

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Исупова Кристина Павловна

Оглавление

1 Цель работы	3
2 Задание.....	4
3 Теоретическое введение.....	5
4 Выполнение лабораторной работы	6
4.1 Основы работы с Midnight Commander	6
4.2 Работа в NASM	9
4.3 Подключение внешнего файла.....	10
4.4 Задание для самостоятельной работы.....	13
5 Выводы.....	20
Список литературы	21

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 (четверёхбайтное 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 operand 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 в терминале (Рис. 1) **Error! Reference source not found.**) я открываю Midnight Commander (Рис. 2)

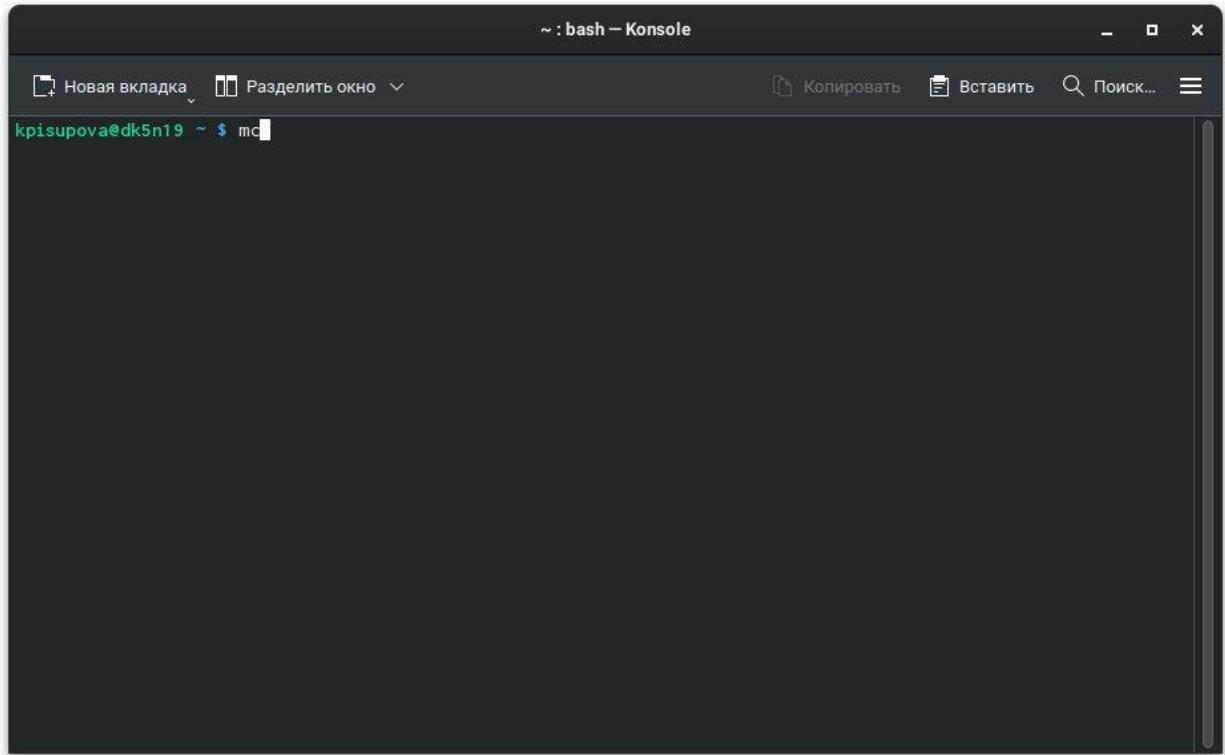


Рис. 1 Открытие Midnight Commander.

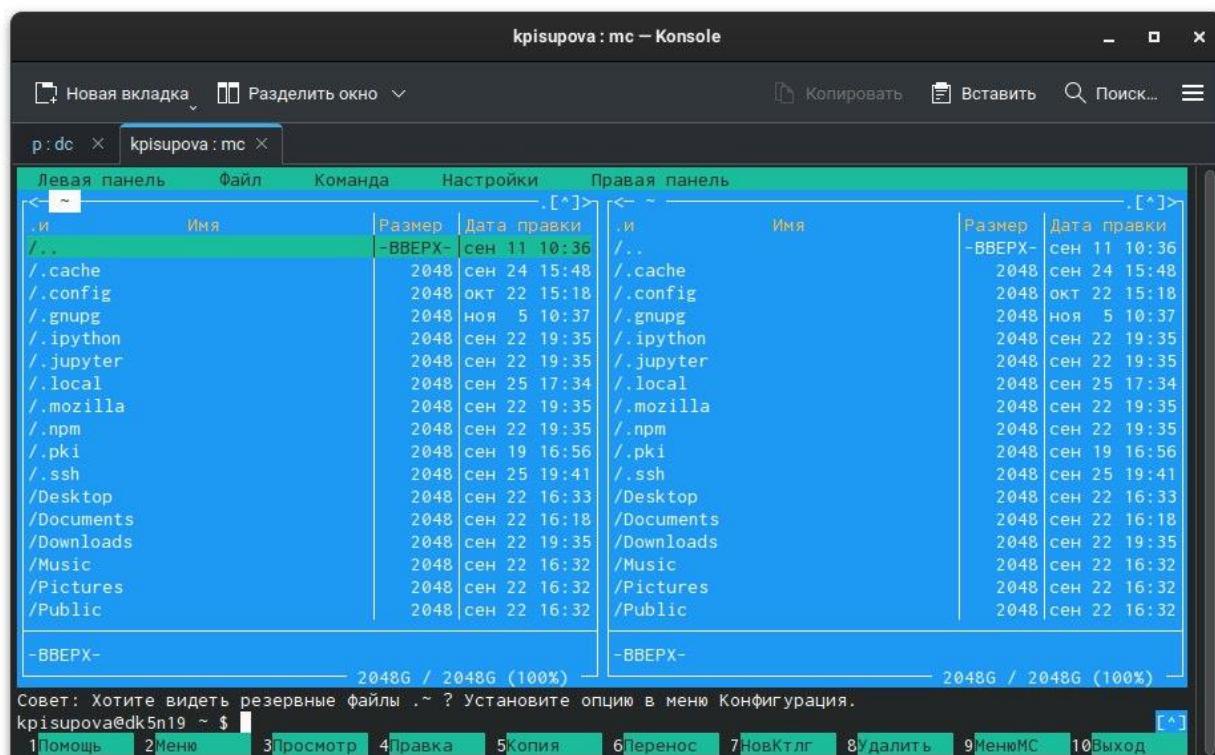


Рис. 2 Интерфейс Midnight Commander

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

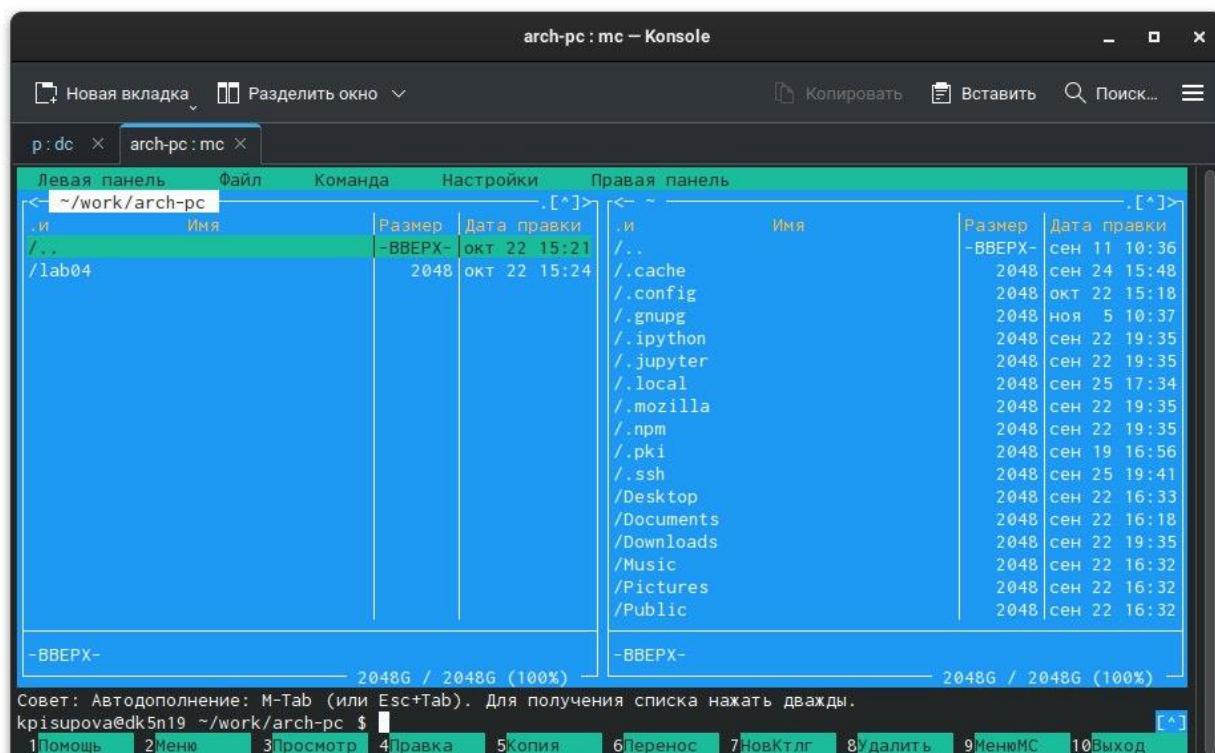


Рис. 3 Открытый каталог arch-pc.

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

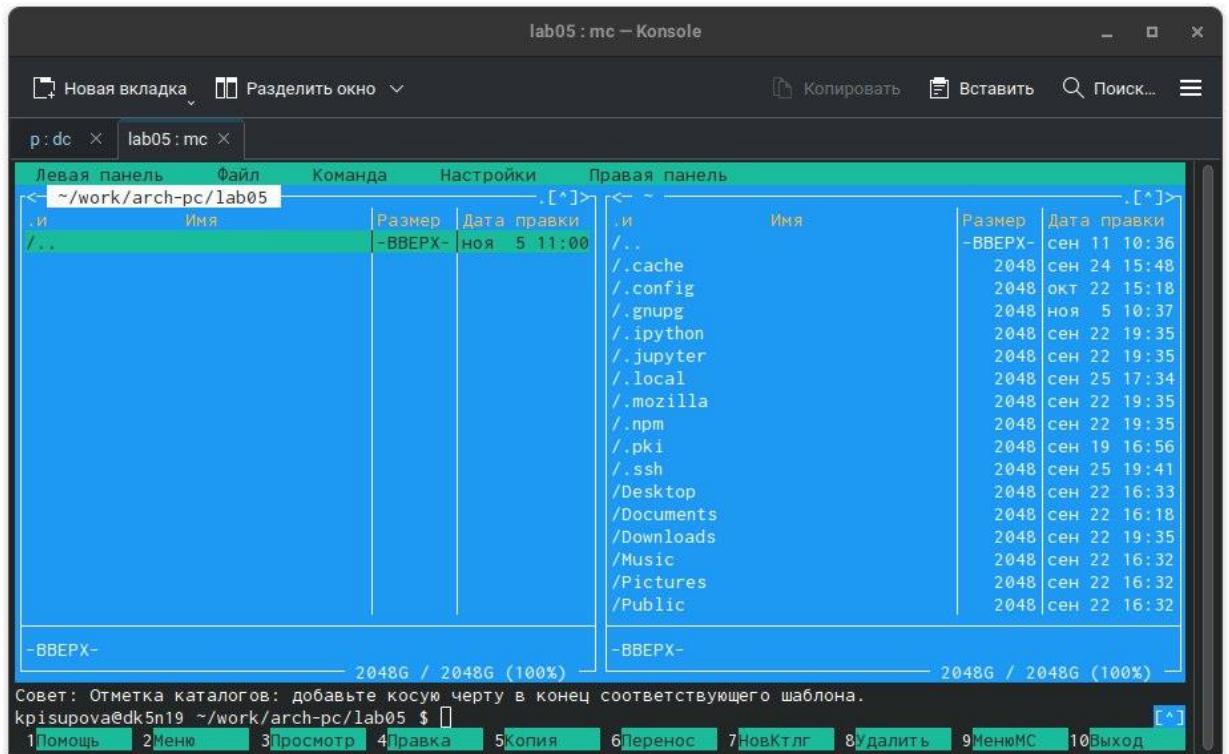


Рис. 4 Создание рабочего подкаталога

В строке ввода ввожу команду touch и создаю файл (Рис. 5).

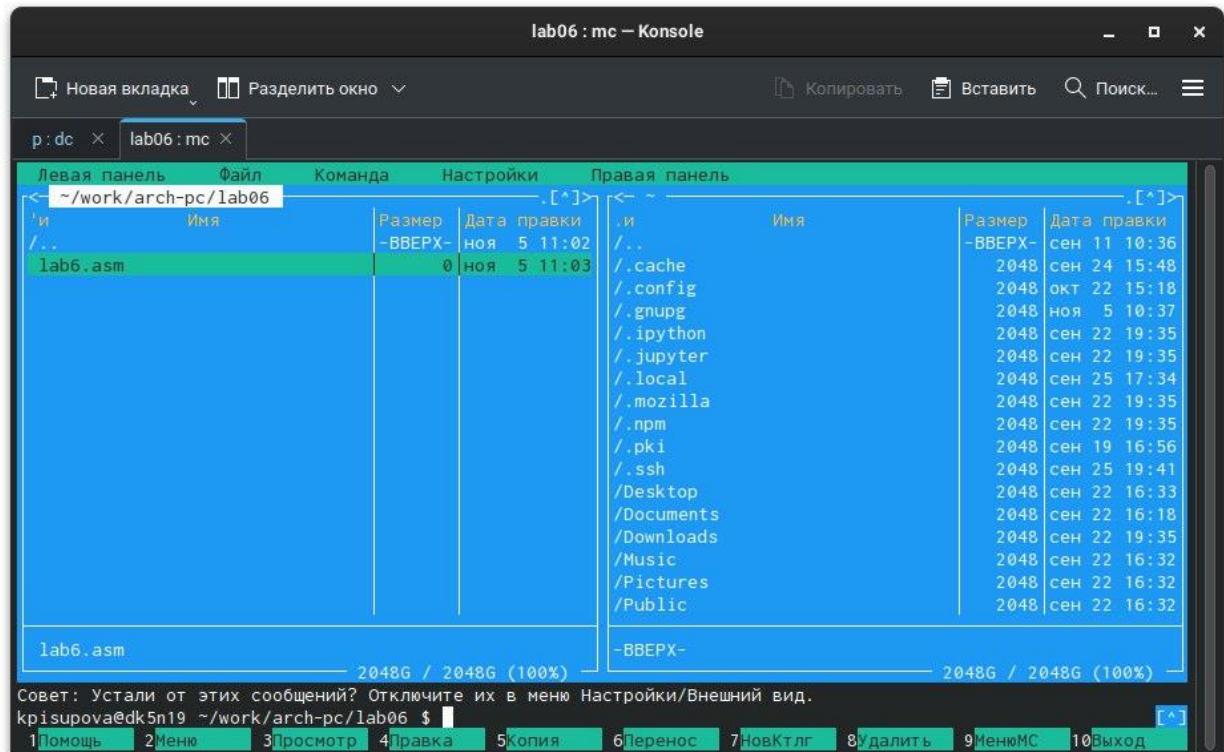


Рис. 5 Создание файла в Midnight Commander

4.2 Работа в NASM

С помощью F4 открываю только что созданный файл и вношу код с листинга (Рис. 6).

```

SECTION .text
msg: DB 'Hello world!', 10
msglen equ $ - msg
SECTION .data
buf1: DB 80h, 0
SECTION .bss
GLOBAL _start
_start:
    mov ax, 4
    mov bx, 1
    mov cx, msg
    mov dx, msgLen
    int 80h
    mov ah, 2
    mov dx, 0
    mov cx, buf1
    mov dx, buf1
    int 80h
    mov ah, 1
    mov dx, 0
    int 80h

```

Рис. 6 Редактирование файла в Midnight Commander

Проверяю сохраненные изменения с помощью клавиши F3 (Рис. 7).

```

lab05 : mc ~ bash ~ lab05 : mc ~ lab05 : mc ~
/afs/ck.sci.psu.edu.ru/home/k/p/kpisupova/work/arch-pc/lab05/lab05-1.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

```

Рис. 7 Проверка сохранения сделанных изменений

Транслирую и компоную измененный файл, запускаю (Рис. 8).

```

kpiupova@dk6n18 ~work/arch-pc/lab05$ nasm -f elf lab05-1.asm
lab05-1.asm:3: error: parser: instruction expected
lab05-1.asm:4: error: label or instruction expected at start of line
lab05-1.asm:8: error: label or instruction expected at start of line
lab05-1.asm:11: warning: label alone on a line without a colon might be in error [-w+label-orphan]
lab05-1.asm:19: error: parser: instruction expected
lab05-1.asm:20: error: parser: instruction expected
lab05-1.asm:21: error: parser: instruction expected
lab05-1.asm:22: error: parser: instruction expected
lab05-1.asm:23: error: parser: instruction expected
lab05-1.asm:24: warning: label alone on a line without a colon might be in error [-w+label-orphan]
lab05-1.asm:26: error: parser: instruction expected
lab05-1.asm:27: error: label or instruction expected at start of line
lab05-1.asm:30: error: parser: instruction expected
lab05-1.asm:31: error: parser: instruction expected
lab05-1.asm:32: error: parser: instruction expected
lab05-1.asm:33: error: parser: instruction expected
lab05-1.asm:34: warning: label alone on a line without a colon might be in error [-w+label-orphan]
lab05-1.asm:36: error: label or instruction expected at start of line
lab05-1.asm:38: error: parser: instruction expected
lab05-1.asm:39: error: parser: instruction expected
lab05-1.asm:40: warning: label alone on a line without a colon might be in error [-w+label-orphan]
kpiupova@dk6n18 ~work/arch-pc/lab05$ ld -m elf_i386 -o lab05-1 lab05-1.o
kpiupova@dk6n18 ~work/arch-pc/lab05$ ./lab05-1
Введите строку:
Исупова Кристина Павловна
kpiupova@dk6n18 ~work/arch-pc/lab05$ 

```

Рис. 8 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. 9).

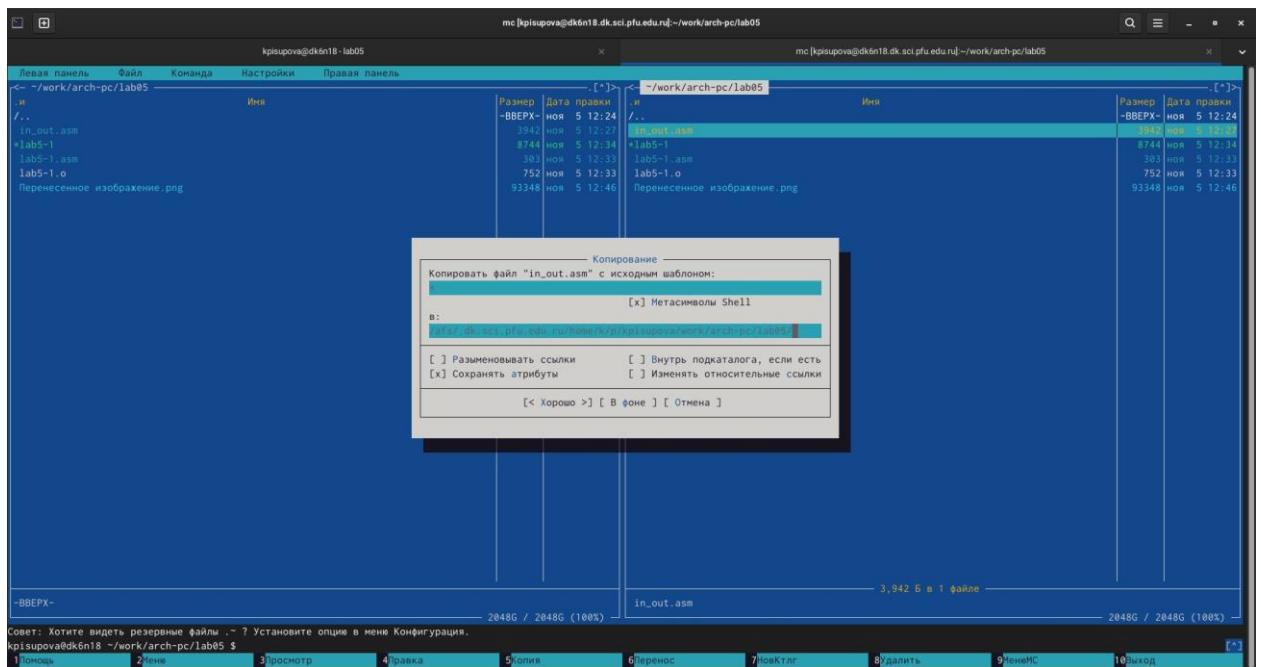


Рис. 9 Копирование файла в рабочий каталог

Создаю копию файла для последующей работы с ним (Рис. 10).

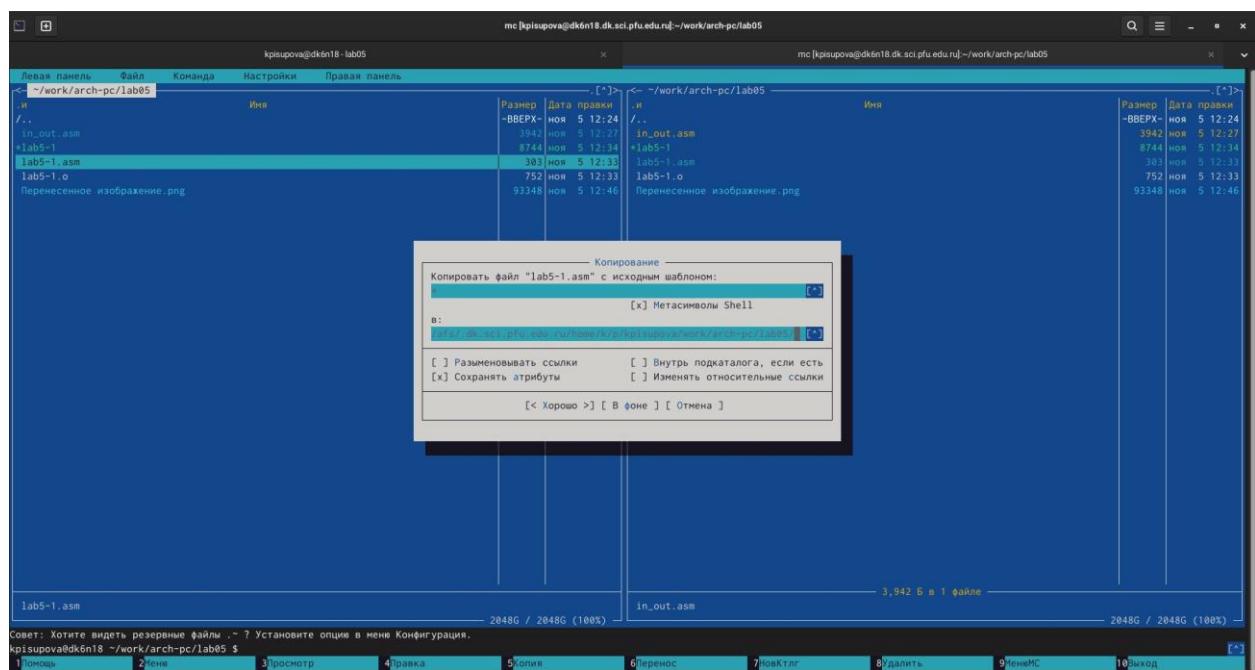


Рис. 10 Создание копии файла в Midnight Commander

В копии файла подключаю подпрограмм из подключенного файла (Рис. 11).

The screenshot shows a terminal window titled 'krisupova@dk6n18 - lab05'. It contains assembly code for a program named 'lab05'. The code includes sections for .data, .bss, and .text, defining variables and performing operations like printing strings and reading from memory. The bottom of the window shows the nano editor's status bar with file names and a 'Изменен' (Changed) indicator.

```
GNU nano 8.6
%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   sprintLF
    mov    ecx, buf1
    mov    edi,80
    call   sread
    all quit
```

Рис. 11 Изменение программы

Транслирую, компоную и запускаю программу с подключенным файлом (Рис. 12).

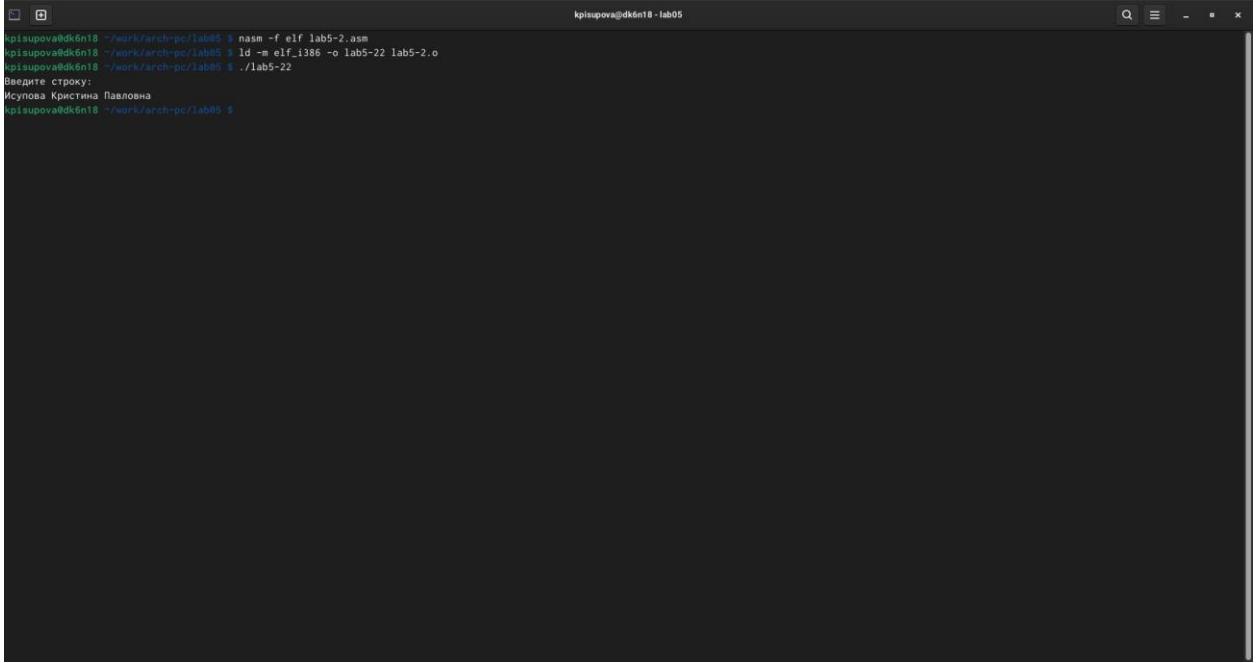
The screenshot shows a terminal window titled 'krisupova@dk6n18 - lab05'. It displays the command 'mc [krisupova@dk6n18 dk.sci.pfu.edu.ru] ~/work/arch-pc/lab05' followed by the output of the assembly program. The output shows the program prompting for input and then displaying the string 'Введите строку:' followed by the user input 'Испупова Кристина Павловна'. The terminal window has multiple tabs open at the top.

```
krisupova@dk6n18 - lab05
mc [krisupova@dk6n18 dk.sci.pfu.edu.ru] ~/work/arch-pc/lab05
krisupova@dk6n18 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Испупова Кристина Павловна
krisupova@dk6n18 ~/work/arch-pc/lab05 $
```

Рис. 12 Запуск измененной программы

Редактирую файл и заменяю в нем подпрограмму sprintLF на sprint. Разница подпрограмм в том, что вторая вызывает ввод на той же строке (Рис. 13).

4.4 Задание для самостоятельной работы



```
kpisupova@dk6n18 ~% cd /work/arch-pe/lab05
kpisupova@dk6n18 ~/work/arch-pe/lab05 % nasm -f elf lab5-2.asm
kpisupova@dk6n18 ~/work/arch-pe/lab05 % ld -m elf_i386 -o lab5-22 lab5-2.o
kpisupova@dk6n18 ~/work/arch-pe/lab05 % ./lab5-22
Введите строку:
Исупова Кристина Павловна
kpisupova@dk6n18 ~/work/arch-pe/lab05 %
```

Рис. 13 Запуск изменённой программы с другой подпрограммой

Создаю копию lab5-1.asm, редактирую так, чтобы в конце выводилась введённая мною строка с клавиатуры (Рис. 14).

```
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
```

Рис. 14 Редактирование копии

Транслирую, компоную и запускаю свою программу (Рис. 15).

```
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ touch lab5-3.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-3
bash: /lab5-3: Нет такого файла или каталога
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Исупова Кристина Павловна
Исупова Кристина Павловна
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ touch lab5-4.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
lab5-4.asm:1: error: parser: instruction expected
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
lab5-4.asm:1: error: parser: instruction expected
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Исупова Кристина Павловна
Исупова Кристина Павловна
kpisupova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 15 Запуск своей программы.

Код прикладываю

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, редактирую так, чтобы в конце выводилась введеная мною строка с клавиатуры (Рис. 16).

```
%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
```

Рис. 16 Редактирование копии

Транслирую, компоную и запускаю свою программу (Рис. 17).

The screenshot shows a terminal window titled "kpisupova@dk3n55 - lab05". The terminal displays the following command-line session:

```
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ touch lab5-3.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-3
bash: ./lab5-3: Нет такого файла или каталога
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Исупова Кристина Павловна
Исупова Кристина Павловна
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ touch lab5-4.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
lab5-4.asm:1: error: parser: instruction expected
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
lab5-4.asm:1: error: parser: instruction expected
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
kpisupova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Исупова Кристина Павловна
Исупова Кристина Павловна
kpisupova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 17 Запуск своей программы

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 освоила

Список литературы

1. [Пример выполнения лабораторной работы](#)
2. [Курс на ТУИС](#)
3. [Лабораторная работа №5](#)
4. [Программирование на языке ассемблера NASM Столяров А. В.](#)