

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Архитектура компьютера

Студент: Юсупова Амина Руслановна

Группа:НКАбд-06-25

МОСКВА

2025 г.

Содержание

1	Цель работы	2
2	Задание.....	2
3	Теоретическое введение	2
4	Выполнение лабораторной работы.....	3
4.1	Основы работы с Midnight Commander.....	3
4.2	Работа в NASM.....	6
4.3	Подключение внешнего файла	8
4.4	Задания для самостоятельной работы	10
5	Выводы	15
	Список литературы	15

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 инициализированных (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 инициализированных 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 размером 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 (учетверенное 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

4.1 Основы работы с Midnight Commander

Введя соответствующую команду (mc) в терминале (рис. 1), я открываю Midnight Commander (рис. 2).

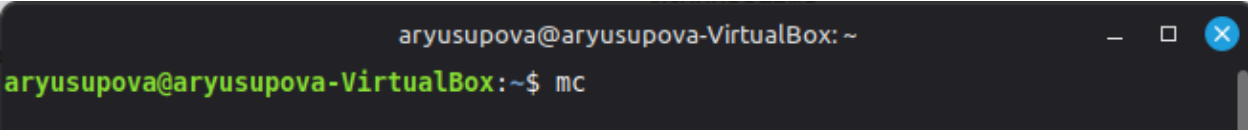


Рисунок 1: Открытие Midnight Commander

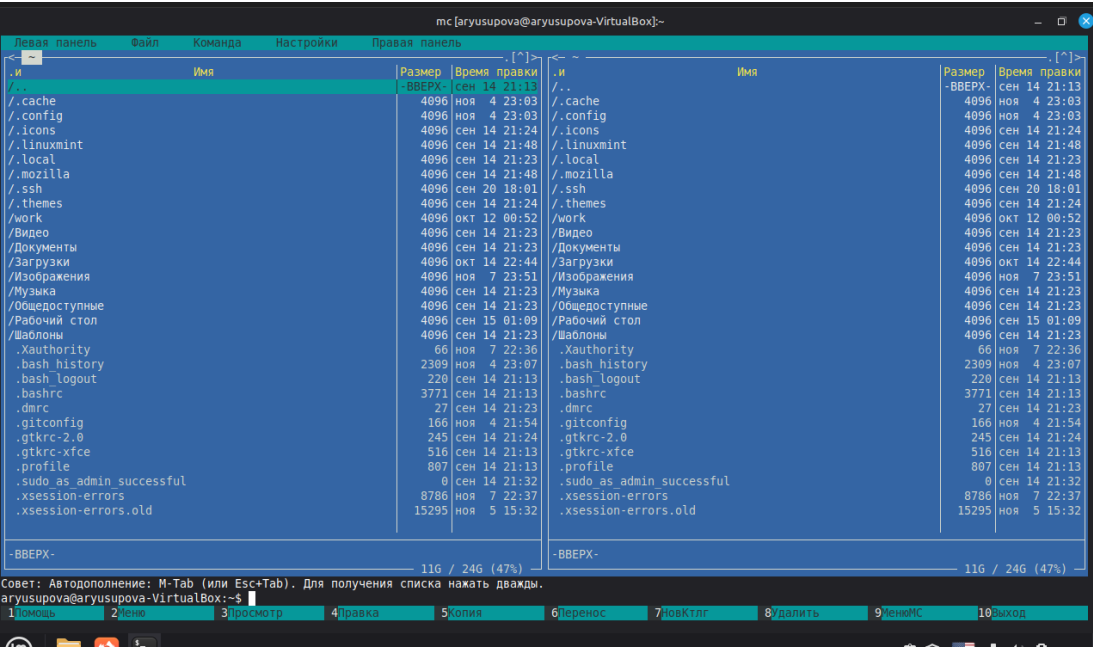


Рисунок 2: Интерфейс Midnight Commander

Перехожу в созданный каталог в предыдущей лабораторной работе (рис. 3).

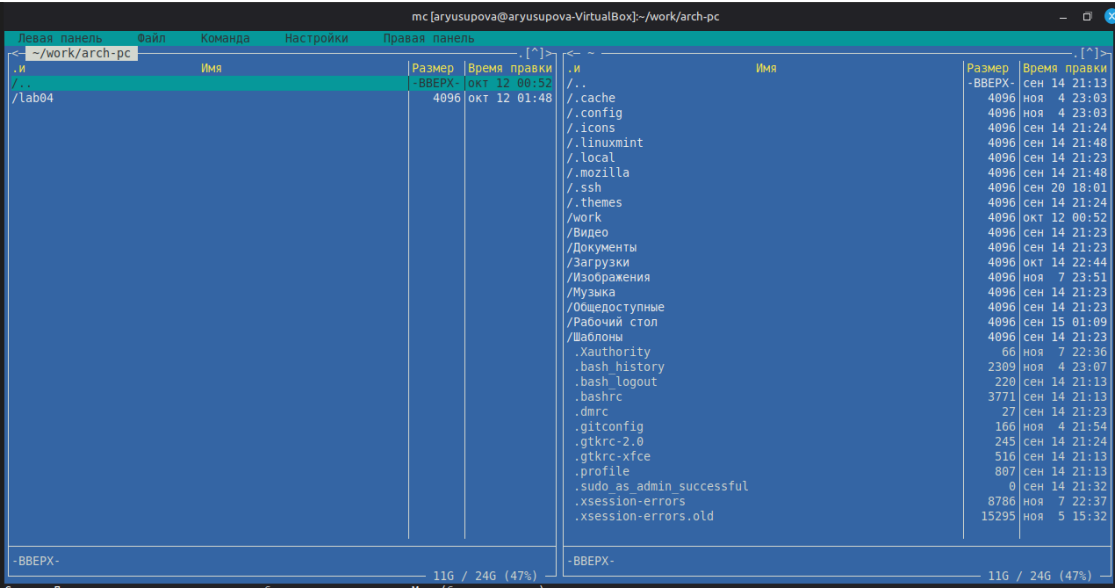


Рисунок 3: Открытие каталога arch-pc

С помощью функциональной клавиши, я создаю подкаталог lab05, в котором буду работать (рис. 4).

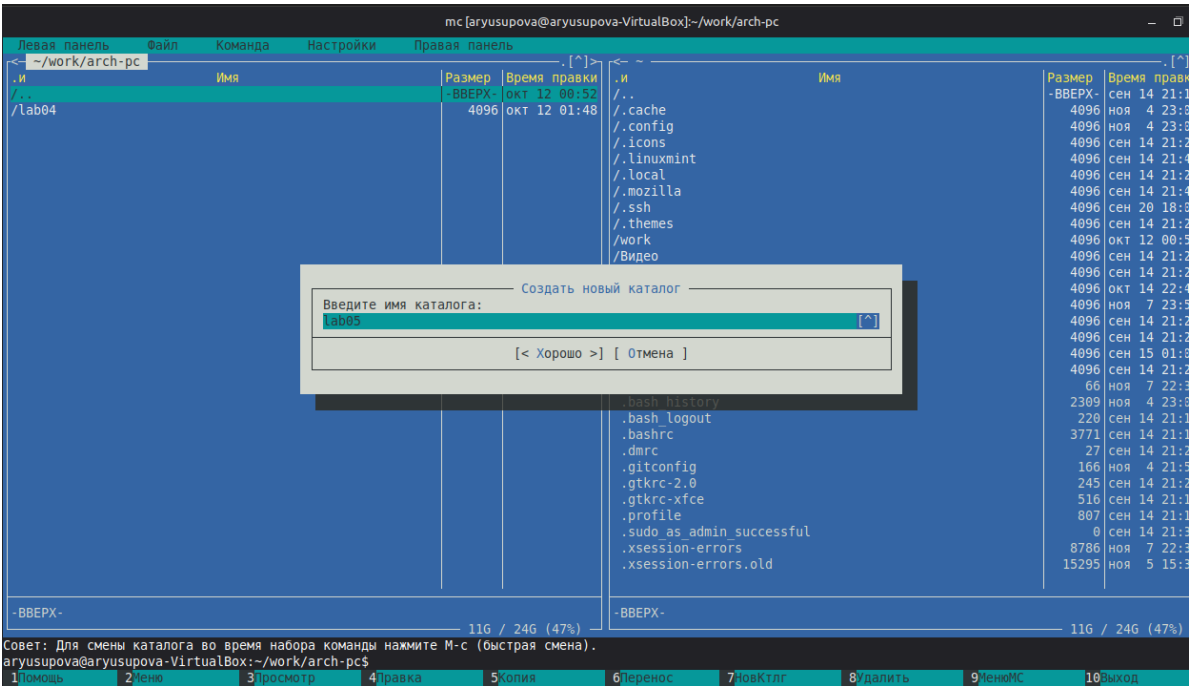


Рисунок 4: Создание рабочего подкаталога

В строке ввода ввожу команду touch и создаю файл (рис. 5). Проверяю создание файла (рис. 6).

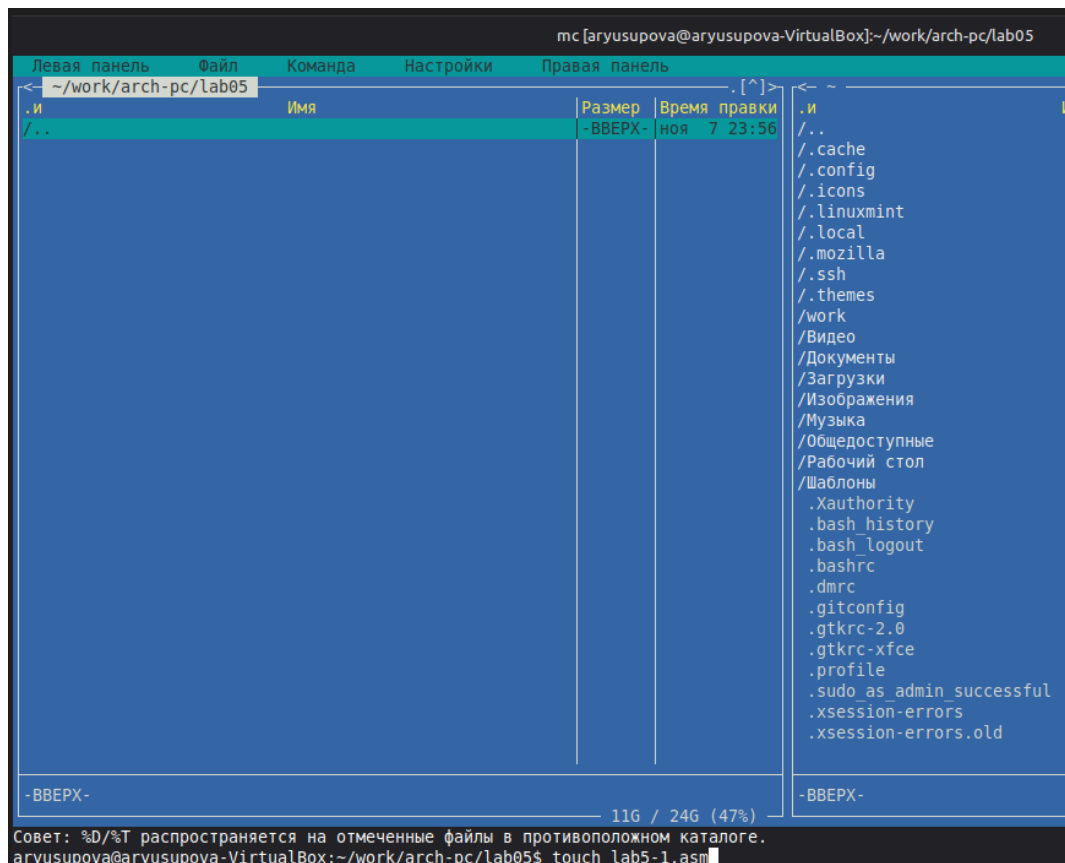


Рисунок 5: Создание файла asm

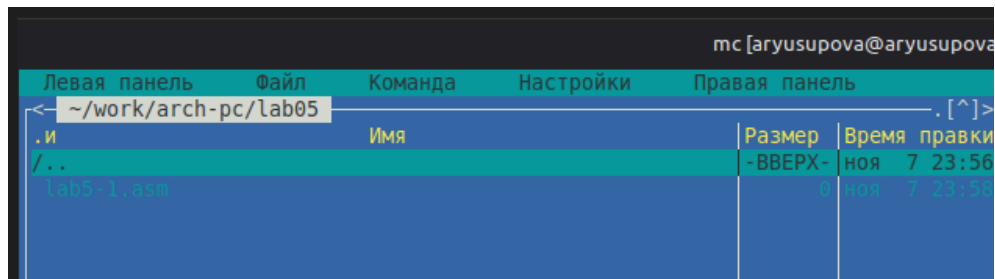


Рисунок 6: Созданный файл

4.2 Работа в NASM

С помощью F4 открываю только что созданный файл и вношу код с листинга (рис. 7). Открываю созданный и отредактированный файл lab5-1.asm и проверяю сохранений внесённых изменений (рис. 8).

```
mc [aryusupova@aryusupova-VirtualBox]:~/work/arch-pc/lab05
GNU nano 7.2 /home/aryusupova/work/arch-pc/lab05/lab5-1.asm *
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, 1
    mov     ebx, 0
    int     80h
```

Рисунок 7: Редактирование файла в Midnight Commander

```
mc [aryusupova@aryusupova-VirtualBox]:~/work/arch-pc/lab05
GNU nano 7.2 /home/aryusupova/work/arch-pc/lab05/lab5-1.asm
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
    [ Прочитано 26 строк ]
```

Рисунок 8: Проверка сохраненных изменений

Транслирую и компоную измененный файл, запускаю (рис. 9). Программа работает корректно: выводит строку 'Введите строку:' и ожидает ввода с клавиатуры.

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Юсупова Амина Руслановна
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$
```

Рисунок 9: Трансляция, компоновка и последующий запуск программы

4.3 Подключение внешнего файла

Скачанный с ТУИС файл сохраняю в общую папку на своем компьютере, на виртуальной машине в интерфейсе Midnight Commander перехожу в директорию общей папки, копирую файл в рабочий подкаталог (рис. 10).

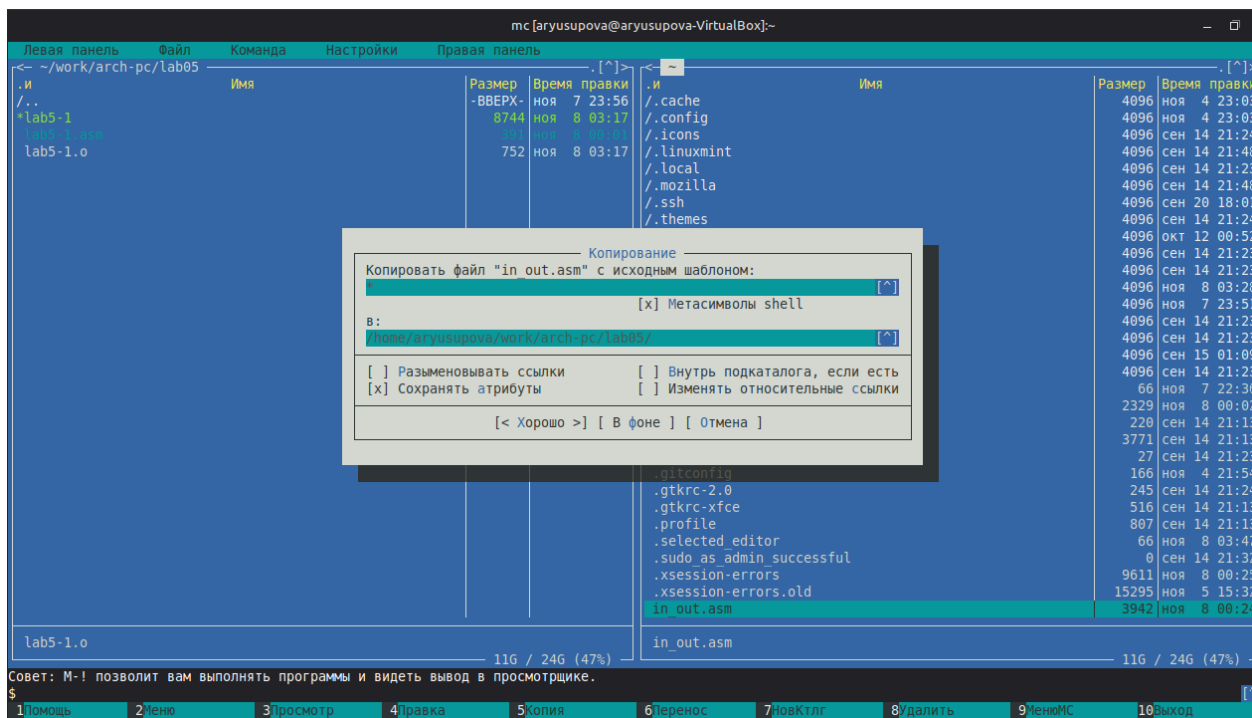


Рисунок 10: Копирование файла в рабочий каталог

Создаю копию файла lab5-1.asm для последующей работы с ним, копию исходного файла именуую lab5-2.asm (рис. 10).

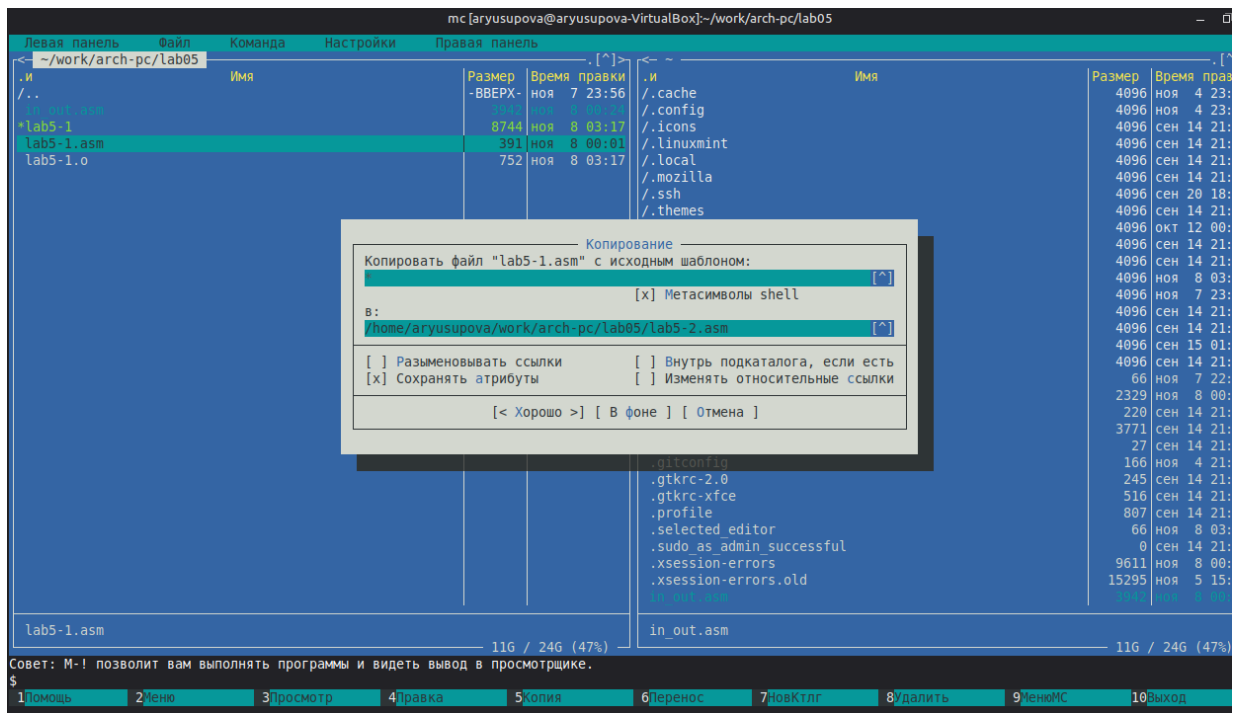


Рисунок 11: Создание копии файла в Midnight Commander

В копии файла подключаю подпрограмм из подключенного файла (рис. 12).



Рисунок 12: Изменение программы

Транслирую, компоную и запускаю программу с подключенным файлом (рис. 13).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Юсупова Амина Руслановна
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$
```

Рисунок 13: Трансляция, компоновка и запуск программы

Редактирую файл lab5-2.asm и заменяю в нем подпрограмму sprintLF на sprint. Разница подпрограмм в том, что вторая вызывает ввод на той же строке (рис. 14).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Юсупова Амина Руслановна
```

Рисунок 14: Редактирование файла lab5-2.asm

4.4 Задания для самостоятельной работы

Создаю копию lab5-1.asm, именую ее как lab5-1copy.asm, редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры (фамилия). (рис. 15).

```
aryusupova@aryusupova-VirtualBox: ~/work/arch-pc/lab05
GNU nano 7.2 /home/aryusupova/work/arch-pc/lab05/lab5-1copy.asm
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, buf1
    int     80h
    mov     eax, 1
    mov     ebx, 0
    int     80h
```

Рисунок 15: Редактирование копии

Транслирую, компоную и запускаю свою программу файл lab5-1copy.asm (рис. 16).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-1copy.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1copy lab5-1copy.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1copy
Введите строку:
Юсупова Амина Руслановна
Юсупова Амина Руслановна
```

Рисунок 16: Запуск изменённой программы

Код из программы (рис.15).

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, buf1
    int     80h

    mov     eax, 1
    mov     ebx, 0
    int     80h
```

Создаю копию lab5-2.asm, именую ее как lab5-2copy.asm (рис. 17), редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры (рис. 18).

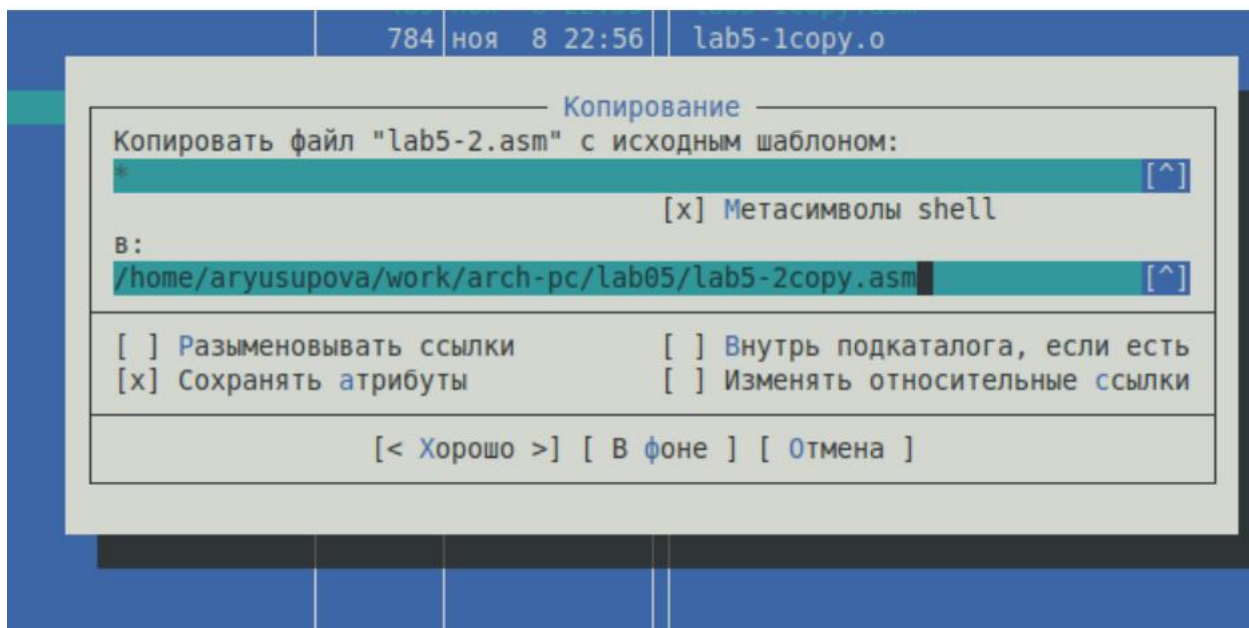


Рисунок 17: Создание копии файла lab5-2copy.asm



Рисунок 18: Редактирование копии lab5-2copy.asm

Транслирую, компоную и запускаю свою программу lab5-2copy.asm (рис. 19).

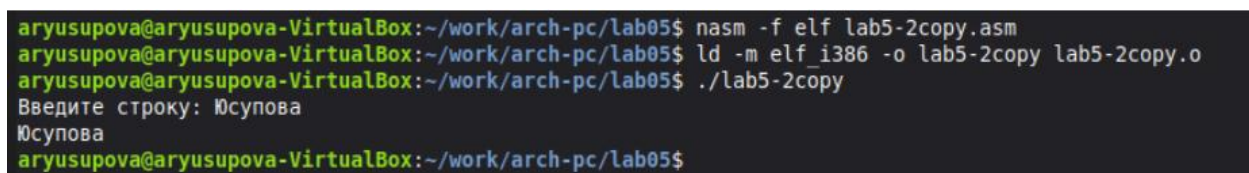


Рисунок 19: Редактирование, компоновка и запуск программы lab5-2copy.asm

Код прикладываю:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg: DB 'Введите строку: ', 0h
```

```
msgLen: EQU $-msg
```

```
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
```

5 Выводы

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

Список литературы

1. <https://esystem.rudn.ru/mod/resource/view.php?id=1030553>
2. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
3. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
4. https://github.com/Amina-colab/study_2025-2026_arh_pc/tree/master/labs