

# 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: Мурылев И.В.

Группа: НПИбд-03-25

МОСКВА

2025г.

## **Содержание**

<b>1. Цель работы</b>	<b>2</b>
<b>2. Теоретическое введение</b>	<b>2</b>
<b>3. Задание</b>	<b>3</b>
<b>4. Выполнение лабораторной работы</b>	<b>4</b>
<b>5. Выводы</b>	<b>6</b>
<b>Список литературы</b>	

## Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## Теоретическое введение

NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64. Типичный формат записи команд NASM имеет вид: [метка:] мнемокод [операнд {, операнд}] [; комментарий]

Здесь мнемокод — непосредственно мнемоника инструкции процессору, которая является обязательной частью команды. Операндами могут быть числа, данные, адреса регистров или адреса оперативной памяти. Метка — это идентификатор, с которым ассемблер ассоциирует некоторое число, чаще всего адрес в памяти. Т.о. метка перед командой связана с адресом данной команды.

Допустимыми символами в метках являются буквы, цифры, а также следующие символы:

`_`, `$`, `#`, `@`, `~`, `.` и `?`.

Начинаться метка или идентификатор могут с буквы, `.`, `_` и `?`. Перед идентификаторами, которые пишутся как зарезервированные слова, нужно писать `$`, чтобы компилятор трактовал его верно (так называемое экранирование).

Максимальная длина идентификатора 4095 символов.

Программа на языке ассемблера также может содержать директивы — инструкции, не переводящиеся непосредственно в машинные команды, а управляющие работой транслятора.

Например, директивы используются для определения данных (констант и переменных) и обычно пишутся большими буквами.

## Задание

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/`. Загрузите файлы на Github.

## Выполнение лабораторной работы

1. Перемещение в локальный репозиторий по адресу `/home/ivmurihlev/Documents/work/study/2025-2026/Arch-pc/arch_pc/arch_evm/labs/lab04/report`.

```
[ivmurihlev@PersonalMach report]$ pwd
/home/ivmurihlev/Documents/work/study/2025-2026/Arch-pc/arch_pc/arch_evm/labs/lab04/report
```

Копирование файла `hello.asm` в `lab4.asm` и проверка успешности копирования `ls`.

```
[ivmurihlev@PersonalMach report]$ ls
arch-pc--lab04--report.qmd  _assets  bib  hello.asm  image  Makefile  _quarto.yml  report.md  _resources
[ivmurihlev@PersonalMach report]$ cp hello.asm lab4.asm
[ivmurihlev@PersonalMach report]$ ls
arch-pc--lab04--report.qmd  _assets  bib  hello.asm  image  lab4.asm  Makefile  _quarto.yml  report.md  _resources
```

2. После изменения проверяем командой `cat`.

```
[ivmurihlev@PersonalMach report]$ cat lab4.asm
SECTION .data
    hello:      db "Мурылев Иван",0xa
               helloLen: equ $ - hello
SECTION .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, helloLen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

3. Транслируем файл.

```
[ivmurihlev@PersonalMach report]$ nasm -f elf lab4.asm
[ivmurihlev@PersonalMach report]$ ls
arch-pc--lab04--report.qmd  bib  image  lab4.o  _quarto.yml  _resources
_assets  hello.asm  lab4.asm  Makefile  report.md
```

И как в примере создадим объектный файл.

```
[ivmurihlev@PersonalMach report]$ nasm -o obiekt.o -f elf -g -l list.lst lab4.asm
[ivmurihlev@PersonalMach report]$ ls
arch-pc--lab04--report.qmd  bib  image  lab4.o  Makefile  _quarto.yml  _resources
_assets  hello.asm  lab4.asm  list.lst  obiekt.o  report.md
```

Через него компанируем и получаем исполняемую программу (Созданы 2 объектных файлов)

```
[ivmurihlev@PersonalMach report]$ ld -m elf_i386 lab4.o -o lab4
[ivmurihlev@PersonalMach report]$ ls
arch-pc--lab04--report.qmd  bib  image  lab4.asm  list.lst  obiekt.o  report.md
_assets  hello.asm  lab4  lab4.o  Makefile  _quarto.yml  _resources
```

4. Запуск исполняемых файлов

```
[ivmurihlev@PersonalMach report]$ ./main
```

Мурылев Иван

```
[ivmurihlev@PersonalMach report]$ ./lab4
```

Мурылев Иван

## **Выводы**

Были освоены базовые команды Ассамблера и его различие от других языков программирования.