

**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: Слабоспицкий Платон Сергеевич

Группа: НКАбд-05-25

**МОСКВА 2025 г.**

## Оглавление

1. Цель работы: .....	3
2. Выполнение лабораторной работы .....	4
3. Задания для самостоятельной работы .....	7
Вывод.....	10

## 1. Цель работы:

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

## 2. Выполнение лабораторной работы

Откроем Midnight Commander, перейдём в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4, далее создадим папку `lab05`, перейдём в него и с помощью команды **touch** создадим файл `lab5-1.asm`

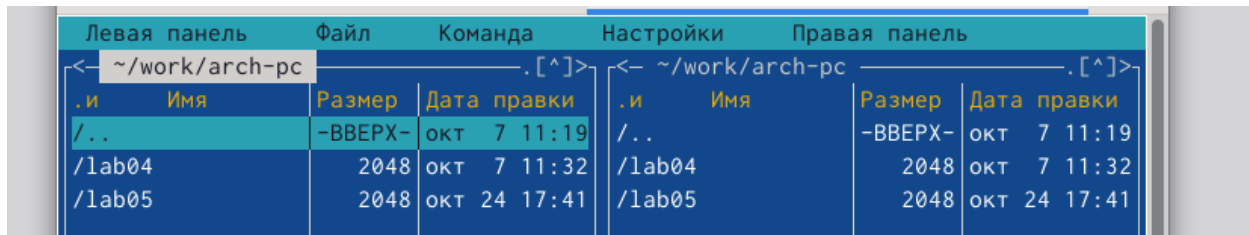


Рисунок 1 Создали папку lab05

```
psslabospickiyj@dk6n50 ~ $ touch lab5-1.asm
psslabospickiyj@dk6n50 ~ $
```

Рисунок 2 Используем команду touch

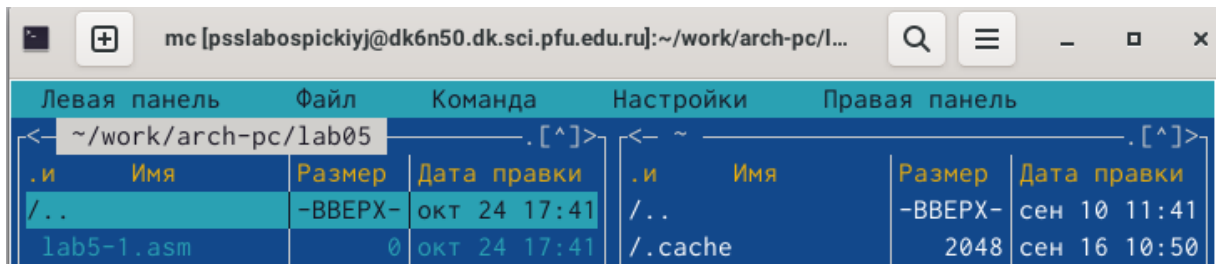
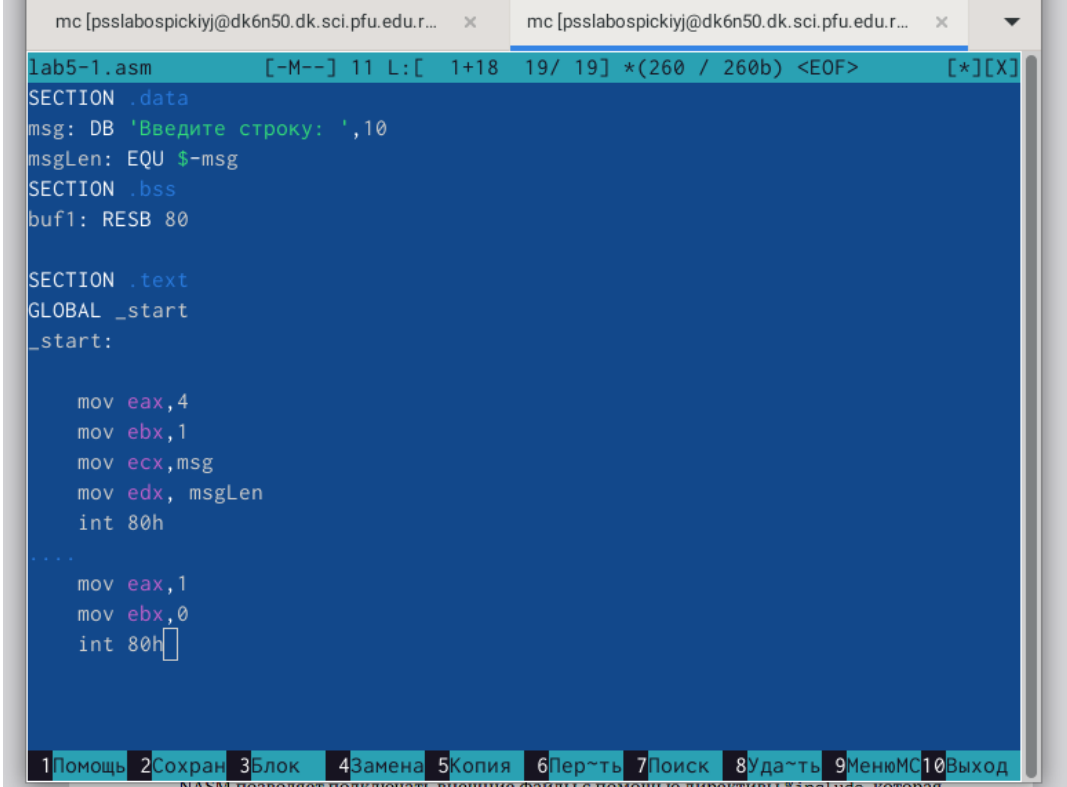


Рисунок 3 Создали файл lab5-1.asm в папке lab05

С помощью функциональной клавиши откроем файл *lab5-1.asm* для редактирования во встроенном редакторе, введём текст программы из листинга 5.1, сохраним и закроем её.



```
lab5-1.asm [-M--] 11 L:[ 1+18 19/ 19] *(260 / 260b) <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

    ....

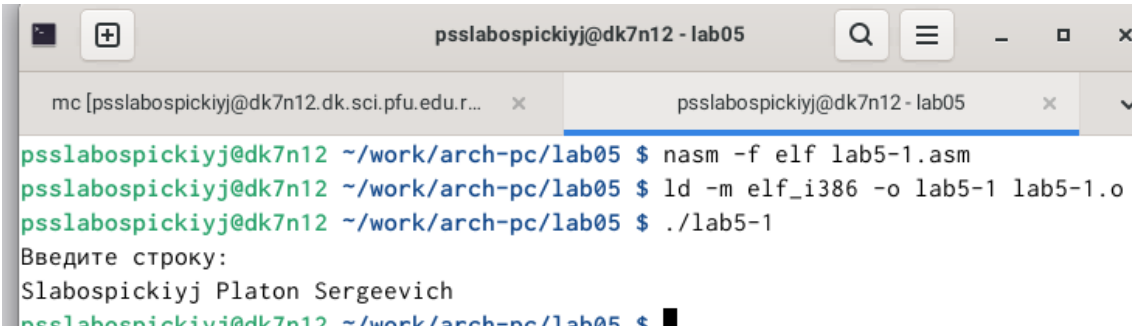
    mov eax,1
    mov ebx,0
    int 80h
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход

NASM позволяет подключать внешние файлы с помощью директивы %include, которая

Рисунок 4 Заполнили текст в соответствии с листингом 5.1

Оттранслируем текст программы *lab5-1.asm* в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа должна вывести строку 'Введите строку:', она будет ожидать ввода с клавиатуры. На запрос введём своё ФИО.



```
psslabospickiyj@dk7n12 - lab05
mc [psslabospickiyj@dk7n12.dk.sci.pfu.edu.r... x psslabospickiyj@dk7n12 - lab05 x
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Slabospickiyj Platon Sergeevich
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $
```

Рисунок 5 Работаем с нашим файлом, запускаем его

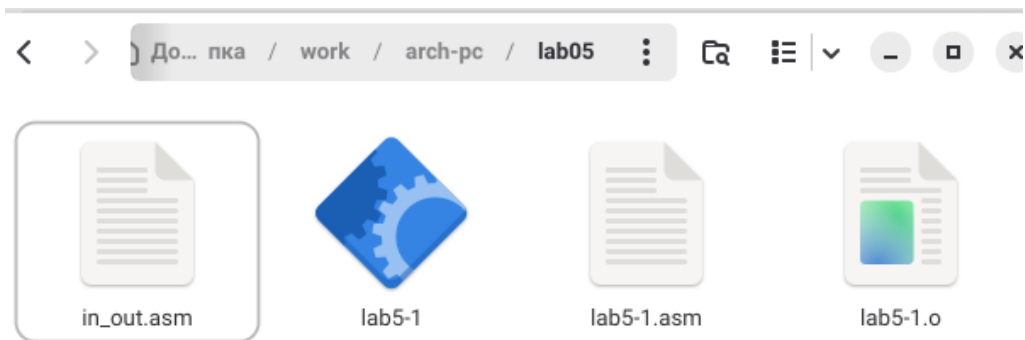


Рисунок 4 Наличие файлов *lab5-1* и файла *in\_out.asm* в одном каталоге

С помощью функциональной клавиши F6 создадим копию файла *lab5-1.asm* с именем *lab5-2.asm*. Исправим текст программы в файле *lab5-2.asm* с использованием подпрограмм из внешнего файла *in\_out.asm* в соответствии с листингом 5.2. Создадим исполняемый файл и проверим его работу.

```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
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, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рисунок 5 Текст программы из листинга 5.2

```
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
psslabospickiyj@dk7n12 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Slabospickiyj Platon Sergeevich
```

Рисунок 6 Создали файл *lab5-2*, выполнили компоновку объектного файла и запустили получившийся исполняемый файл

Заметим, что при использовании в *lab5-2.asm* подпрограмм из внешнего файла *in\_out.asm* запрашиваемая строка выводится сразу же, а не на следующей строке, как это было в файле *lab5-1.asm*.

### 3. Задания для самостоятельной работы

1) Создадим копию файла *lab5-1.asm* и назовём его *lab5-3.asm*.

Внесём изменения в программу (без использования внешнего файла *in\_out.asm*), так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран

Получим исполняемый файл и проверим его работу. На приглашение ввести строку введём свою фамилию.

```
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Slabospickii Platon Sergeevich
Slabospickii Platon Sergeevich
```

Рисунок 7 Выполнение компоновки объектного файла и его запуск

```

SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start:

; Вывод приглашения для ввода
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра

; Ввод строки с клавиатуры
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

; Сохраняем длину введенной строки для последующего вывода
mov esi, eax ; Сохраняем количество прочитанных байтов

; Вывод введенной строки на экран
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,buf1 ; Адрес буфера с введенной строкой
mov edx,esi ; Количество байтов для вывода (длина введенной строки)
int 80h ; Вызов ядра

; Завершение программы
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h

```

Рисунок 8 текст программы файла lab5-3.asm



2) Создадим копию файла *lab5-2.asm* с названием *lab5-4.asm* .

Исправим текст программы с использованием подпрограмм из внешнего файла *in\_out.asm*, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

Создадим исполняемый файл и проверим его работу.

```
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
psslabospickiyj@dk7n03 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Slabospickii Platon Sergeevich
Slabospickii Platon Sergeevich
```

Рисунок 9 Выполнение компоновки объектного файла и его запуск

```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',10; сообщение
SECTION .bss ; Секция не инициированных 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
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рисунок 10 текст программы файла *lab5-4.asm*

## **Вывод**

Приобрел практические навыки работы в Midnight Commander. Освоил инструкции языка ассемблера `mov` и `int`.