

# **Шаблон отчёта по лабораторной работе №4**

**Дисциплина: Архитектура компьютера**

Вакутайпа Милдред

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение заданий для самостоятельной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>
<b>6</b>	<b>Список литературы</b>	<b>13</b>

# Список иллюстраций

3.1	Рис 1.0	. . . . .	6
3.2	Рис 1.1	. . . . .	6
3.3	Рис 1.2	. . . . .	6
3.4	Рис 1.3	. . . . .	7
3.5	Рис 2.0	. . . . .	7
3.6	Рис 2.1	. . . . .	7
3.7	Рис 3.0	. . . . .	7
3.8	Рис 4.0	. . . . .	8
3.9	Рис 4.1	. . . . .	8
3.10	Рис 5.0	. . . . .	8
4.1	Рис 6.0	. . . . .	9
4.2	Рис 6.1	. . . . .	9
4.3	Рис 6.2	. . . . .	9
4.4	Рис 6.3	. . . . .	10
4.5	Рис 6.4	. . . . .	10
4.6	Рис 6.5	. . . . .	10
4.7	Рис 6.6	. . . . .	10
4.8	Рис 6.7	. . . . .	10
4.9	Рис 6.8	. . . . .	11

# 1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла

## 3 Выполнение лабораторной работы

### 1. Создание программы Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в созданный каталог:

```
mwakutaipa@mwakutaipa:~$ mkdir -p ~/work/arch-pc/lab04
mwakutaipa@mwakutaipa:~$ cd ~/work/arch-pc/lab04
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 3.1: Рис 1.0

Создаю текстовый файл с именем hello.asm и проверяю с помощью ls:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ touch hello.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello.asm
```

Рис. 3.2: Рис 1.1

Открываю созданный файл в текстовом редакторе gedit и вставляю в него программу для вывода “Hello world!”:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.3: Рис 1.2

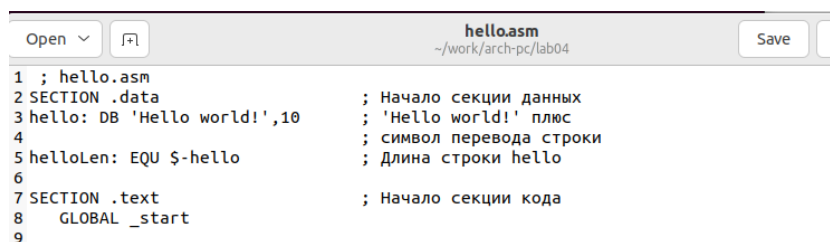


Рис. 3.4: Рис 1.3

## 2. Работа с транслятором NASM

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`. Потом проверяю с помощью `ls`:

```

mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ nasm -f elf hello.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$

```

Рис. 3.5: Рис 2.0

```

mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$

```

Рис. 3.6: Рис 2.1

## 3. Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o` и проверяю с помощью `ls`:

```

mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst
hello.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$

```

Рис. 3.7: Рис 3.0

## 4. Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello и проверяю с помощью ls:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 3.8: Рис 4.0

Выполняю следующую команду (рис. 4.1). Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 3.9: Рис 4.1

## 5. Запуск исполняемого файла

Запускаю на выполнение созданный файл hello:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ./hello
Hello world!
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 3.10: Рис 5.0



## 4 Выполнение заданий для самостоятельной работы

С помощью `cp` создаю копию файла `hello.asm` с именем `lab4.asm`:

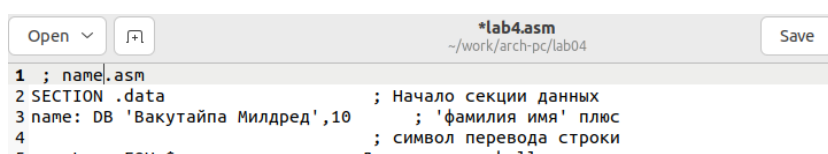
```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 4.1: Рис 6.0

С помощью `gedit` открываю файл `lab4.asm` и вношу изменения в программу так, чтобы она выводила имя и фамилию:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис. 4.2: Рис 6.1



```
1 ; name.asm
2 SECTION .data
3 name: DB 'Вакутайпа Милдред',10
4 ; символ перевода строки
5 ; начало секции .data
```

Рис. 4.3: Рис 6.2

Компилирую текст программы в объектный файл и проверяю с помощью `ls`, что файл `lab4.o` создан:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 4.4: Рис 6.3

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить файл lab4:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 4.5: Рис 6.4

Запускаю исполняемый файл lab4:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ ./lab4
Вакутайпа Милдред
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$
```

Рис. 4.6: Рис 6.5

Копирую файлы hello.asm и lab4.asm в локальный репозиторий с помощью ср:

```
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/
"Архитектура компьютера"/arch-pc/labs/lab04/
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/
"Архитектура компьютера"/arch-pc/labs/lab04/
mwakutaipa@mwakutaipa:~/work/arch-pc/lab04$ cd ~/work/study/2023-2024/"Архитекту
ра компьютера"/arch-pc/labs/lab04/
mwakutaipa@mwakutaipa:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs
/lab04$ ls
hello.asm lab4.asm presentation report
mwakutaipa@mwakutaipa:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs
/lab04$
```

Рис. 4.7: Рис 6.6

С помощью команд git add . и git commit добавляю файлы на GitHub:

```
/lab04$ git add .
mwakutaipa@mwakutaipa:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs
/lab04$ git commit -m "Add files"
[master 60e77a0] Add files
2 files changed, 40 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
```

Рис. 4.8: Рис 6.7

Отправляю файлы на сервер с помощью команды git push:

```
mwakutaipa@mwakutaipa:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs
/Lab04$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.05 KiB | 41.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:wakutaipa/study1_2023-2024_arh-pc.git
  2ca6e15..60e77a0  master -> master
mwakutaipa@mwakutaipa:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs
/Lab04$
```

Рис. 4.9: Рис 6.8

## **5 Выводы**

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## **6 Список литературы**

Архитектура ЭВМ