

# **Отчет по лабораторной работе №4**

**Дисциплина: Архитектура компьютера**

Стрижов Дмитрий Павлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Задание</b>	<b>4</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
3.1	Программа Hello world! . . . . .	5
3.2	Транслятор NASM . . . . .	6
3.3	Расширенный синтаксис командной строки NASM . . . . .	6
3.4	Компоновщик LD . . . . .	7
3.5	Запуск исполняемого файла . . . . .	8
3.6	Задание для самостоятельной работы . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>10</b>
	<b>Список литературы</b>	<b>11</b>

# 1 Цель работы

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

## 2 Задание

1. Программа Hello world!
2. Транслятор NASM
3. Расширенный синтаксис командной строки NASM
4. Компоновщик LD
5. Запуск исполняемого файла
6. Задание для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Программа Hello world!

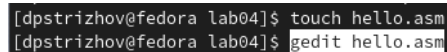
Создаем каталог для работы с программами на языке ассемблера NASM и переходим в него (рис. 3.1).



```
[dpstrizhov@fedora ~]$ mkdir -p ~/work/arch-pc/lab04  
[dpstrizhov@fedora ~]$ cd ~/work/arch-pc/lab04
```

Рис. 3.1: Репозиторий для работы с программами на языке ассемблера NASM

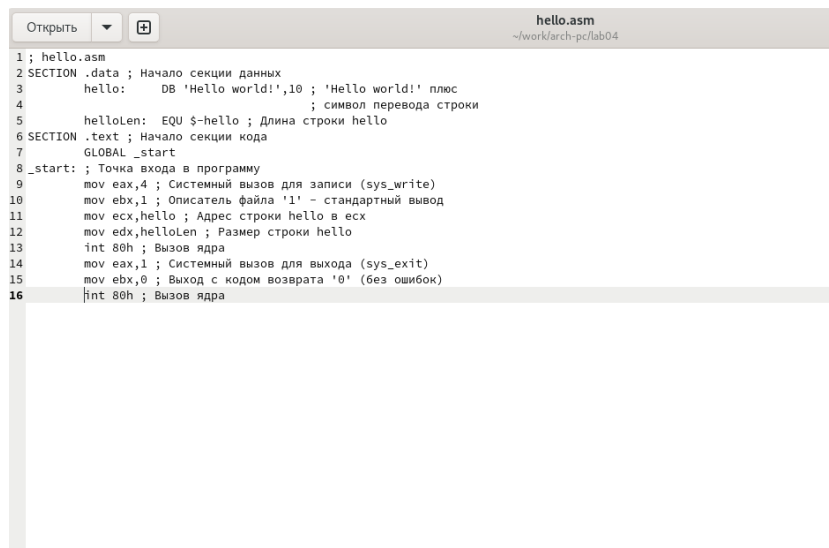
Создаем текстовый файл с именем hello.asm и открываем его(рис. 3.2).



```
[dpstrizhov@fedora lab04]$ touch hello.asm  
[dpstrizhov@fedora lab04]$ gedit hello.asm
```

Рис. 3.2: Создание текстового файла hello.asm

Вводим программу(рис. 3.3).



```
1; hello.asm
2SECTION .data ; Начало секции данных
3    hello:    DB 'Hello world!',10 ; 'Hello world!' плюс
4              ; символ перевода строки
5    helloLen: EQU $-hello ; Длина строки hello
6SECTION .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 ecx
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
```

Рис. 3.3: Программа выводящая “Hello world!”

## 3.2 Транслятор NASM

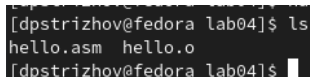
Компилируем код(рис. 3.4).



```
[dpstrizhov@fedora report]$ nasm -f elf hello.asm
```

Рис. 3.4: Команда для компиляции кода

Проверяем создание нужного файла(рис. 3.5).



```
[dpstrizhov@fedora lab04]$ ls
hello.asm  hello.o
[dpstrizhov@fedora lab04]$
```

Рис. 3.5: Проверка компиляции

Объектный файл имеет имя hello.o.

## 3.3 Расширенный синтаксис командной строки NASM

Выполняем следующую команду(рис. 3.6).

```
[dpstrizhov@fedora report]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 3.6: Компиляция кода

Проверяем наличие созданных файлов (рис. 3.7).

```
[dpstrizhov@fedora lab04]$ ls
hello.asm hello.o list.lst obj.o
[dpstrizhov@fedora lab04]$
```

Рис. 3.7: Проверка создания файлов

## 3.4 Компоновщик LD

Передаем объектный файл на обработку компоновщику (рис. 3.8).

```
[dpstrizhov@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[dpstrizhov@fedora lab04]$
```

Рис. 3.8: Обработка объектного файла

Проверяем наличие исполняемого файла (рис. 3.9).

```
[dpstrizhov@fedora lab04]$ ls
hello hello.asm hello.o list.lst obj.o
[dpstrizhov@fedora lab04]$
```

Рис. 3.9: Проверка наличия исполняемого файла

Выполняем следующую программу (рис. 3.10).

```
[dpstrizhov@fedora lab04]$ ld -m elf_i386 obj.o -o main
[dpstrizhov@fedora lab04]$
```

Рис. 3.10: Компоновка файла obj.o

Исполняемый файл будет иметь имя main, а объектный файл будет иметь имя obj.o.

## 3.5 Запуск исполняемого файла

Запускаем нашу программу (рис. 3.11).

```
[dpstrizhov@fedora lab04]$ ./hello
Hello world!
```

Рис. 3.11: Запуск программы

## 3.6 Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создаю копию файла `hello.asm` с именем `lab4.asm` (рис. 3.12).

```
[dpstrizhov@fedora lab04]$ cp hello.asm lab4.asm
[dpstrizhov@fedora lab04]$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main.o  obj.o
```

Рис. 3.12: Создание копии `hello.asm`

2. Вношу изменения в файле `lab4.asm` так, чтобы вместо `Hello world!` выводилась строка с моими именем и фамилией (рис. 3.13).

```
1: hello.asm
2 SECTION .data ; Начало секции данных
3     hello:    DB 'Стризов 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 ecx
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
```

Рис. 3.13: Изменения в файле `lab4.asm`

3. Создаем объектный файл, компонуем его и запускаем (рис. 3.14, рис. 3.15, рис. 3.16).

```
[dpstrizhov@fedora lab04]$ nasm -f elf lab4.asm
```

Рис. 3.14: Создание объектного файла



```
[dpstrizhov@fedora lab04]$ ld -m elf_i386 lab4.o -o lab4
```

Рис. 3.15: Компиляция файла

```
[dpstrizhov@fedora lab04]$ ./lab4  
Стрижов Дмитрий
```

Рис. 3.16: Запуск файла lab4

4. Копирую файлы hello.asm и lab4.asm в мой локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 3.17 ). Загружаю файлы на Github (рис. 3.18 ).

```
[dpstrizhov@fedora lab04]$ cp hello.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/
```

Рис. 3.17: Копию файлы в мой локальный репозиторий

```
[dpstrizhov@fedora report]$ git push  
Перечисление объектов: 36, готово.  
Подсчет объектов: 100% (36/36), готово.  
При сжатии изменений используется до 4 потоков  
Сжатие объектов: 100% (31/31), готово.  
Запись объектов: 100% (31/31), 369.58 КиБ | 2.78 МиБ/с, готово.  
Всего 31 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.  
To github.com:StrizhovDmitriy/study_2023-2024_arch-pc-.git  
e493498..1386594 master -> master
```

Рис. 3.18: Отправляю файлы на Github

## **4 Выводы**

В ходе данной лабораторной работы я освоил процедуру компиляции и сборки программ, написанных на ассемблере NASM.

## Список литературы

Что такое ассемблер и нужно ли его изучать. Источник: <https://skillbox.ru/media/code/что-такое-asmembler/?ysclid=lo5rfb07ik426340922>