

Отчет по лабораторной работе №5

Дисциплина: Архитектура компьютера

Белоусова Елизавета Валетиновна

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).

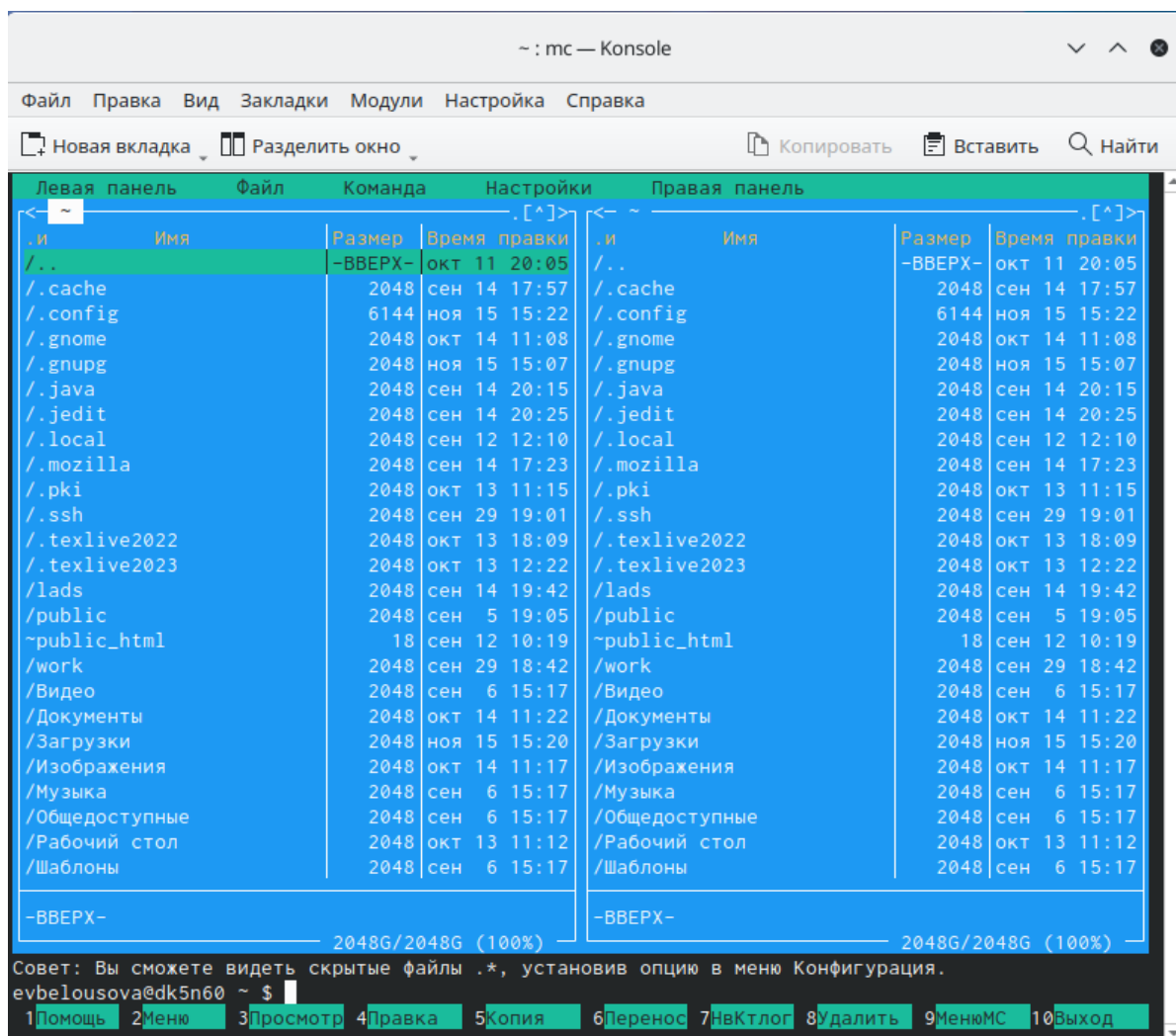


Рис. 1: Открытый mc

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

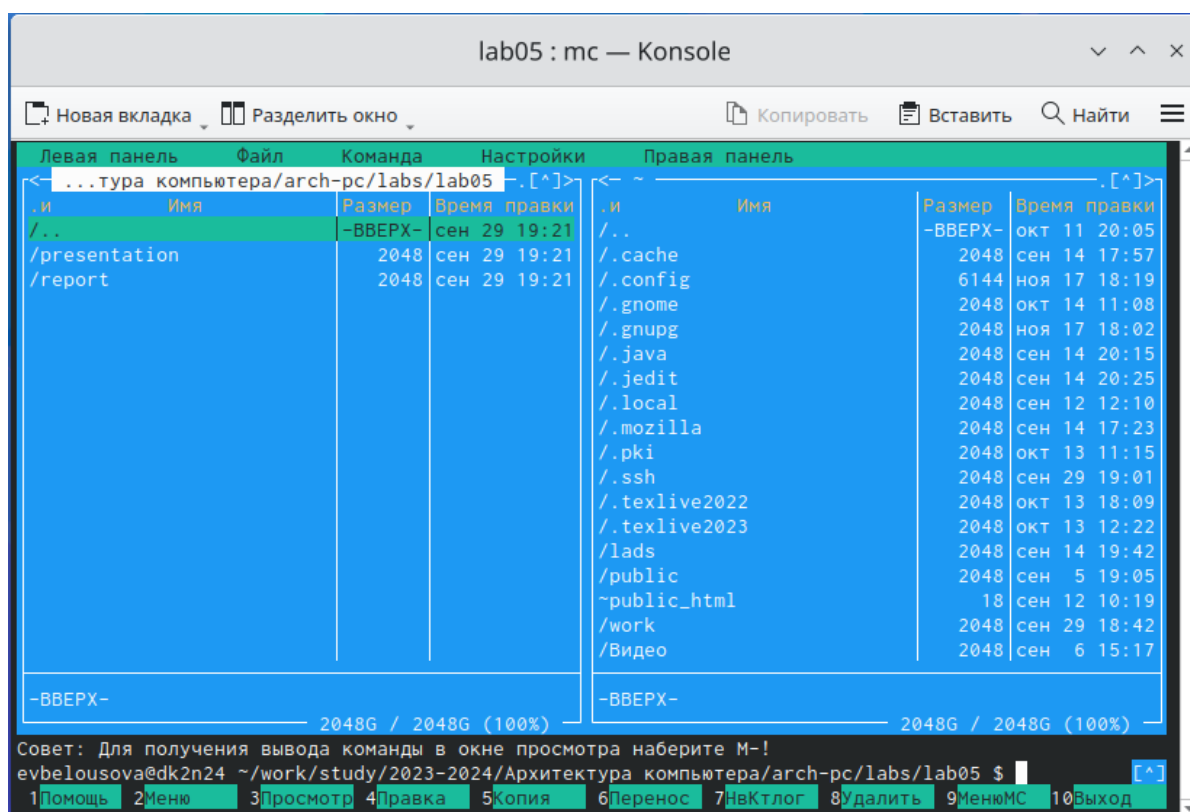


Рис. 2: Перемещение между директориями

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).

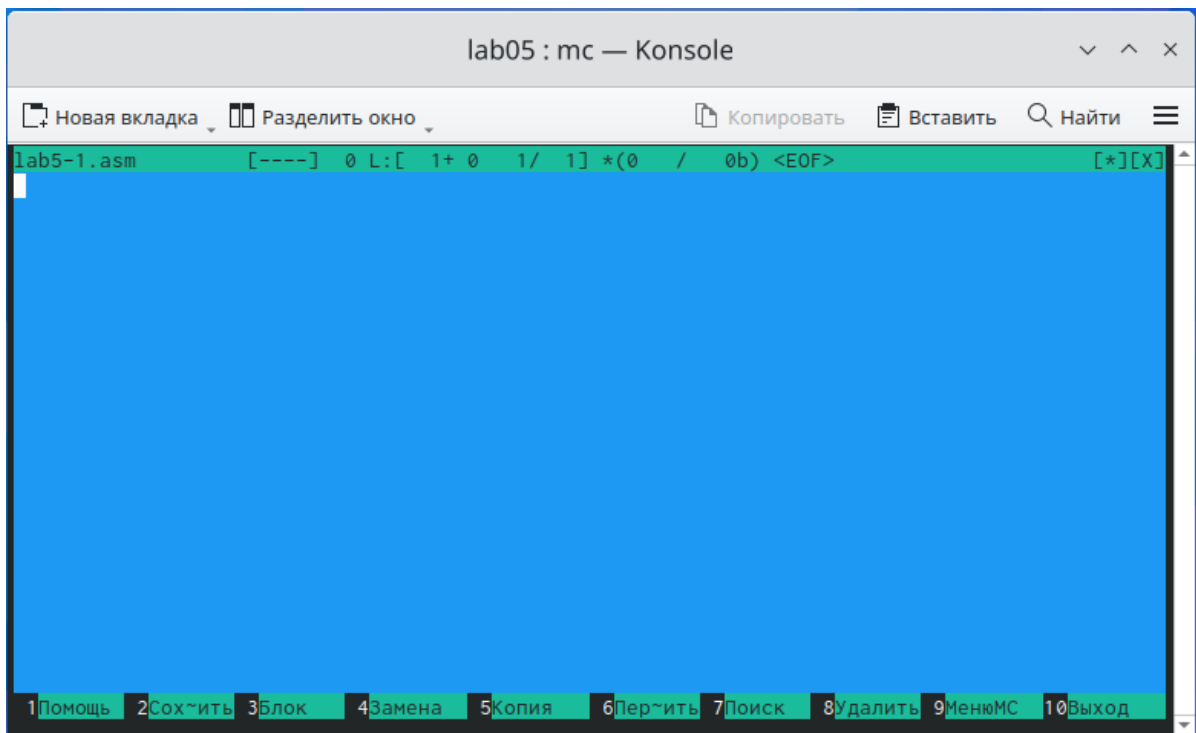


Рис. 3: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 4).

Далее выхожу из файла (Ctrl+X), сохраняя изменения (Y, Enter).

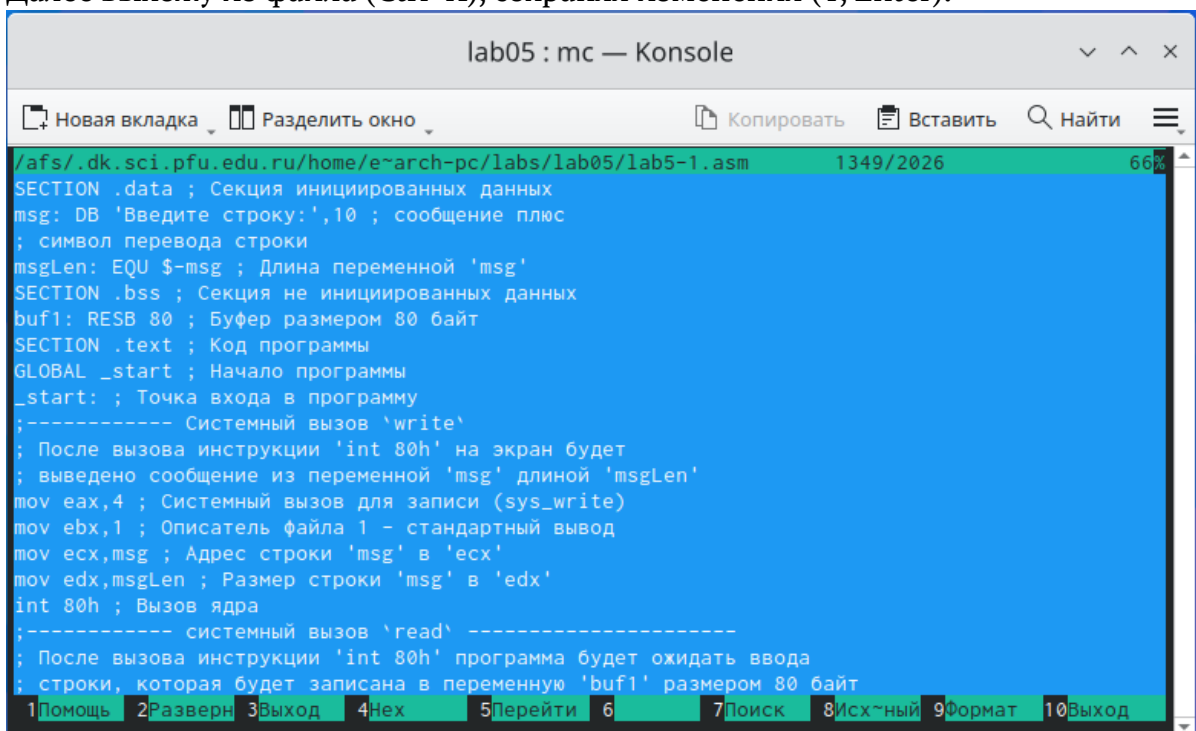


Рис. 4: Редактирование файла

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 lab6-1.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).

```
evbelousova@dk2n24 ~ $ mc
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ touch lab5-1.asm
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-1.asm
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1
Введите строку:
Белюсова Елизавета Валентиновна
```

Рис. 5: Исполнение файла

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

Загрузки : mc — Konsole							
Левая панель				Правая панель			
Имя	Размер	Время	Правки	Имя	Размер	Время	Правки
./	-ВВЕРХ-	ноя 15 15:50		./	-ВВЕРХ-	окт 11 20:05	
/install-tl-20231013	2048	окт 13 01:46		./cache	2048	сен 14 17:57	
in_out.asm	3842	ноя 17 18:32		./config	6144	ноя 17 18:32	
install-tl.zip	25419K	окт 13 11:32		./gnome	2048	окт 14 11:08	

Рис. 6: Скачанный файл

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

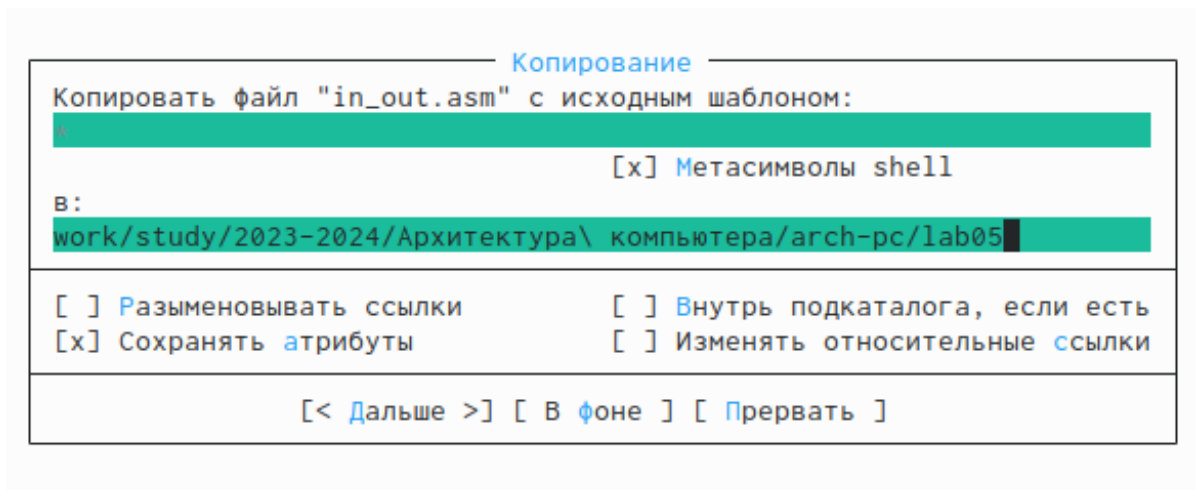


Рис. 7: Копирование файла

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).

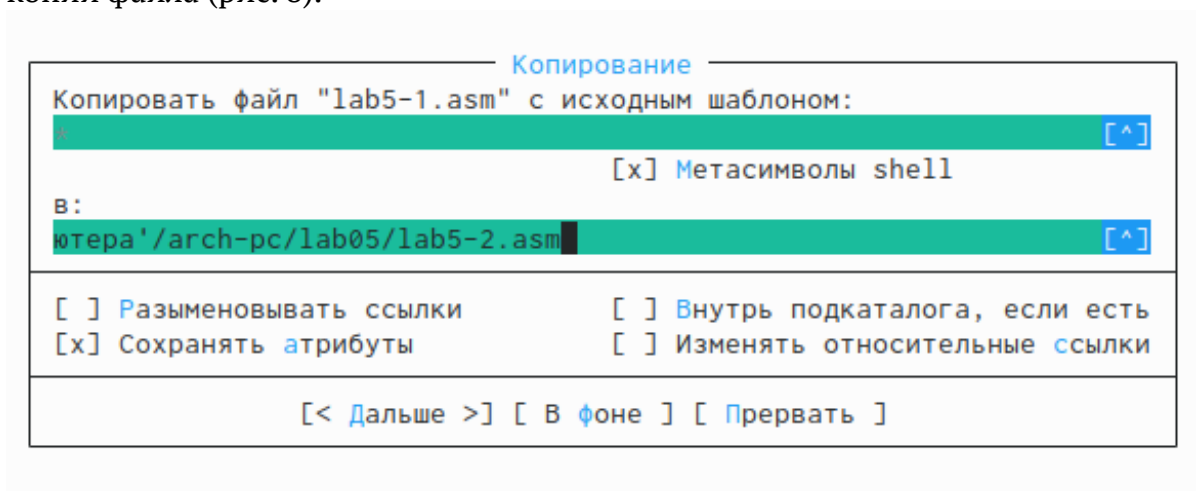


Рис. 8: Копирование файла

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

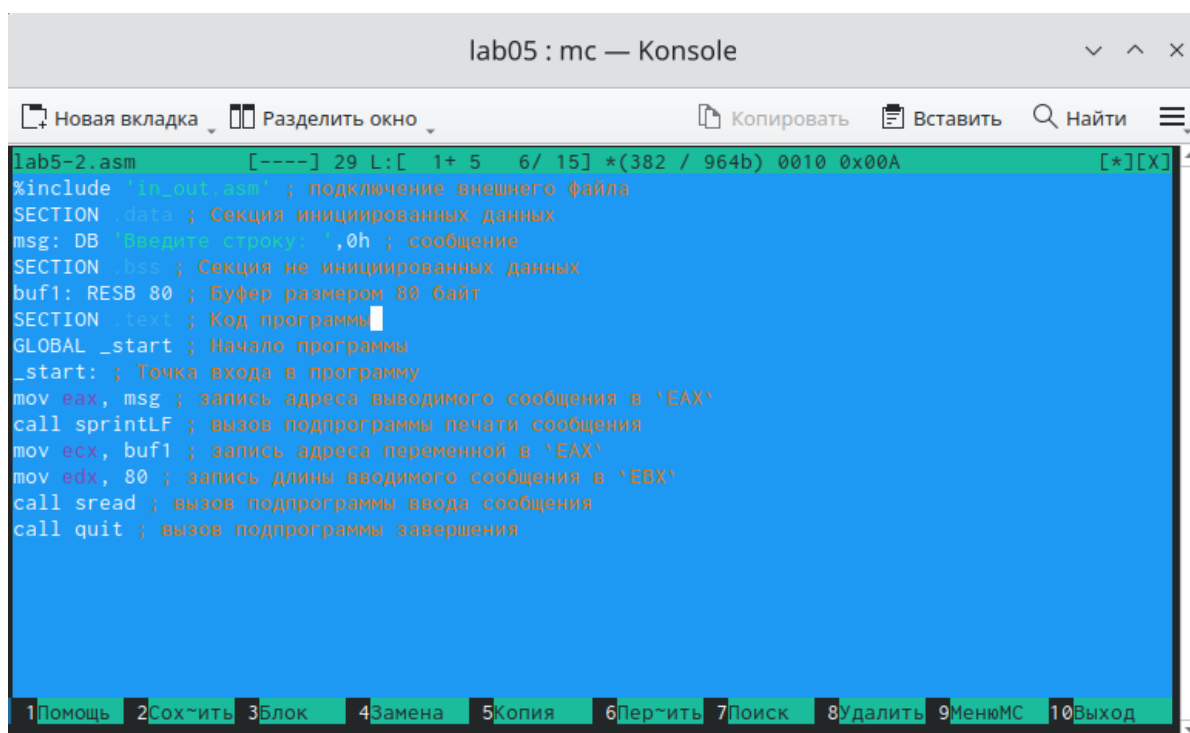


Рис. 9: Редактирование файла

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).

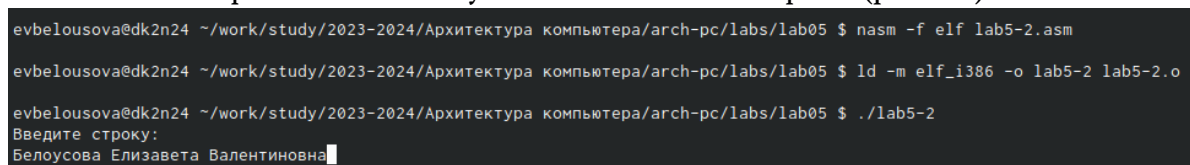
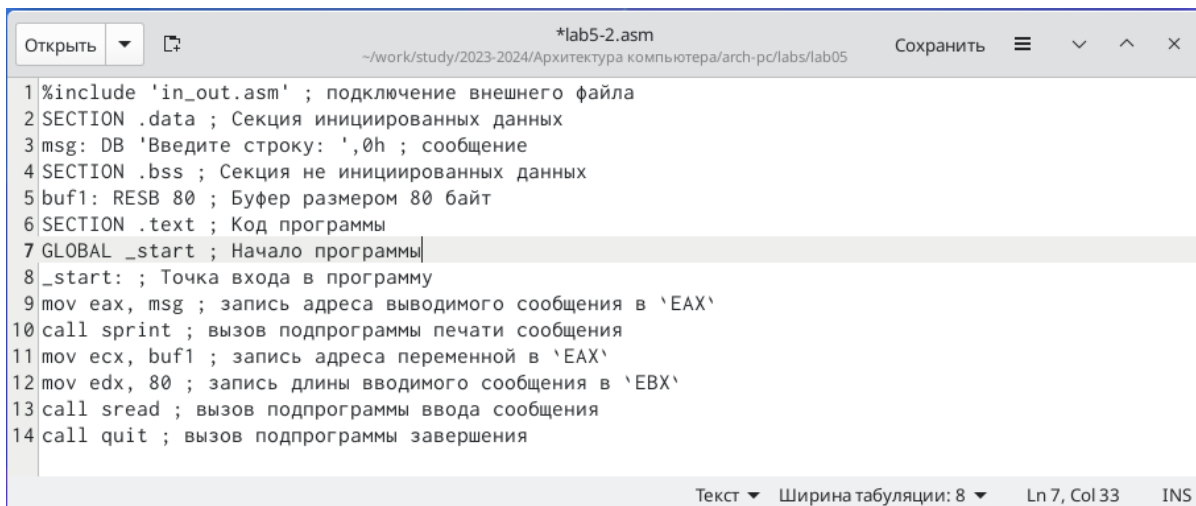


Рис. 10: Исполнение файла

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



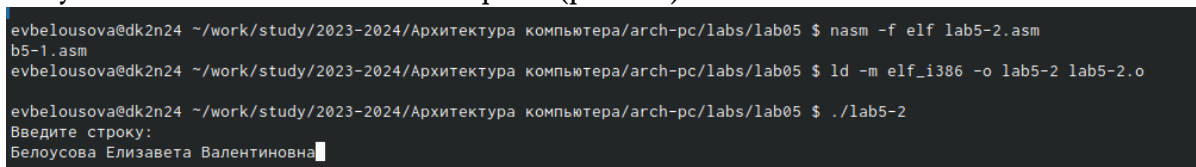
The screenshot shows a text editor window titled '*lab5-2.asm' with the file path '~/.work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05'. The code is as follows:

```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data ; Секция инициализированных данных
3 msg: DB 'Введите строку: ',0h ; сообщение
4 SECTION .bss ; Секция не инициализированных данных
5 buf1: RESB 80 ; Буфер размером 80 байт
6 SECTION .text ; Код программы
7 GLOBAL _start ; Начало программы
8 _start: ; Точка входа в программу
9 mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
10 call sprint ; вызов подпрограммы печати сообщения
11 mov ecx, buf1 ; запись адреса переменной в 'EAX'
12 mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
13 call sread ; вызов подпрограммы ввода сообщения
14 call quit ; вызов подпрограммы завершения
```

The status bar at the bottom indicates 'Текст', 'Ширина табуляции: 8', 'Ln 7, Col 33', and 'INS'.

Рис. 11: Отредактированный файл

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



The terminal window shows the following commands and output:

```
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-2.asm
b5-1.asm
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-2
Введите строку:
Белуосова Елизавета Валентиновна
```

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).

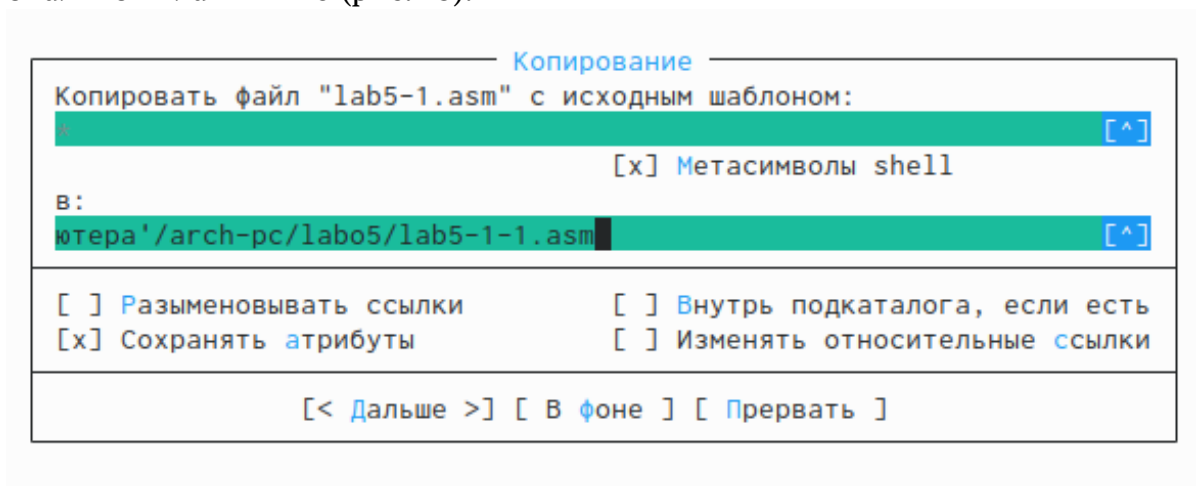


Рис. 13: Копирование файла

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).

```

lab5-1-1.asm  [-M--]  9 L: [ 16+19  35/ 35] *(2132/2152b) 0032 0x020  [*][X]
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 ;Descriptor файла 0 – стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ;
mov ecx, buf1 ;
mov edx, buf1 ;
int 80h ;
mov eax, 1 ; Системный вызов для выхода (sys_exit)
mov ebx, 0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перенести 7Поиск 8Удалить 9МенюМС 10Выход

```

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).

```

evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-1-1.asm
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1-1
Введите строку:
Белуосова Елизавета Валентиновна

```

Рис. 15: Исполнение файла

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

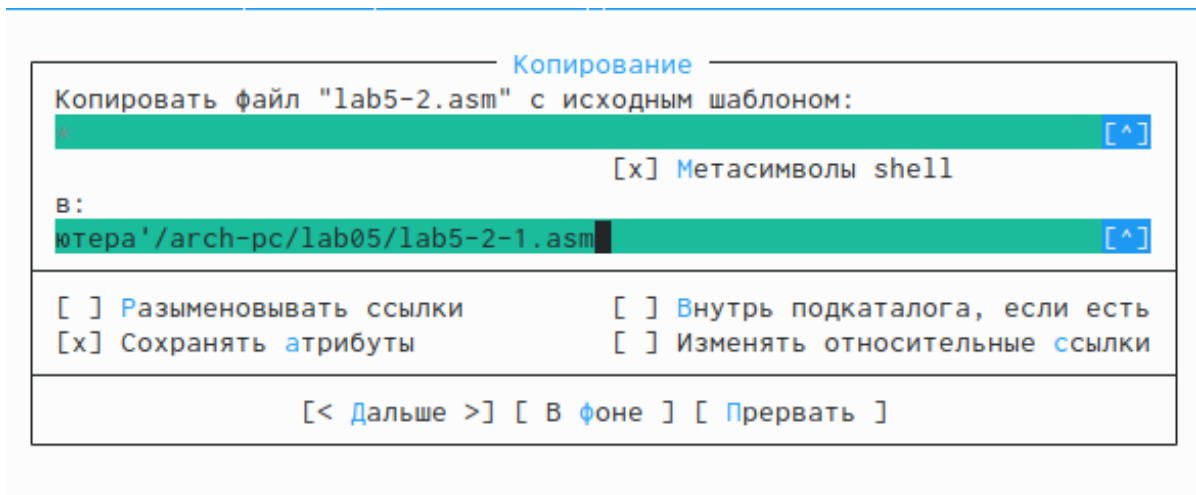


Рис. 16: Копирование файла

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).

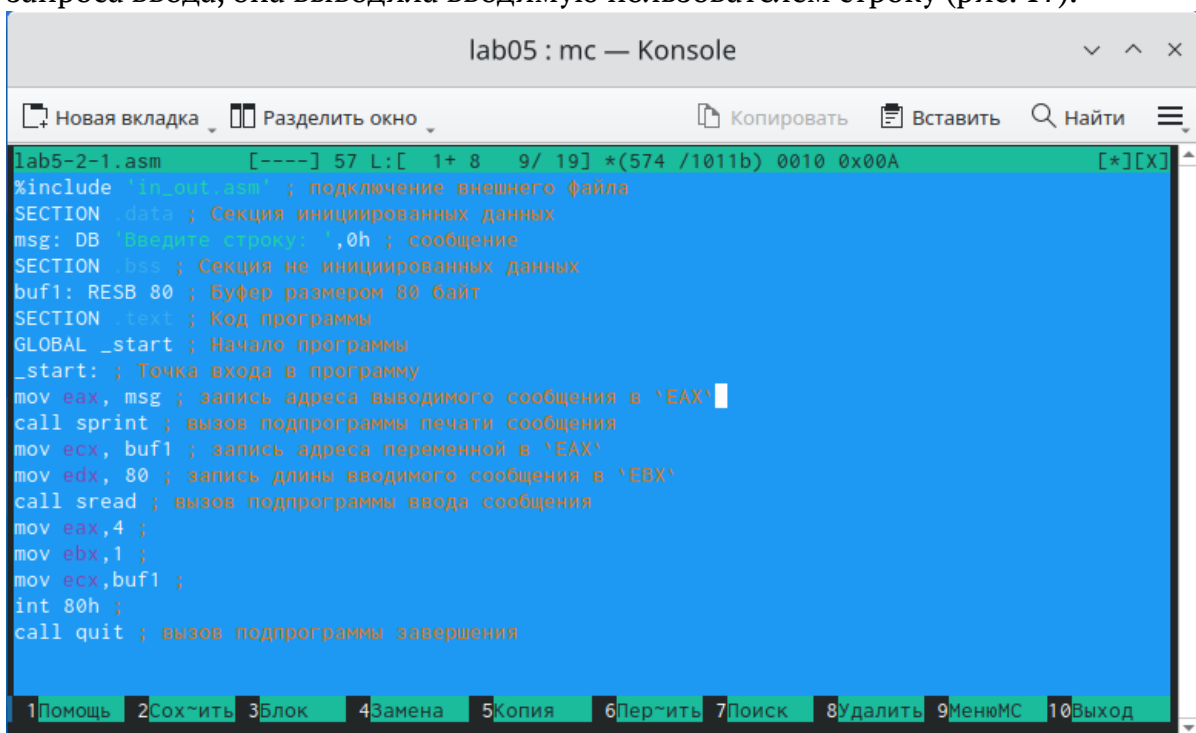


Рис. 17: Редактирование файла

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).

```
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-2-1.asm
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
evbelousova@dk2n24 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-2-1
Введите строку: Белоусова Елизавета Валентиновна
Белоусова Елизавета Валентиновна
```

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