

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: Арсакаев Дени

Группа: НКАбд-05-24

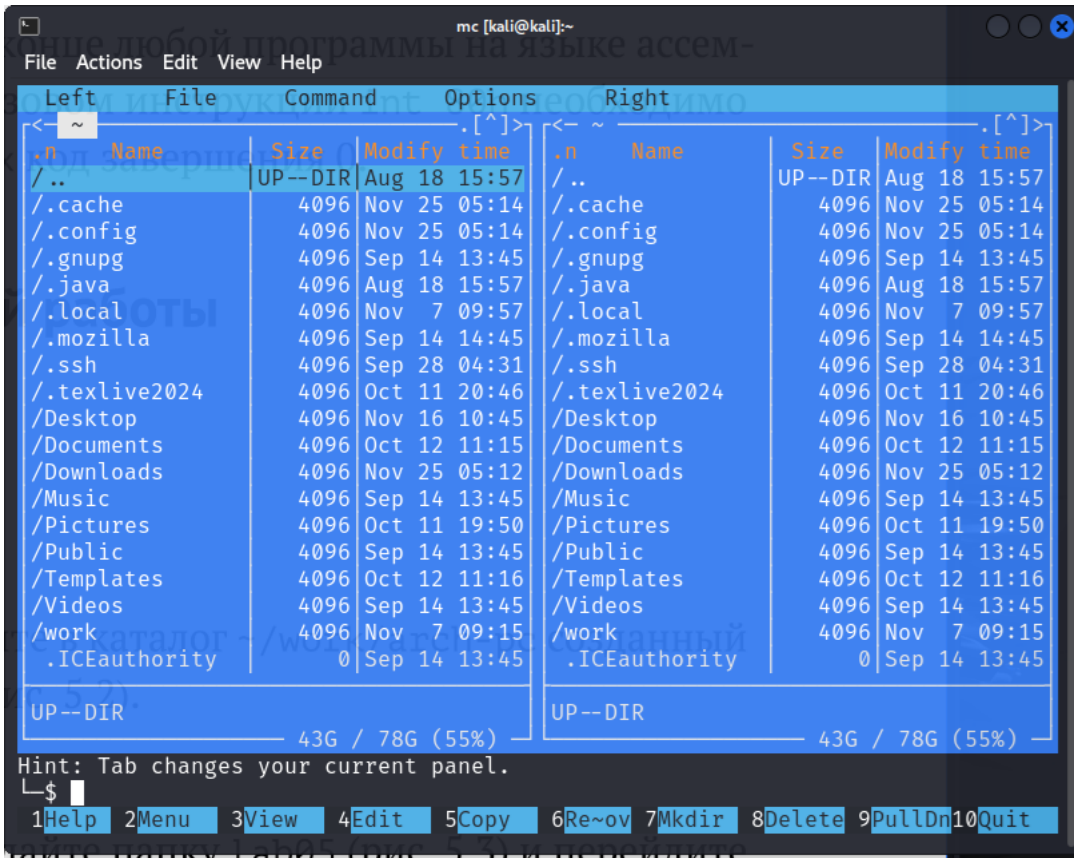
МОСКВА

2024 г.

Цель работы

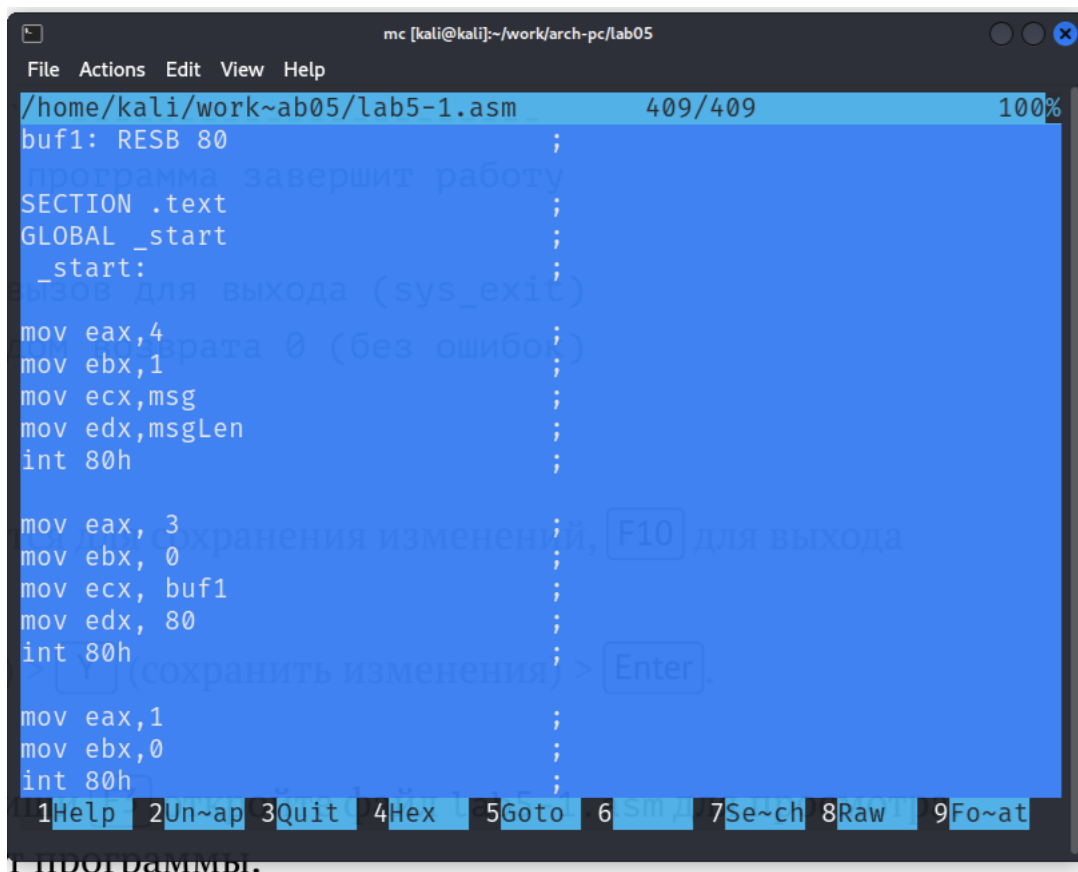
Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

Открываю Midnight Commander командой **mc**



Перехожу в каталог `~/work/study/2023-2024/Архитектура Компьютера/arch-pc`, используя файловый менеджер mc

С помощью функциональной клавиши F7 создаю каталог lab05



```
mc [kali@kali]:~/work/arch-pc/lab05
File Actions Edit View Help
/home/kali/work~ab05/lab5-1.asm 409/409 100%
buf1: RESB 80
;
SECTION .text
;
GLOBAL _start
;
_start:
;
; вызов для выхода (sys_exit)
;
mov eax,4
;
mov ebx,1
;
mov ecx,msg
;
mov edx,msgLen
;
int 80h
;

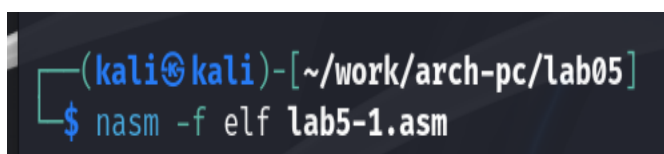
mov eax, 3
;
mov ebx, 0
;
mov ecx, buf1
;
mov edx, 80
;
int 80h
;

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

1Help 2Un~ap 3Quit 4Hex 5Goto 6 Sm~z 7Se~ch 8Raw 9Fo~at
```

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

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`.



```
(kali@kali)-[~/work/arch-pc/lab05]
$ nasm -f elf lab5-1.asm
```

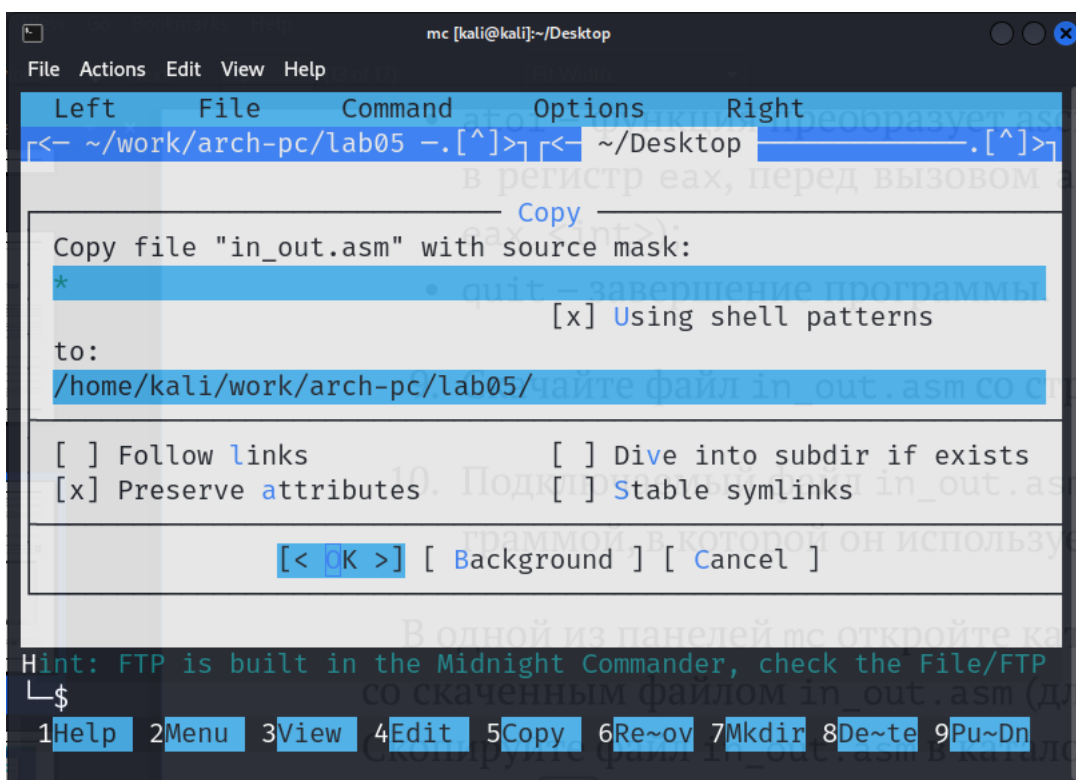
Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу

```
(kali㉿kali)-[~/work/arch-pc/lab05]
$ ld -m elf_i386 -o lab5-1 lab5-1.o

(kali㉿kali)-[~/work/arch-pc/lab05]
$ ./lab5-1
Введите строку:
Дени Арсакаев
```

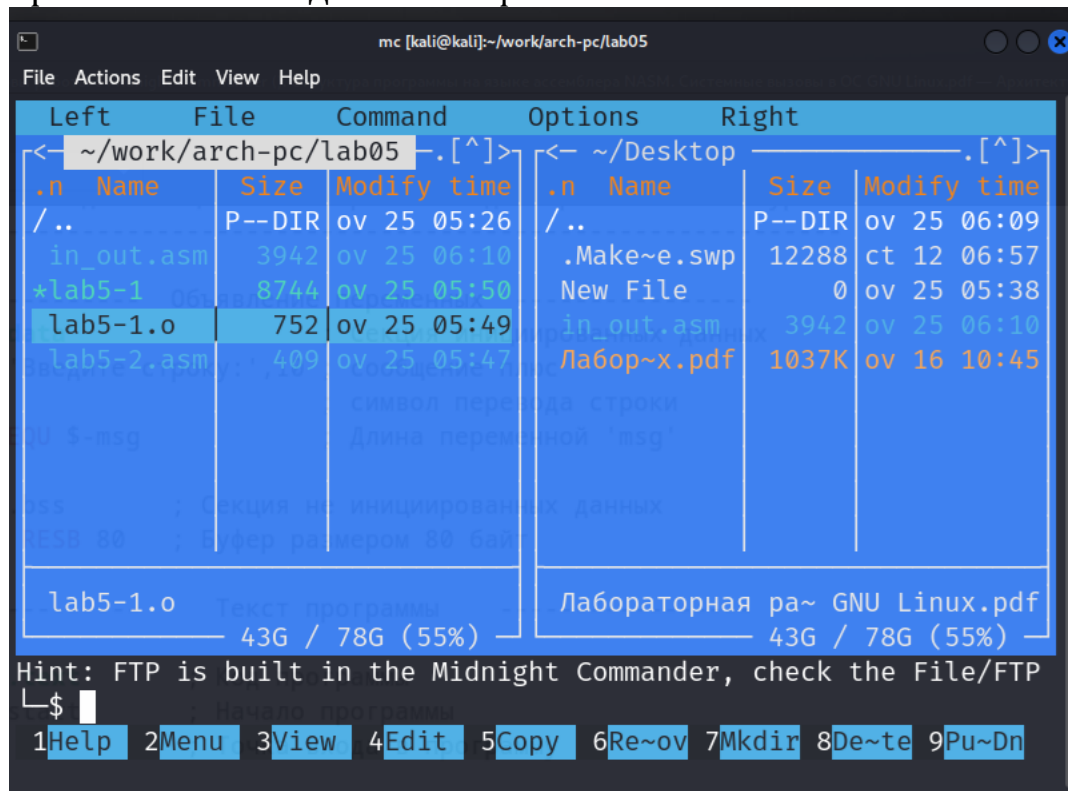
Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Download”

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Download в созданный каталог lab05

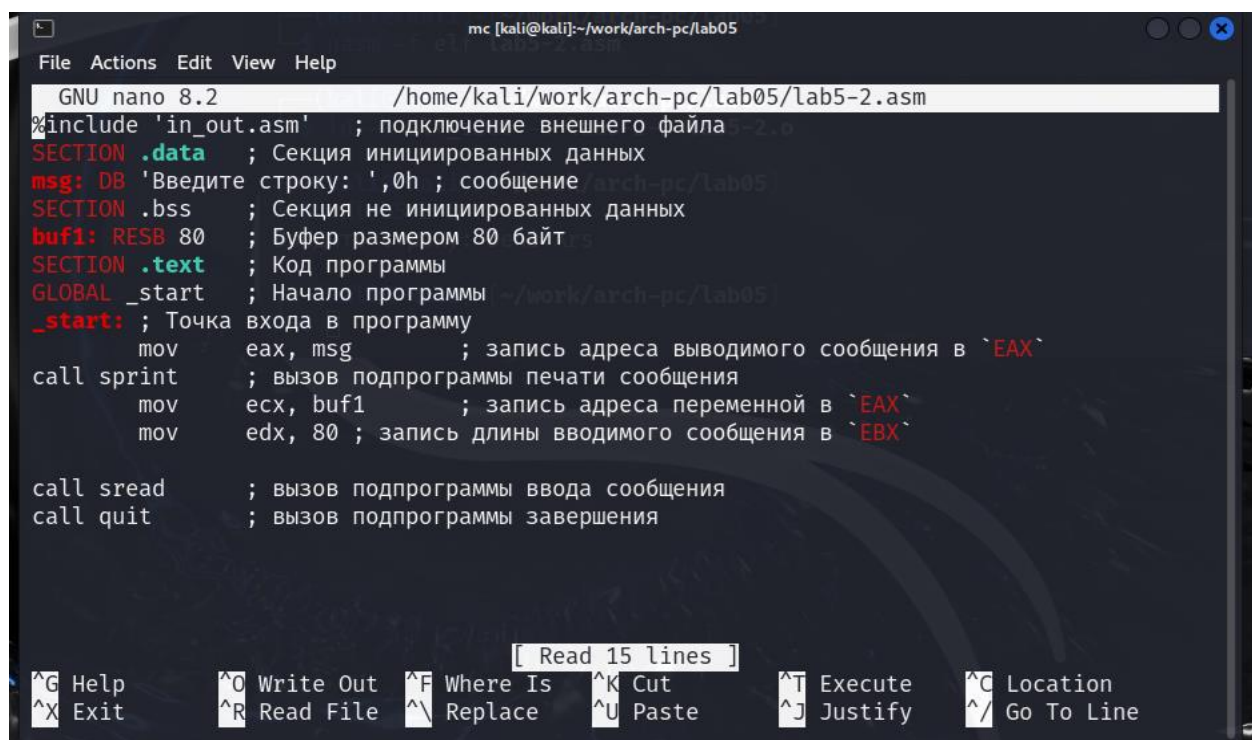


С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc

прописываю имя для копии файла



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



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

```
(kali㉿kali)-[~/work/arch-pc/lab05]  
$ ./lab5-2-2  
Введите строку:  
Deni Ars
```

Разница между sprint и sprintSL

sprint ==

```
(kali㉿kali)-[~/work/arch-pc/lab05]  
$ ./lab5-2-3  
Введите строку: Deni Ars
```

sprintSL ==

```
(kali㉿kali)-[~/work/arch-pc/lab05]  
$ ./lab5-2-2  
Введите строку:  
Deni Ars
```

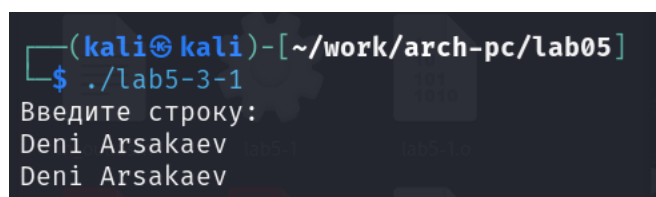
Выполнение заданий для самостоятельной работы

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

Код №1

```
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описание файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
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 ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описание файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```



```
(kali@kali)-[~/work/arch-pc/lab05]
$ ./lab5-3-1
Введите строку:
Deni Arsakaev
Deni Arsakaev
```

Код №2

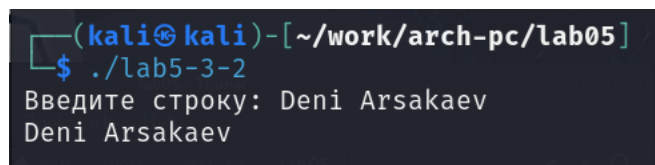
```
%include 'in_out.asm'
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
```



```

SECTION .bss ; Секция не инициализированных 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 sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Описание файла '1' - стандартный вывод
mov ecx, buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения

```



```

(kali㉿kali)-[~/work/arch-pc/lab05]
$ ./lab5-3-2
Введите строку: Deni Arsakaev
Deni Arsakaev

```

Итог:

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