

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

дисциплина: Архитектура компьютера

Студент: Моисеев Дмитрий Вячеславович

Группа: НБИбд-02-25

МОСКВА

2025 г.

1. Цель работы: Целью данной лабораторной работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2. Выполнение лабораторной работы:

4.3.1. Программа Hello world! Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

Создадим

```
dvmoiseev@dk3n55 ~ $ mkdir -p ~/work/arch-pc/lab04
```

 каталог

для работы с программами на языке ассемблера NASM:

Перейдем в

```
dvmoiseev@dk3n55 ~ $ cd ~/wok/arch-pc/lab04
```

 созданный

каталог

Создадим текстовый файл с именем hello.asm

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ touch hello.asm
```

Откроем этот файл с помощью любого текстового редактора, например, gedit

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ gedit hello.asm
```

И введем в него следующий текст:

The screenshot shows a text editor window with the following assembly code:

```
1 ; hello.asm
2 SECTION .data
3     hello: DB 'Hello world!',10
4             ; Начало секции данных
        ; 'Hello world!' плюс
        ; символ перевода строки
5     helloLen: EQU $-hello
6 SECTION .text
7     GLOBAL _start
8 _start:
9     mov eax,4
10    mov ebx,1
11    mov ecx,hello
12    mov edx,helloLen
13    int 80h
14    mov eax,1
15    mov ebx,0
16    int 80h
```

Each line of assembly code has a corresponding Russian comment explaining its purpose.

At the bottom of the editor window, there are status bars with the text "Текст" and "Ширина табуляции: 8", and a cursor position indicator "Ln 4, Col 41".

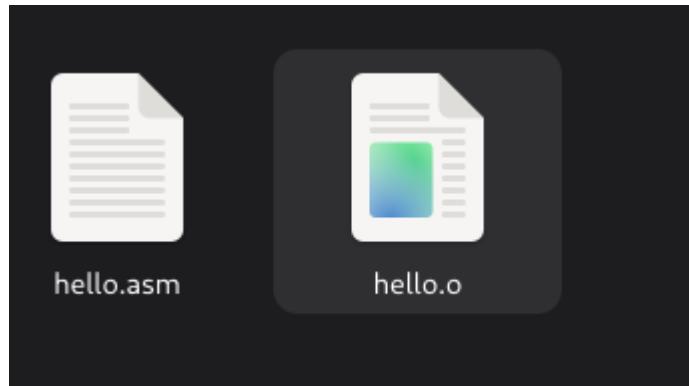
В отличие от многих современных высокоровневых языков программирования, в ассемблерной программе каждая команда располагается на отдельной строке. Размещение нескольких команд на одной строке недопустимо. Синтаксис ассемблера NASM является чувствительным к регистру, т.е. есть разница между большими и малыми буквами.

4.3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

Проверим, что объектный файл был создан.



4.3.3. Расширенный синтаксис командной строки NASM

```
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Выполним следующую команду:

С помощью
команды ls
проверим что

```
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ls  
hello.asm hello.o list.lst obj.o
```

файлы были созданы.

4.4. Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

```
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

С помощью команды ls проверим, что исполняемый файл hello был создан.

```
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ls  
hello hello.asm hello.o list.lst obj.o
```

4.4.1. Запуск исполняемого файла

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге,

можно,
набрав в
командной

```
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ./hello  
Hello world!  
dmoiseev@dk3n55 ~/work/arch-pc/lab04 $
```

строке:

3. Выполнение заданий для самостоятельной работы

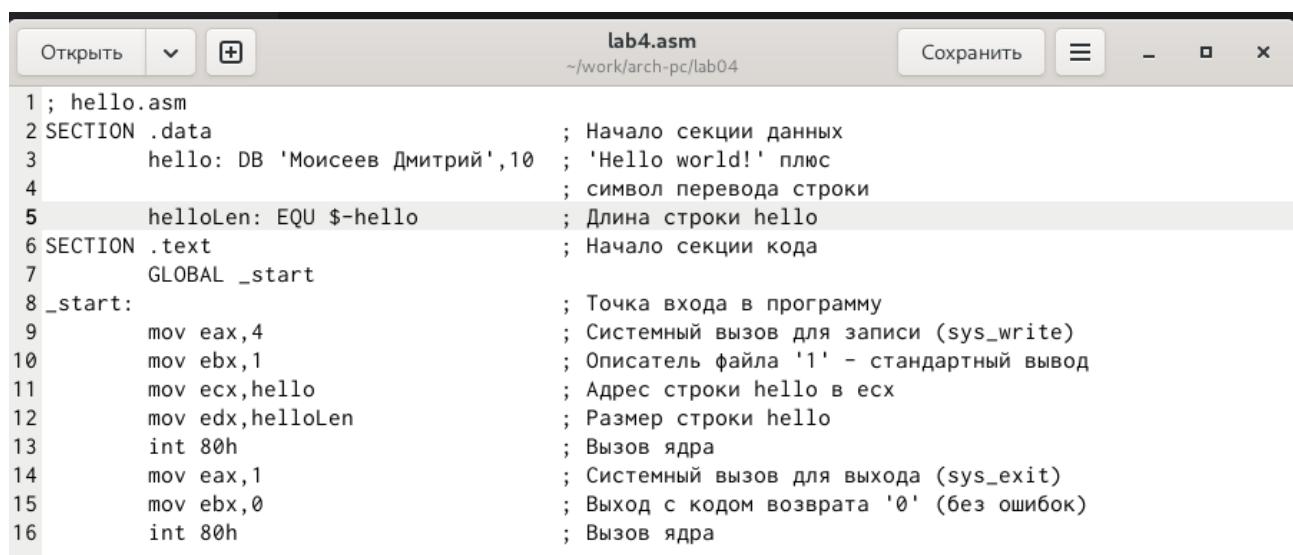
1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создадим копию файла

```
hell  
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm  
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ls  
hello hello.asm hello.o lab4.asm list.lst obj.o  
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $
```

именем lab4.asm

2. С помощью любого текстового редактора внесем изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем.

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ gedit lab4.asm
```



The screenshot shows a text editor window titled "lab4.asm" with the file path "~/work/arch-pc/lab04". The code is annotated with Russian comments:

```
1 ; hello.asm  
2 SECTION .data ; Начало секции данных  
3     hello: DB 'Моисеев Дмитрий',10 ; 'Hello world!' плюс  
4 ; символ перевода строки  
5     helloLen: EQU $-hello ; Длина строки hello  
6 SECTION .text ; Начало секции кода  
7     GLOBAL _start  
8 _start: ; Точка входа в программу  
9         mov eax,4 ; Системный вызов для записи (sys_write)  
10        mov ebx,1 ; Описатель файла '1' - стандартный вывод  
11        mov ecx,hello ; Адрес строки hello в ех  
12        mov edx,helloLen ; Размер строки hello  
13        int 80h ; Вызов ядра  
14        mov eax,1 ; Системный вызов для выхода (sys_exit)  
15        mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
16        int 80h ; Вызов ядра
```

3. Отранслируем полученный текст программы lab4.asm в объектный файл.

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
```

Выполним компоновку объектного файла и запустим получившийся исполняемый

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ./lab4
```

файл.

```
dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ./lab4
Моисеев Дмитрий
```

4. Скопируем файлы hello.asm и lab4.asm в наш локальный репозиторий в каталог ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузите файлы на Github.

Выводы:

В рамках данной работы мы освоенили процедуры компиляции и сборки программ,

н

а

п

и

с

а

н

н

ы

х

н

а

а

с

с

е

м