

**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: Томилова Валентина

Группа: НКАбд-06-25

МОСКВА

2025г.

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Основы работы с Midnight Commander	9
4.2	Работа в NASM	12
4.3	Подключение внешнего файла	14
4.4	Задание для самостоятельной работы.....	17
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Открытие Midnight Commander.....	9
4.2	Интерфейс Midnight Commander.....	10
4.3	Открытый каталог arch-pc.....	10
4.4	Создание рабочего подкаталога.....	11
4.5	Создание файла в Midnight Commander.....	11
4.6	Редактирование файла в Midnight Commander.....	12
4.7	Проверка сохранения сделанных изменений.....	13
4.8	Трансляция, компоновка и последующий запуск программы.....	13
4.9	Копирование файла в рабочий каталог.....	14
4.10	Создание копии файла в Midnight Commander.....	15
4.11	Изменение программы.....	15
4.12	Запуск измененной программы.....	16
4.13	Запуск измененной программы с другой подпрограммой.....	16
4.14	Редактирование копии.....	17
4.15	Запуск своей программы.....	17
4.16	Редактирование копии.....	18
4.17	Запуск своей программы.....	18

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 тс
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 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 (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 `intn` предназначена для вызова прерывания с указанным номером.

`intn`

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. 4.1), я открываю Midnight Commander (рис. 4.2).

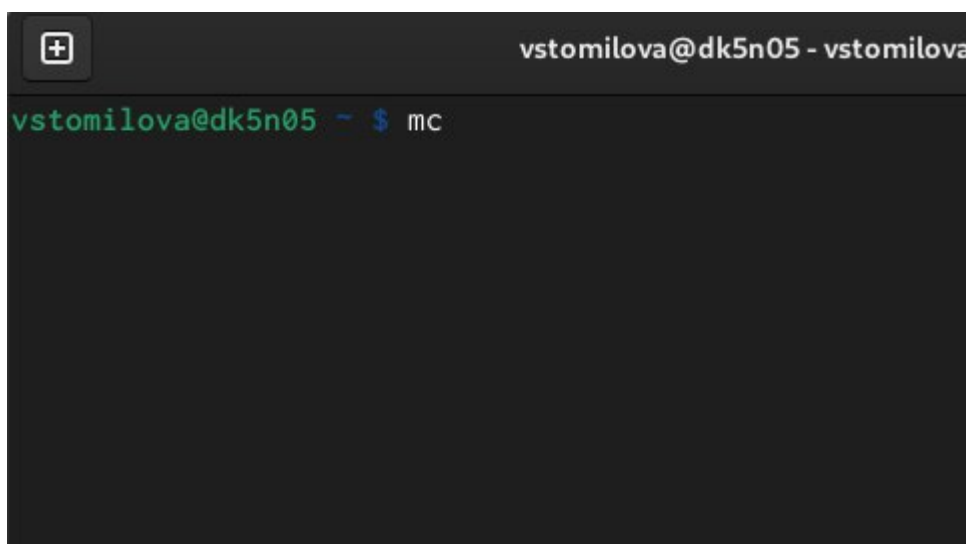


Рис. 4.1: Открытие Midnight Commander

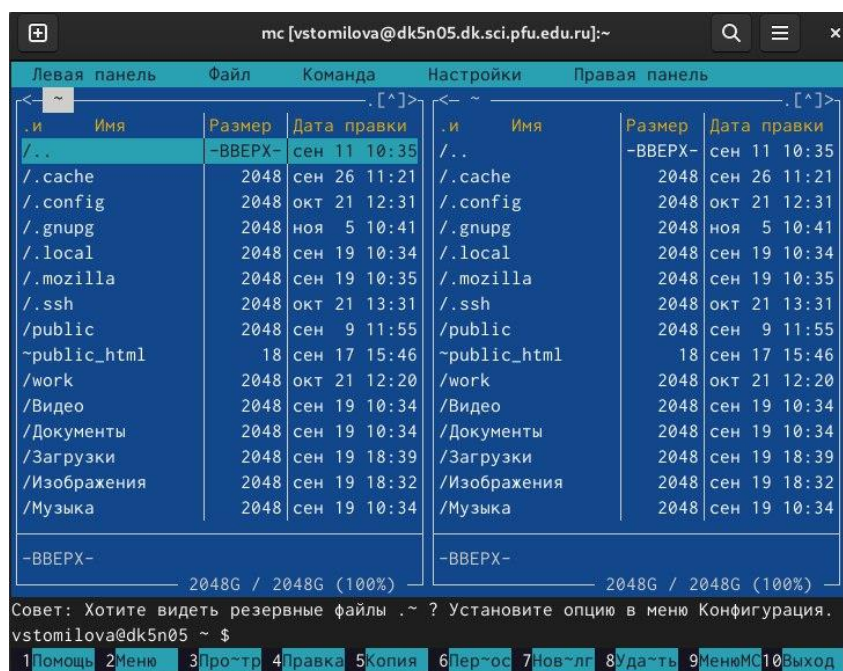


Рис. 4.2: Интерфейс Midnight Commander

Перехожу в созданный каталог в предыдущей лабораторной работе (рис. 4.3).

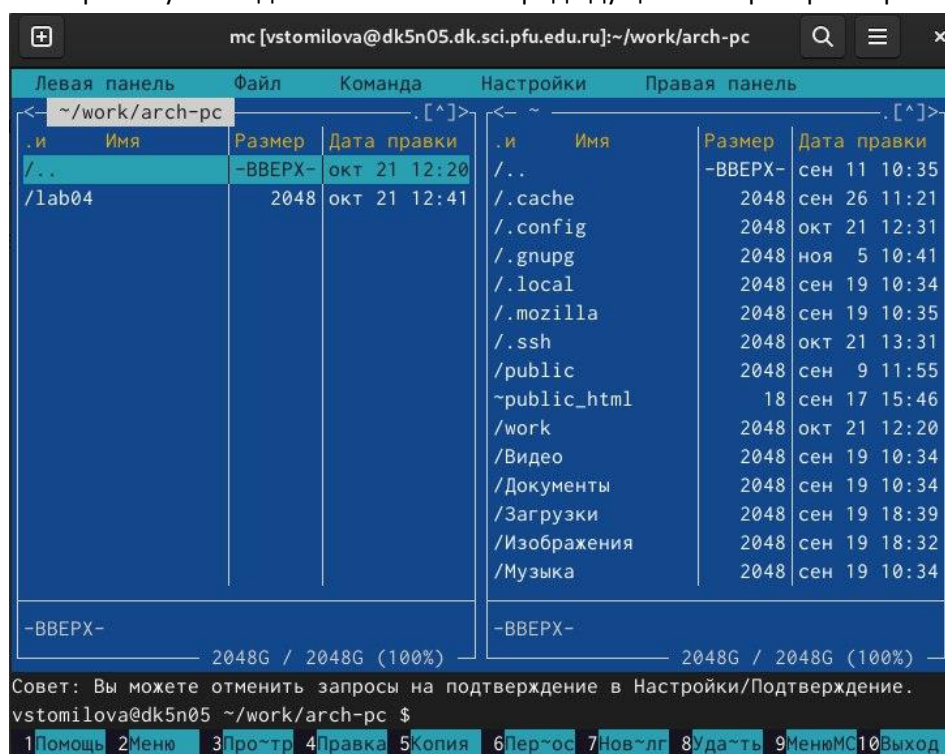


Рис. 4.3: Открытый каталог arch-pc

С помощью функциональной клавиши, я создаю подкаталог lab05, в котором буду

работать (рис. 4.4).

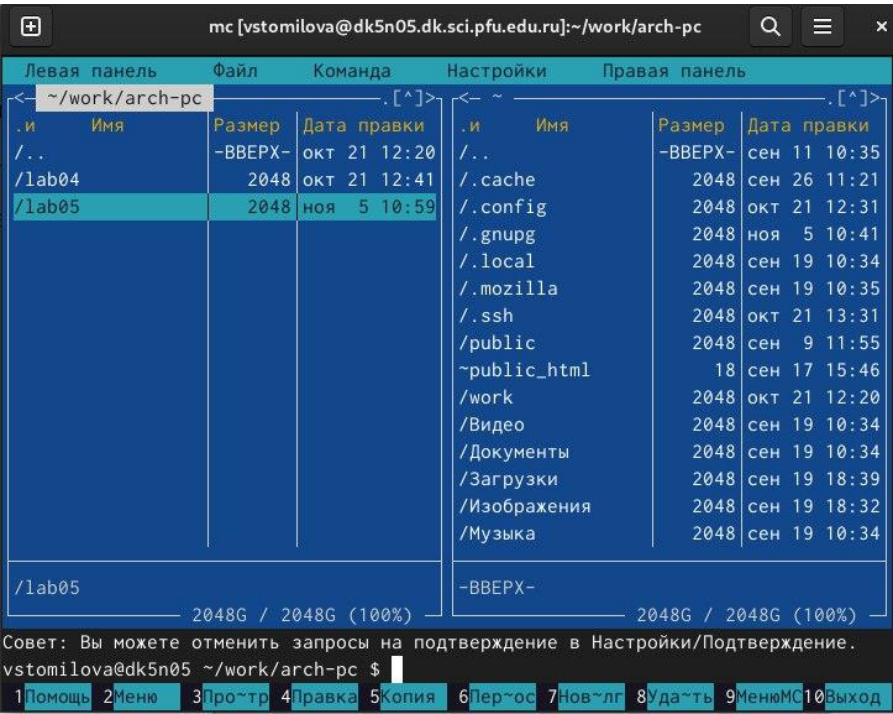


Рис. 4.4: Создание рабочего подкаталога

В строке ввода вводжу команду touch и создаю файл (рис. 4.5).

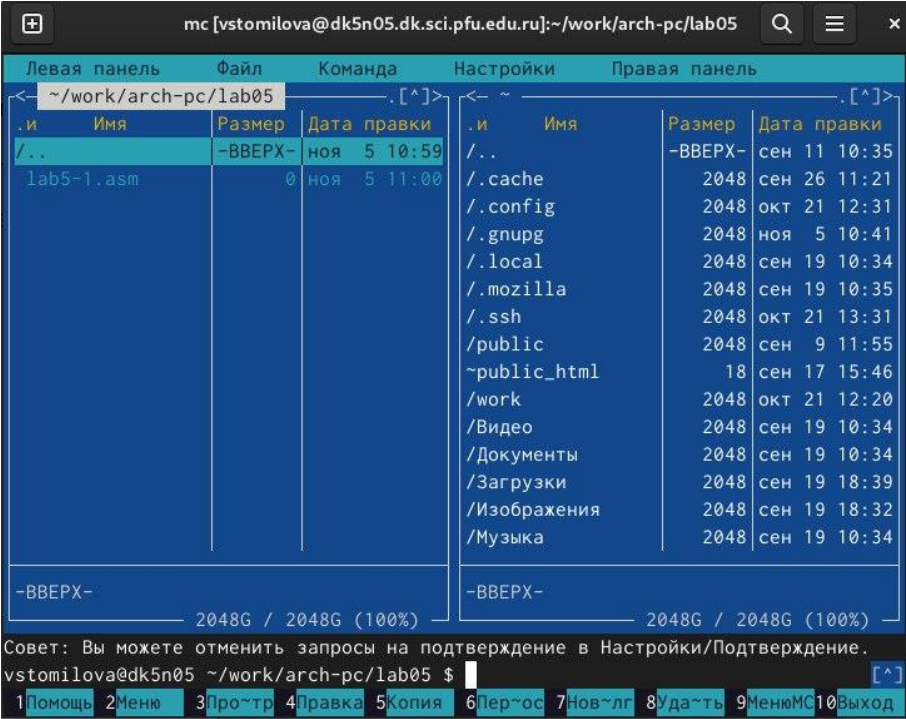
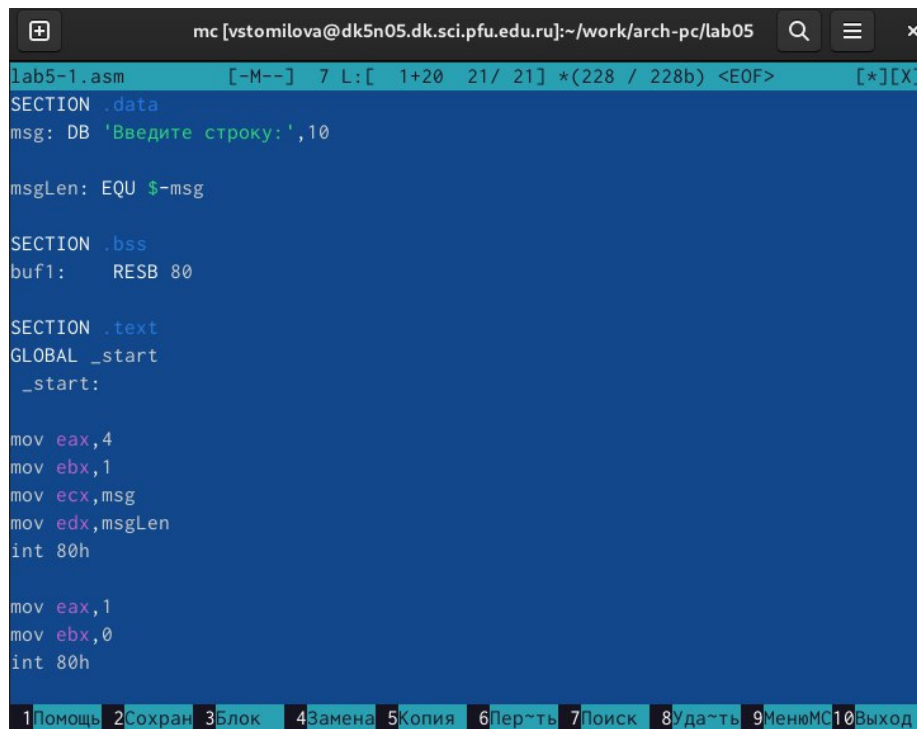


Рис. 4.5: Создание файла в Midnight Commander

4.2 Работа в NASM

С помощью F4 открываю только что созданный файл и вношу код с листинга (рис. 4.6).



```
mc [vstomilova@dk5n05.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab05
lab5-1.asm  [-M--]  7 L: [ 1+20  21/ 21] *(228 / 228b) <EOF>  [*][X]
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

SECTION .bss
buf1:   RESB 80

SECTION .text
GLOBAL _start
_start:

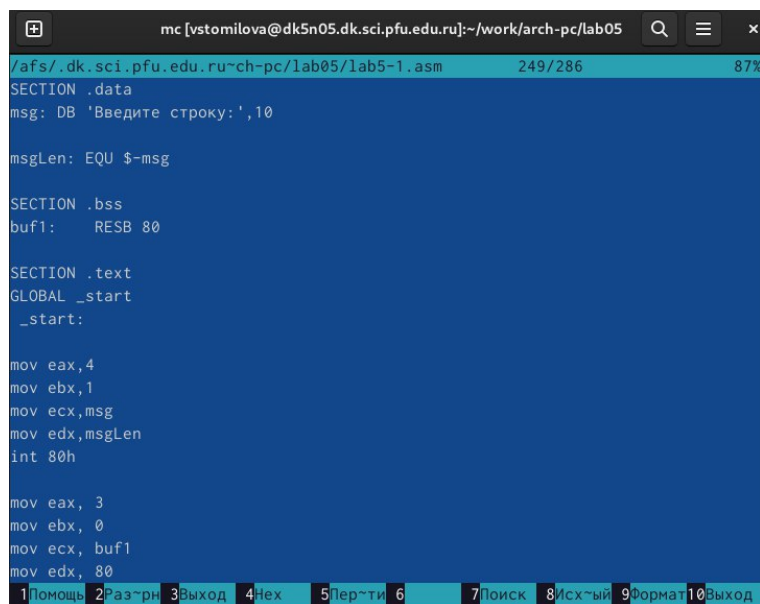
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax,1
mov ebx,0
int 80h

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переть 7Поиск 8Удалить 9МенюMC10Выход
```

Рис. 4.6: Редактирование файла в Midnight Commander

Проверяю сохраненные изменения с помощью клавиши F3 (рис. 4.7).



```
mc [vstomilova@dk5n05.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab05/lab5-1.asm 249/286 87%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

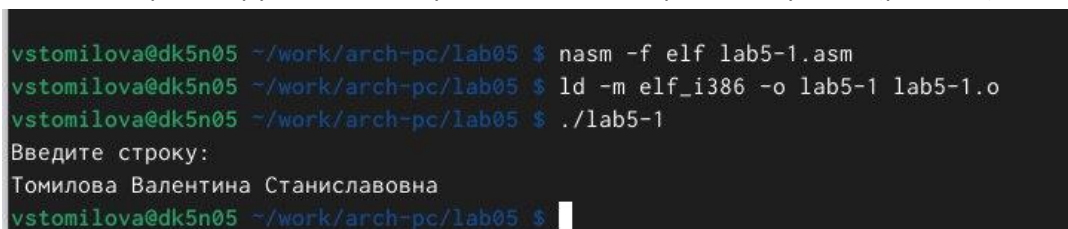
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80

1Помощь 2Разрн 3Выход 4Hex 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 4.7: Проверка сохранения сделанных изменений

Транслирую и компоную измененный файл, запускаю (рис. 4.8).



```
vstomilova@dk5n05 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
vstomilova@dk5n05 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
vstomilova@dk5n05 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Томилова Валентина Станиславовна
vstomilova@dk5n05 ~/work/arch-pc/lab05 $
```

Рис. 4.8: Трансляция, компоновка и последующий запуск программы

4.3 Подключение внешнего файла

Скаченный с ТУИС файл сохраняю в общую папку на своем компьютере, на виртуальной машине в интерфейсе Midnight Commander перехожу в директорию общей папки, перемещаю файл в рабочий подкаталог. (рис. 4.9).

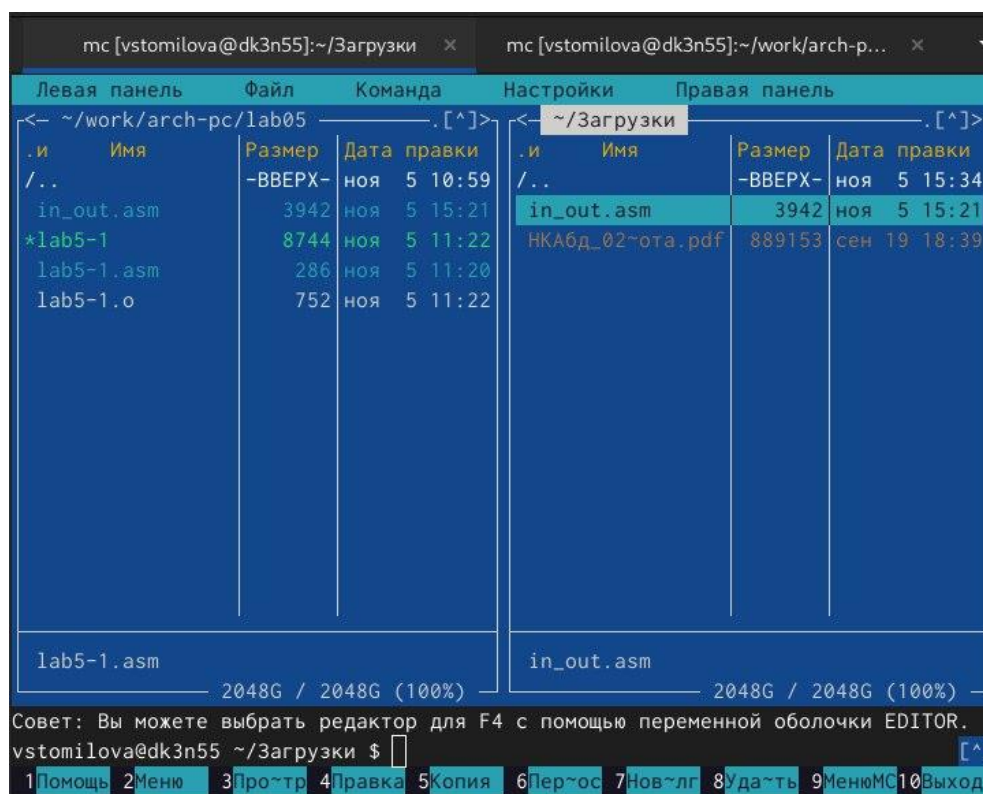


Рис. 4.9: Копирование файла в рабочий каталог

Создаю копию файла для последующей работы с ним (рис. 4.10).

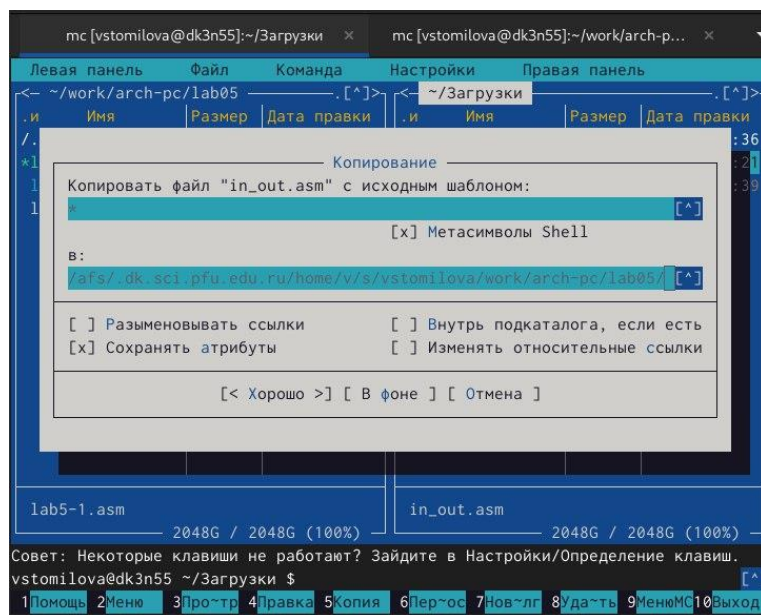


Рис. 4.10: Создание копии файла в Midnight Commander

В копии файла подключаю подпрограмм из подключенного файла (рис. 4.11).

```
.../.dk.sci.pfu.edu.ru/home/v/s/vstomilova/work/arch-pc/lab05/lab5-2.asm Изменё
#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

^G Справка    ^O Записать  ^F Поиск     ^K Вырезать  ^T Выполнить ^M-U Отмена
^X Выход      ^R ЧитФайл  ^\ Замена   ^U Вставить  ^C Позиция   M-E Повтор
```

Рис. 4.11: Изменение программы

Транслирую, компоную и запускаю программу с подключенным файлом (рис. 4.12).

```
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Tomilova Valentina

vstomilova@dk3n55 ~ $
```

Рис. 4.12: 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. 4.13).

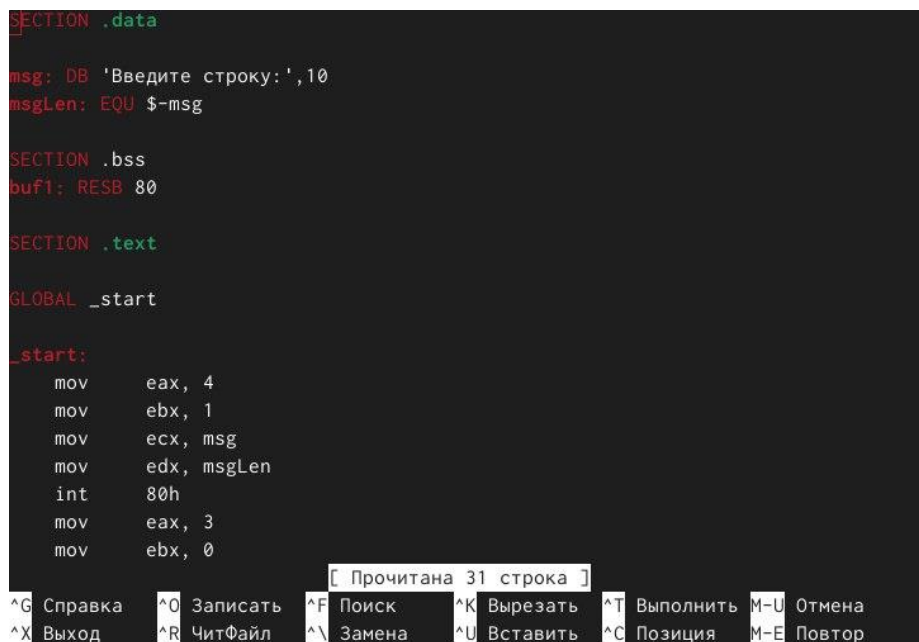
```
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ./lab-22
Введите строку: Tomilova Valentina

vstomilova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 4.13: Запуск изменной программы с другой подпрограммой

4.4 Задание для самостоятельной работы

Создаю копию lab5-1.asm, редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры (рис. 4.14).



```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text

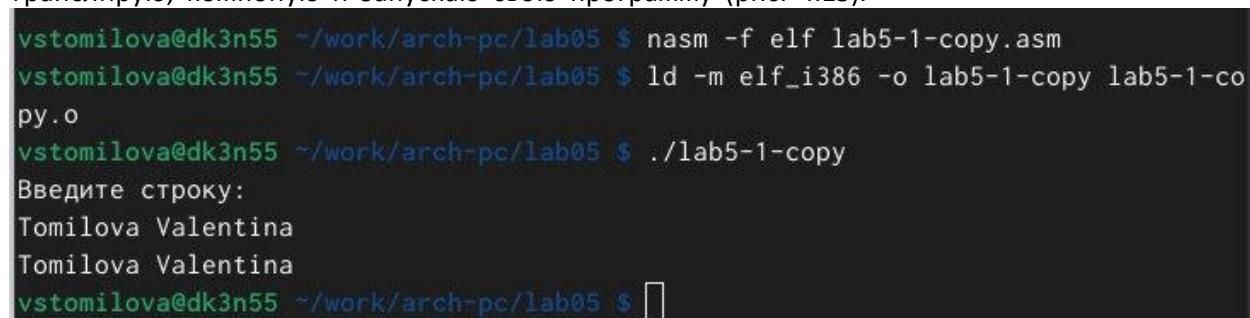
GLOBAL _start

_start:
    mov     eax, 4
    mov     ebx, 1
    mov     ecx, msg
    mov     edx, msgLen
    int     80h
    mov     eax, 3
    mov     ebx, 0

[ Прочитана 31 строка ]
^G Справка    ^O Записать  ^F Поиск     ^K Вырезать  ^T Выполнить ^M-U Отмена
^X Выход      ^R ЧитФайл  ^\ Замена    ^U Вставить  ^C Позиция   ^M-E Повтор
```

Рис. 4.14: Редактирование копии

Транслирую, компоную и запускаю свою программу (рис. 4.15).



```
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-copy.asm
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-copy lab5-1-copy.o
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-1-copy
Введите строку:
Tomilova Valentina
Tomilova Valentina
vstomilova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 4.15: Запуск своей программ

Создаю копию lab5-2.asm, редактирую так, чтобы в конце выводилась введенная мною строка с клавиатуры (рис. 4.16).

```
%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
```

^G Справка ^O Записать ^F Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

Рис. 4.16: Редактирование копии

Транслирую, компоную и запускаю свою программу (рис. 4.17).

```
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-copy.asm
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-copy lab5-2-copy.o
vstomilova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-2-copy
Введите строку: Tomilova
Tomilova
vstomilova@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 4.17: Запуск своей программы

5 Выводы

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

Список литературы

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