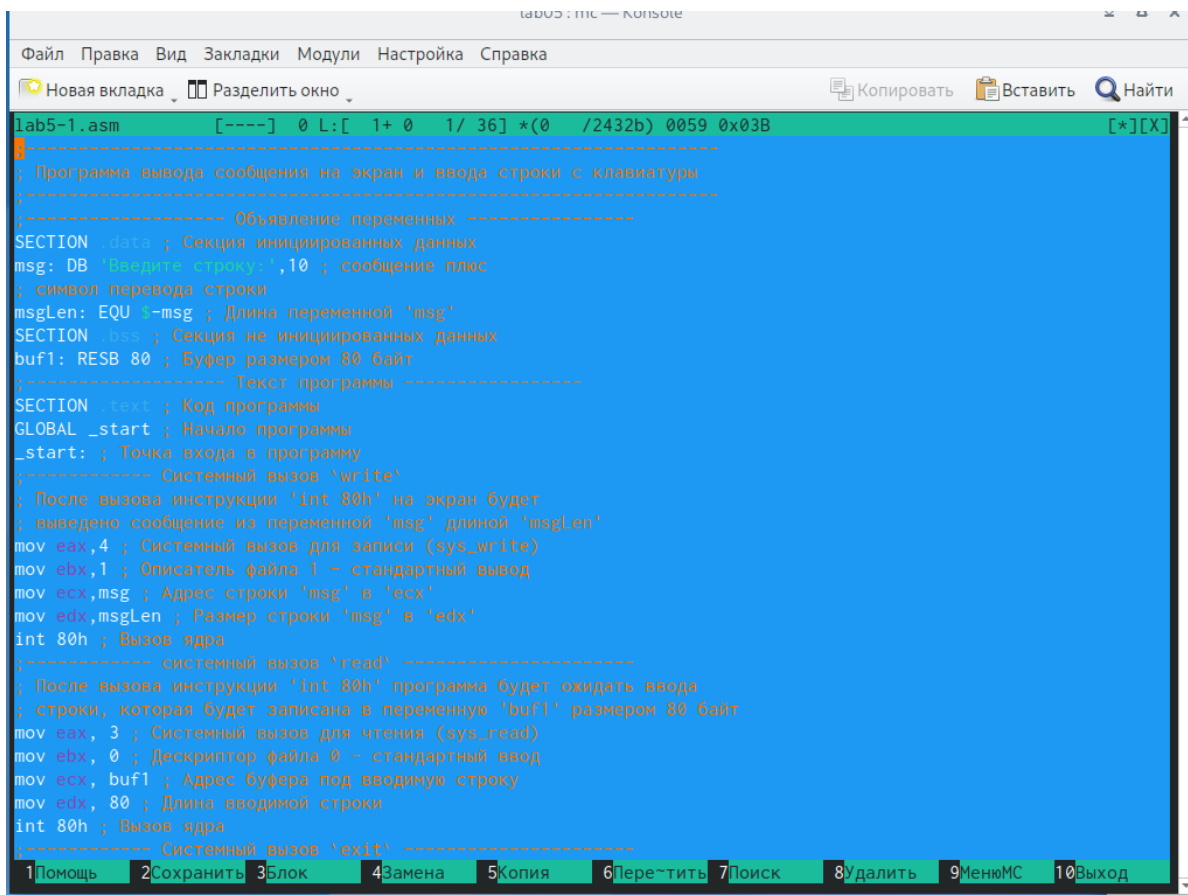


Рис. 3: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 4).
Далее выхожу из файла (Ctrl+X), сохраняя изменения (Y, Enter).

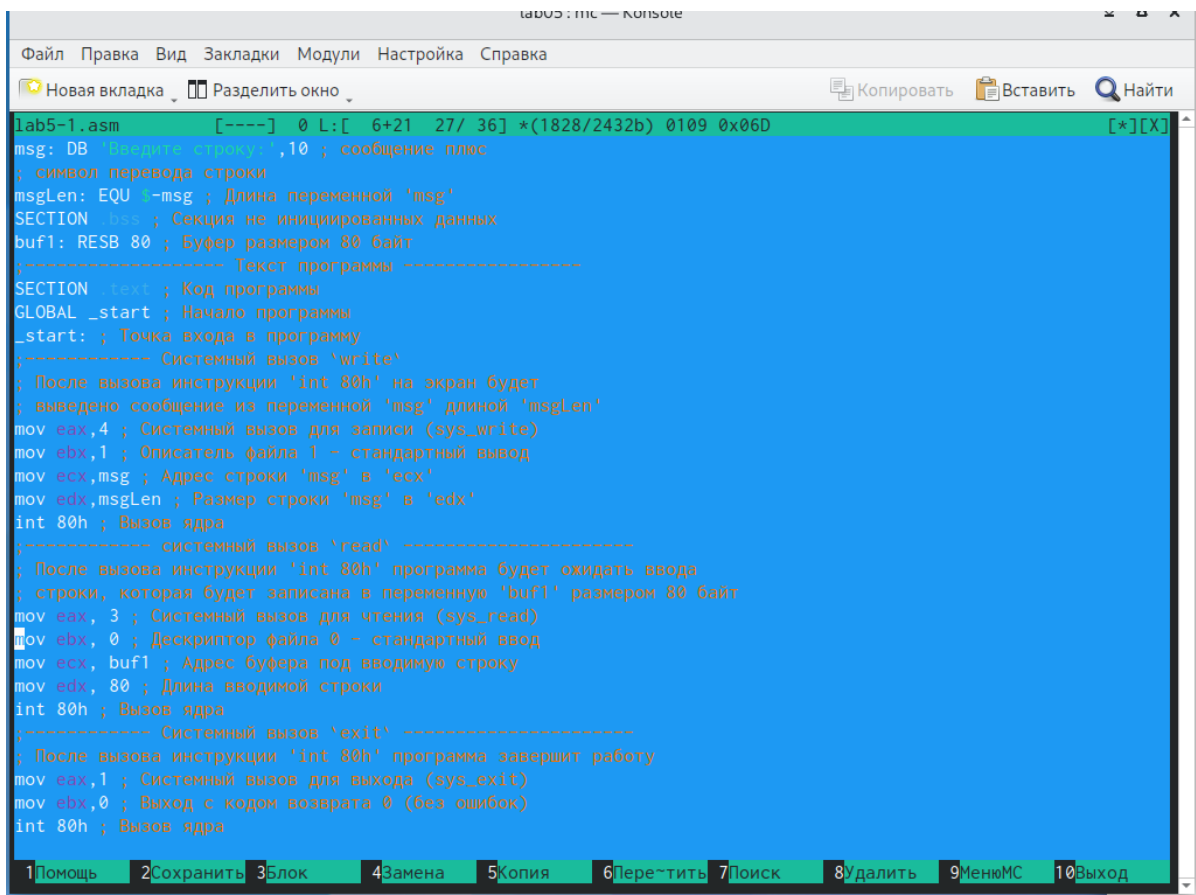


Рис. 4: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы.

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Создался исполняемый файл `lab5-1`. Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 5).

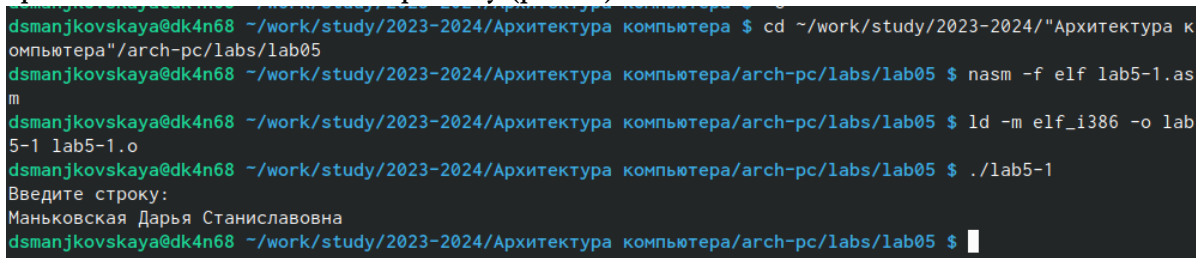


Рис. 5: Исполнение файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. 6).

in_out.asm	3942	ноя 22 15:33
report.docx	271578	окт 26 20:28
Л02_Маньковская_отчет.pdf	1470002	окт 18 17:46
Л03_Маньковская_отчет.docx	1334555	окт 18 20:40

Рис. 6: Скачанный файл

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 7).

Копирование

Копировать файл "in_out.asm" с исходным шаблоном:

>

[x] Метасимволы shell

в:

итектура компьютера/arch-pc/lab05/

☐ Разыменовывать ссылки

☒ Сохранять атрибуты

☐ Внутрь подкаталога, если есть

☐ Изменять относительные ссылки

[< Дальше >]

[В фоне]

[Прервать]

Рис. 7: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. 8).

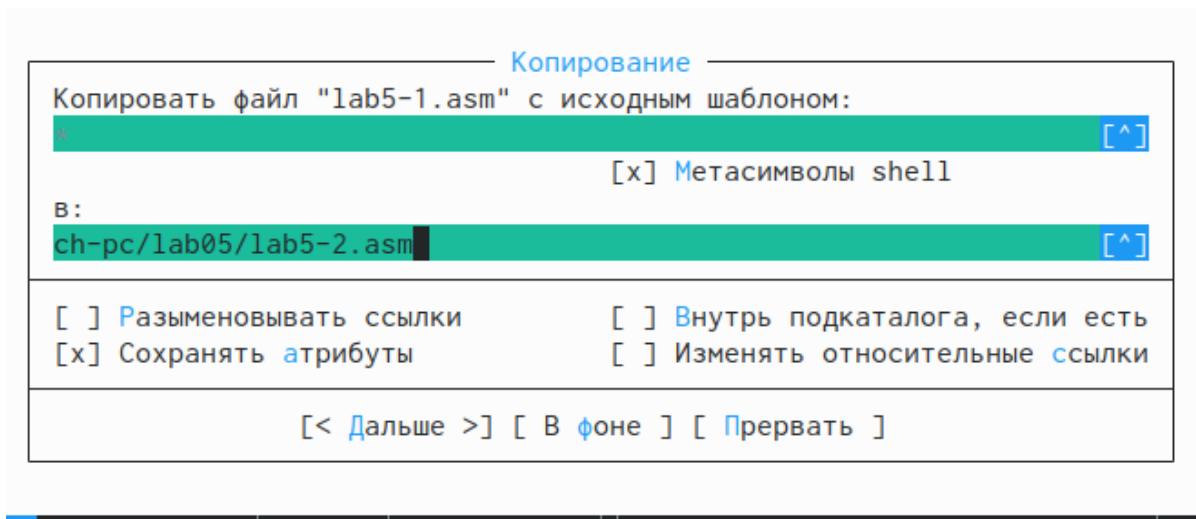


Рис. 8: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano (рис. 9), чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.

```
lab5-2.asm [----] 22 L:[ 1+ 8 9/ 15] *(460 / 894b) 0032 0x020
#include "in_out.asm"
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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; длина вводимой строки в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 9: Редактирование файла

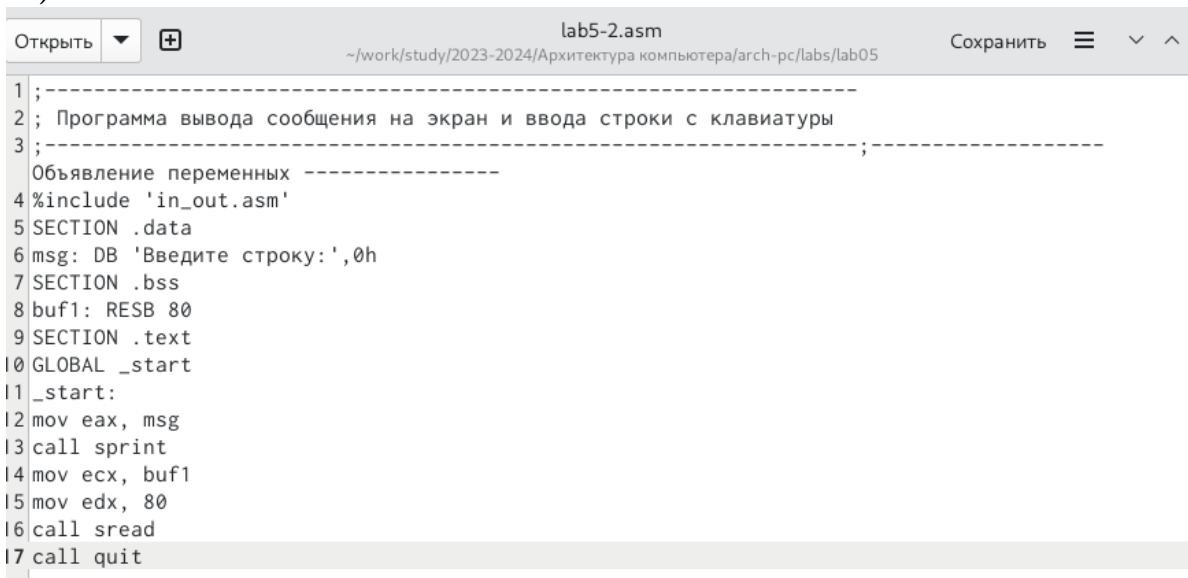
Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл lab5-2.o. Выполняю компоновку

объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o` Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл (рис. 10).

```
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/1
abs/lab05 $ nasm -f elf lab5-2.asm
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/1
abs/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/1
abs/lab05 $ ./lab5-2
Введите строку:
Маньковская Дарья Станиславовна
```

Рис. 10: Исполнение файла

Открываю файл `lab5-2.asm` для редактирования в nano функциональной клавишей F4. Изменяю в нем подпрограмму `sprintLF` на `sprint`. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 11).



```
1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----;-----
4 %include 'in_out.asm'
5 SECTION .data
6 msg: DB 'Введите строку:',0h
7 SECTION .bss
8 buf1: RESB 80
9 SECTION .text
10 GLOBAL _start
11 _start:
12 mov eax, msg
13 call sprint
14 mov ecx, buf1
15 mov edx, 80
16 call sread
17 call quit
```

Рис. 11: Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 12).

```
dsmanjkovskaya@dk4n68 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/  
arch-pc/labs/lab05  
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/  
labs/lab05 $ nasm -f elf lab5-2.asm  
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/  
labs/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o  
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/  
labs/lab05 $ ./lab5-2  
Введите строку:Маньковская Дарья Станиславовна
```

Рис. 12: Запуск исполняемого файла

Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 13).

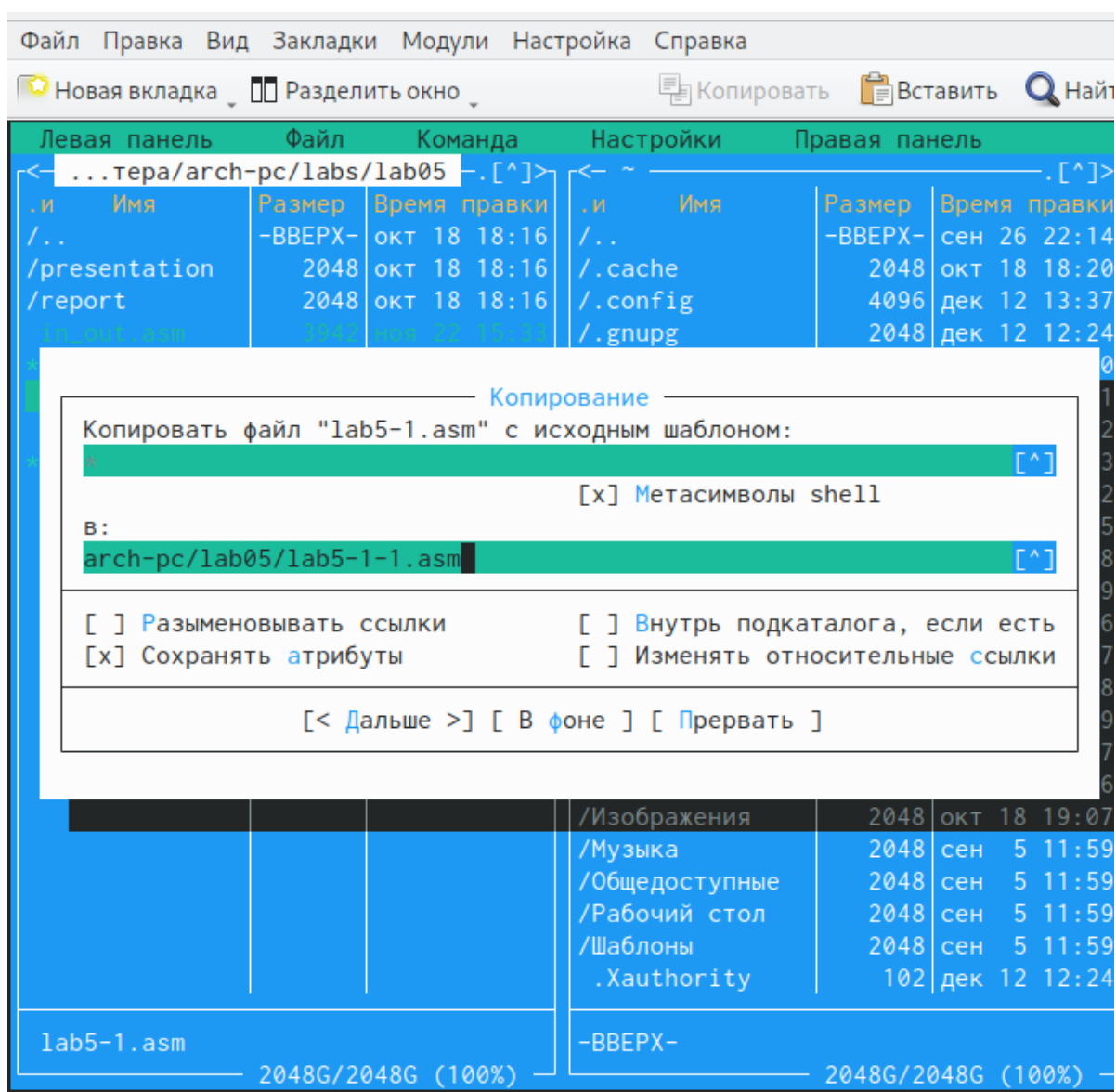
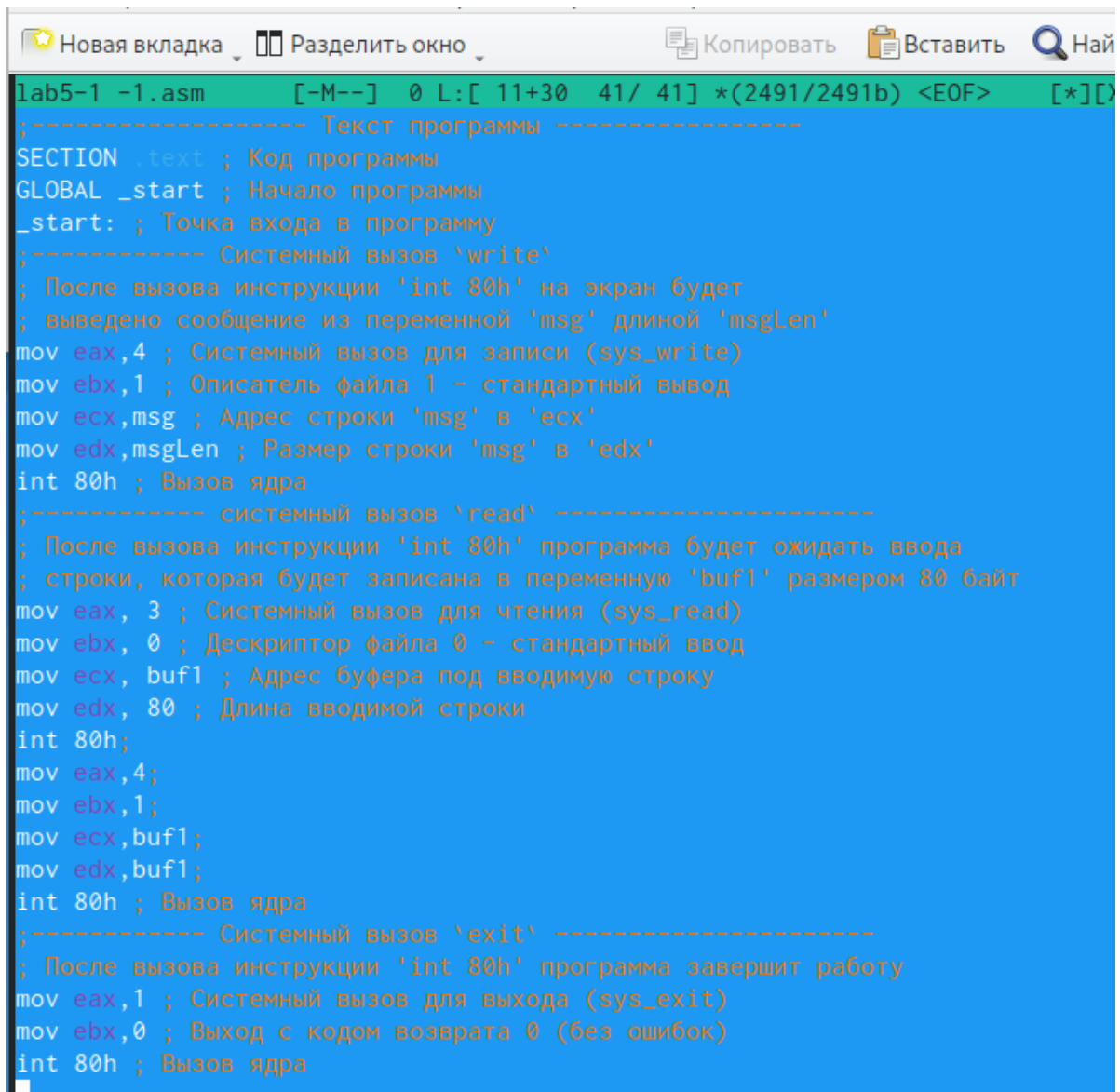


Рис. 13: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 14).



```
Новая вкладка  Разделить окно  Копировать  Вставить  Най
lab5-1 -1.asm  [-M--]  0 L:[ 11+30  41/ 41]  *(2491/2491b)  <EOF>  [*][>
;----- 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'
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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
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;
mov eax,4;
mov ebx,1;
mov ecx,buf1;
mov edx,buf1;
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 14: Редактирование файла

Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 15).

```

dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-1-1.asm
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1-1
Введите строку:
Маньковская Дарья Станиславовна
Маньковская Дарья Станиславовна
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $

```

Рис. 15: Исполнение файла

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 16).

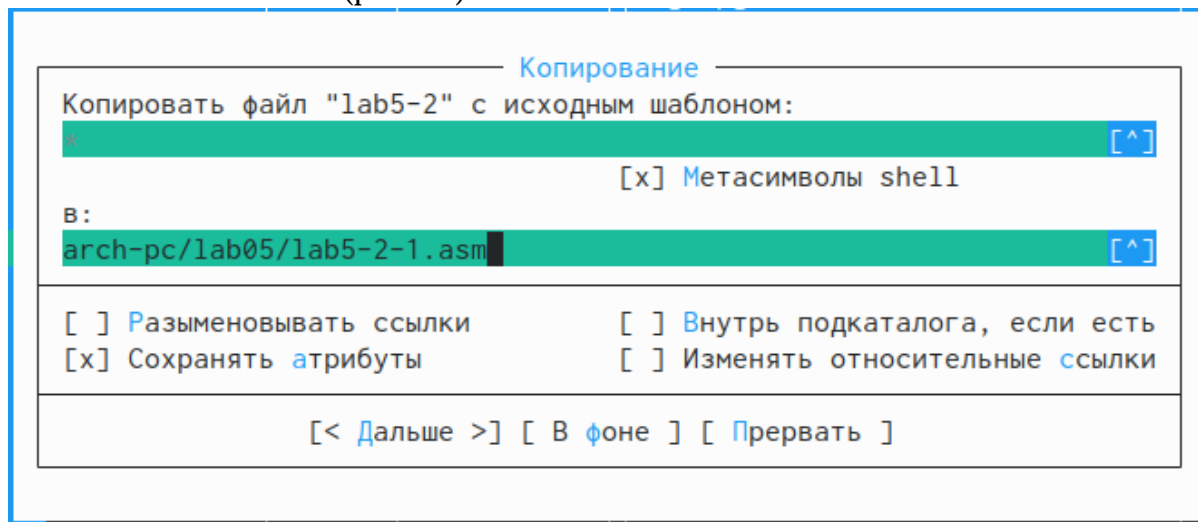


Рис. 16: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 17).


```

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
#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, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit

```

Рис. 17: Редактирование файла

Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 18).

```

dsmanjkovskaya@dk4n68 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab05
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-2-1.asm
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-2-1
Введите строку:Маньковская Дарья Станиславовна
Маньковская Дарья Станиславовна
dsmanjkovskaya@dk4n68 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $

```

Рис. 18: Исполнение файла

5 Выводы

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

6 Список литературы

Архитектура ЭВМ