

# **Отчет по лабораторной работе №4**

**Архитектура компьютера**

Николенко Анна Николаевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>

## Список иллюстраций

4.1	Создание каталога для дальнейшей работы, переход в этот каталог и создание текстового файла . . . . .	8
4.2	Открытие файла . . . . .	8
4.3	Компиляция текста и проверка создания объектного файла . . . .	8
4.4	Компиляция исходного файла и проверка создания всех созданных файлов . . . . .	9
4.5	Передача на обработку компоновщику объектного файла и проверки создания файла . . . . .	9
4.6	Задача имени создаваемого исполняемого файла . . . . .	9
4.7	Запуск исполняемого файла . . . . .	9
4.8	Создание копии файла с другим именем . . . . .	10
4.9	Копирование файлов в каталог локального репозитория . . . . .	10

## Список таблиц

# 1 Цель работы

Цель работы заключается в освоении процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Задание

вывести программу “Hello world!” и своё ФИО.

### 3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 4.1). Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры.

## 4 Выполнение лабораторной работы

Создаю каталог для работы с программами на языке ассемблера NASM, перехожу в этот каталог и создаю текстовый файл с именем hello.asm (рис. 4.1).

```
annikolenko@dk8n75 ~ $ mkdir -p ~/work/arch-pc/lab04
annikolenko@dk8n75 ~ $ cd ~/work/arch-pc/lab04
annikolenko@dk8n75 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис. 4.1: Создание каталога для дальнейшей работы, переход в этот каталог и создание текстового файла

Открываю этот файл с помощью текстового редактора gedit и ввожу в него текст из лабораторной работы (рис. 4.2).

```
annikolenko@dk4n59 ~ $ gedit hello.asm
□
```

Рис. 4.2: Открытие файла

Для компиляции приведённого выше текста программы «Hello World» пишу команду `nasm -f elf hello.asm`, с помощью команды `ls` проверяю, что объектный файл был создан (рис. 4.3).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 4.3: Компиляция текста и проверка создания объектного файла



Компилирую исходный файл `hello.asm` в `obj.o` с помощью команды `nasm -o obj.o -f elf -g -l list.lst hello.asm`. С помощью команды `ls` проверьте, что файлы были созданы (рис. 4.4).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.4: Компиляция исходного файла и проверка создания всех созданных файлов

Чтобы получить исполняемую программу, необходимо передать на обработку компоновщику объектный файл с помощью команды: `ld -m elf_i386 hello.o -o hello`. А затем с помощью команды `ls` проверяю, что исполняемый файл `hello` был создан (рис. 4.5).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.5: Передача на обработку компоновщику объектного файла и проверки создания файла

Выполните следующую команду, задаю в данном случае имя создаваемого исполняемого файла: `ld -m elf_i386 obj.o -o main` (рис. 4.6).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 4.6: Задача имени создаваемого исполняемого файла

Запускаю на выполнение созданный исполняемый файл (рис. 4.7).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ./hello
Hello, world!
```

Рис. 4.7: Запуск исполняемого файла

#Выполнение заданий для самостоятельной работы

В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаю копию файла `hello.asm` с именем `lab4.asm` (рис. 4.8).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
```

Рис. 4.8: Создание копии файла с другим именем

С помощью текстового редактора `gedit` (рис. ??) вношу изменения в текст программы в файле `lab4.asm`: вместо `Hello world!` пишу мои фамилию и имя. На экран теперь будет выводиться строка с моими фамилией и именем (рис. ??).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ gedit lab4.asm
```

```
1 SECTION .data
2     hello:      db "Никол
3                 helloLen:  e
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
```

Оттранслирую полученный текст программы `lab4.asm` в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл (рис. ??), (рис. ??).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
```

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
```

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ld
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ ld
```

Копирую файлы `hello.asm` и `lab4.asm` в свой локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/` (рис. 4.9).

```
annikolenko@dk8n60 ~/work/arch-pc/lab04 $ mv hello.asm lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
```

Рис. 4.9: Копирование файлов в каталог локального репозитория

```

annikolenko@dk8n60: ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
annikolenko@dk8n60: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git add .
annikolenko@dk8n60: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -m 'Add existng file'
[master ca38f77] Add existng file
18 files changed, 49 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/lab4.1.png
create mode 100644 labs/lab04/report/image/lab4.10.png
create mode 100644 labs/lab04/report/image/lab4.11.png
create mode 100644 labs/lab04/report/image/lab4.12.png
create mode 100644 labs/lab04/report/image/lab4.13.png
create mode 100644 labs/lab04/report/image/lab4.2.png
create mode 100644 labs/lab04/report/image/lab4.3.png
create mode 100644 labs/lab04/report/image/lab4.4.png
create mode 100644 labs/lab04/report/image/lab4.5.png
create mode 100644 labs/lab04/report/image/lab4.6.png
create mode 100644 labs/lab04/report/image/lab4.7.png
create mode 100644 labs/lab04/report/image/lab4.8.png
create mode 100644 labs/lab04/report/image/lab4.9.png
create mode 100644 labs/lab04/report/list.lst
create mode 100644 labs/lab04/report/report.docx
create mode 100644 labs/lab04/report/report.pdf
annikolenko@dk8n60: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 29, готово.
Подсчет объектов: 100% (29/29), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (24/24), готово.
Запись объектов: 100% (24/24), 758.99 Киб | 22.32 Миб/с, готово.
Всего 24 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:KIKO-shi/study_2023-2024_arh-pc.git
d0dc8d2..ca38f77 master -> master

```

Загружаю файлы на Github (рис. ??).

#Листинги написанных программ

SECTION .data hello: db "Hello, world!",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

## 5 Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.