

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 ~/work/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:

```
Открыть  hello.asm  Сохранить
~/work/arch-pc/lab04

1 ; hello.asm
2 SECTION .data      ; Начало секции данных
3     hello: DB 'Hello world!',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 есх
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

Текст  Ширина табуляции: 8  Ln 4, Col 41  INS
```

В отличие от многих современных высокоуровневых 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.

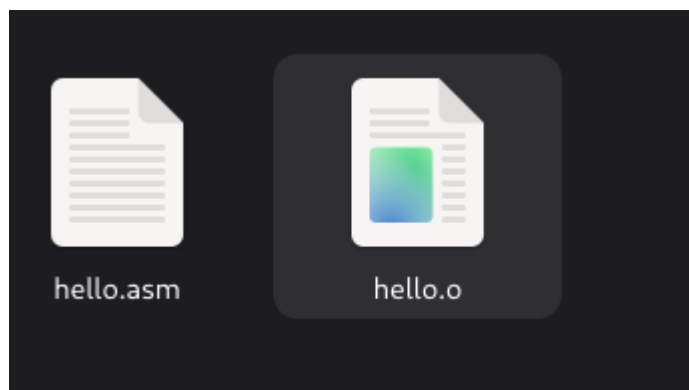
4.3.2. Транслятор NASM

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

```
dvmoiseev@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

файлы были созданы.

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

4.4. Компоновщик LD

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

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

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

```
dvmoiseev@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

строке:

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

3. Выполнение заданий для самостоятельной работы

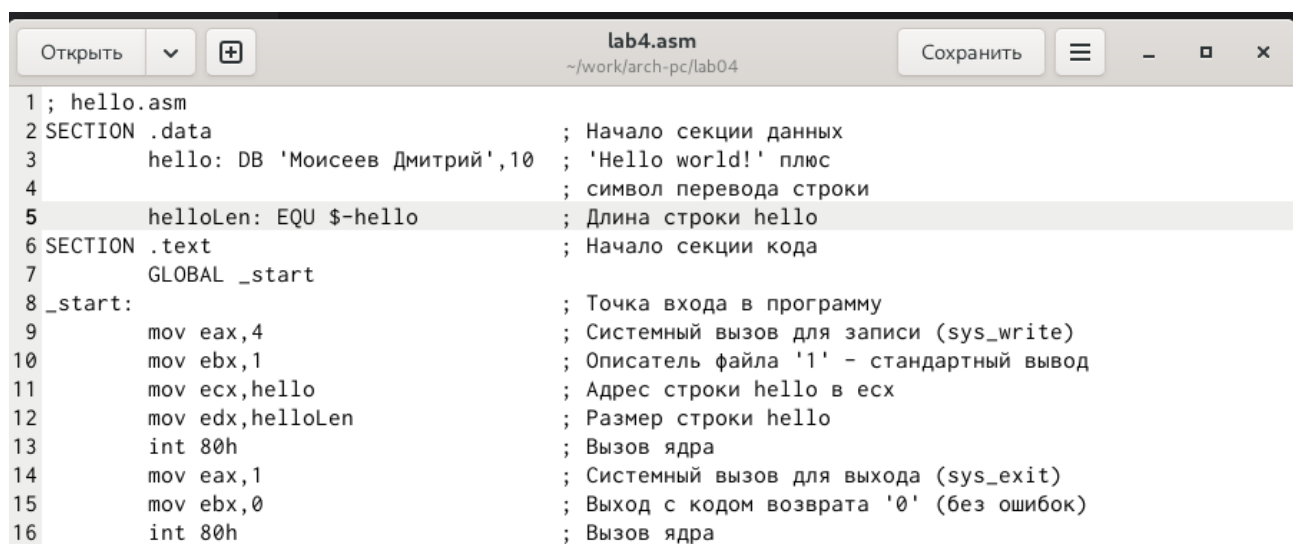
1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создадим копию файла hello

```
o.a dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
sm dvmoiseev@dk3n55 ~/work/arch-pc/lab04 $ ls
c 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
```



```
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 есх
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. Оттранслируем полученный текст программы lab4.asm в объектный файл.

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

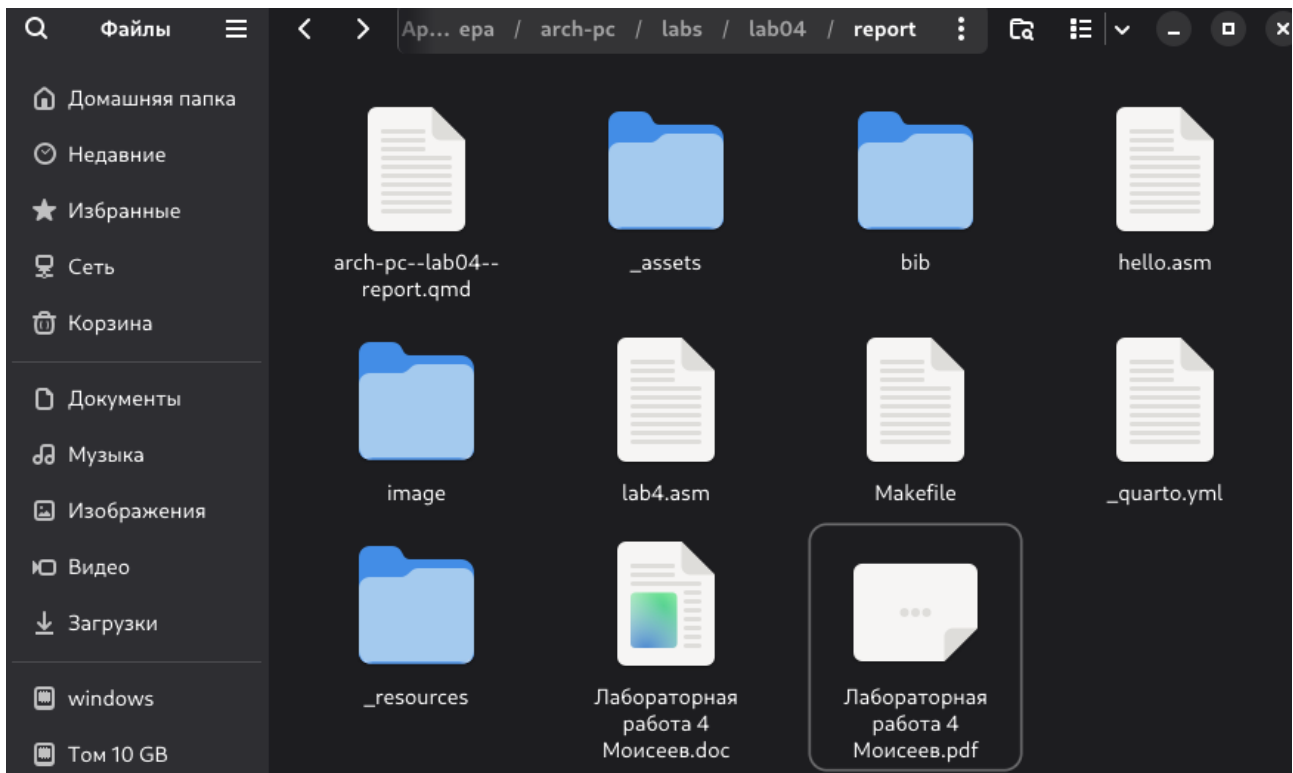
Выполним компоновку объектного файла и запустим получившийся исполняемый

```
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.



```

dvmoiseev@dk3n55 ~ $ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc
dvmoiseev@dk3n55 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git add .
dvmoiseev@dk3n55 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): add files lab-4'
Текущая ветка: master
Ваша ветка опережает «origin/master» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

нечего коммитить, нет изменений в рабочем каталоге
dvmoiseev@dk3n55 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 19, готово.
Подсчет объектов: 100% (19/19), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (12/12), готово.
Запись объектов: 100% (12/12), 779.06 КиБ | 5.30 МиБ/с, готово.
Total 12 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To github.com:dvmoiseev007/study_2025-2026_arch-pc.git
   d2fed5e..ce21098  master -> master
dvmoiseev@dk3n55 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ █

```

Выводы:

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

н

а

п

и

с

а

н

н

ы

х

н

а

а

с

с

е

м

б

л

е

р