

# **Отчёт по лабораторной работе №4**

**Дисциплина: Архитектура компьютера**

Бережной Иван Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Написание программы Hello world! . . . . .	7
3.2	Работа с транслятором NASM . . . . .	8
3.3	Работа с компоновщиком LD . . . . .	9
3.4	Запуск исполняемого файла . . . . .	10
3.5	Задание для самостоятельной работы . . . . .	10
<b>4</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

# Список иллюстраций

3.1	Создание файлов . . . . .	7
3.2	Редактирование hello.asm . . . . .	8
3.3	Создание объектного файла . . . . .	8
3.4	Создание файлов obj.o и list.lst . . . . .	9
3.5	Компоновка hello.exe . . . . .	9
3.6	Компоновка main.exe . . . . .	9
3.7	Запуск исполняемого файла . . . . .	10
3.8	Создание файла lab4.asm . . . . .	10
3.9	Редактирование lab4.asm . . . . .	11
3.10	Создание объектного файла lab4.o . . . . .	11
3.11	Создание исполняемого файла lab4.exe . . . . .	11
3.12	Запуск исполняемого файла lab4.exe . . . . .	12
3.13	Копирование файлов . . . . .	12
3.14	Загрузка файлов на GitHub . . . . .	13

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

1. Написание программы Hello world!
2. Работа с транслятором NASM
3. Работа с компоновщиком LD
4. Запуск исполняемого файла
5. Задание для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Написание программы Hello world!

Создадим каталог для работы с программами на языке ассемблера NASM командой `mkdir`. Перейдём в него (утилита `cd`) и создадим текстовый файл `hello.asm` с помощью команды `touch` (рис. 3.1).

```
[iaberezhnoy@fedora ~]$ mkdir -p ~/work/arch-pc/lab04  
[iaberezhnoy@fedora ~]$ cd ~/work/arch-pc/lab04  
[iaberezhnoy@fedora lab04]$ touch hello.asm
```

Рис. 3.1: Создание файлов

Откроем новый файл в текстовом редакторе и введём в него предложенный текст (рис. 3.2).

Рис. 3.2: Редактирование hello.asm

## 3.2 Работа с транслятором NASM

Напишем следующую команду в терминале: `nasm -f elf hello.asm` и выведем список файлов в репозитории командой `ls`. Видим новый файл: `hello.o`. Расширение `.o` указывает на то, что файл является объектным (рис. 3.3).

Рис. 3.3: Создание объектного файла

Теперь выполним команду `nasm -o obj.o elf -g -l list.lst hello.asm`. Со-



здались файлы obj.o и list.lst (рис. 3.4).

```
[iaberezhnoy@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[iaberezhnoy@fedora lab04]$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.4: Создание файлов obj.o и list.lst

### 3.3 Работа с компоновщиком LD

Передадим объектный файл на обработку компоновщику командой ld. В репозитории появился исполняемый файл hello.exe (рис. 3.5).

```
[iaberezhnoy@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[iaberezhnoy@fedora lab04]$ ls
binutils-0.2.dev0-py3-none-any.whl  hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.5: Компоновка hello.exe

Введём следующую команду (рис. 3.6). В папке появился ещё один исполняемый файл, но с другим именем, а именно main.exe. Объектный файл, из которого был собран исполняемый, называется obj.o (имена различаются, т.к. мы указали значение main после флага -o).

```
[iaberezhnoy@fedora lab04]$ ld -m elf_i386 obj.o -o main
[iaberezhnoy@fedora lab04]$ ls
binutils-0.2.dev0-py3-none-any.whl  hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 3.6: Компоновка main.exe

### 3.4 Запуск исполняемого файла

Запустим файл командой ./hello (рис. 3.7).

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

Рис. 3.7: Запуск исполняемого файла

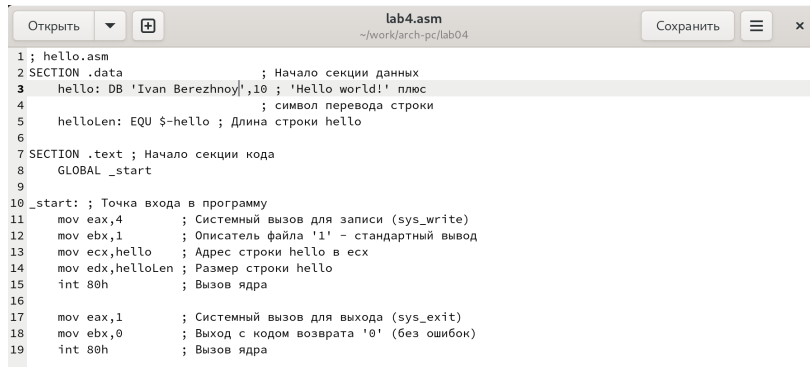
### 3.5 Задание для самостоятельной работы

С помощью команды cp создадим копию файла hello.asm с именем lab4.asm (рис. 3.8).

```
[iaberezhnoy@fedora lab04]$ cp hello.asm lab4.asm
```

Рис. 3.8: Создание файла lab4.asm

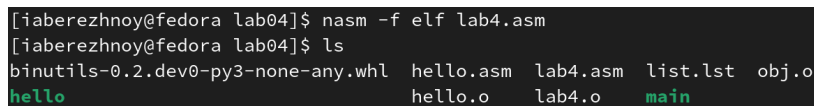
Откроем новый файл в текстовом редакторе и заменим в 3-ей строке текст “Hello world!” на “Ivan Berezhnoy”. Сохраним изменения (рис. 3.9).



```
1; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Ivan Berezchnoy',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6
7 SECTION .text ; Начало секции кода
8 GLOBAL _start
9
10 _start: ; Точка входа в программу
11 mov eax,4 ; Системный вызов для записи (sys_write)
12 mov ebx,1 ; Описатель файла '1' - стандартный вывод
13 mov ecx,hello ; Адрес строки hello в ecx
14 mov edx,helloLen ; Размер строки hello
15 int 80h ; Вызов ядра
16
17 mov eax,1 ; Системный вызов для выхода (sys_exit)
18 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
19 int 80h ; Вызов ядра
```

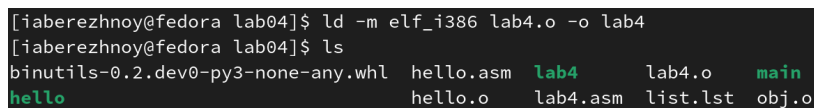
Рис. 3.9: Редактирование lab4.asm

Создадим объектный файл, оттранслировав текст программы lab4.asm - получили объектный файл lab4.o (рис. 3.10). Командой ld скомпануем последний файл - получили исполняемый файл lab4.exe (рис. 3.11). Теперь исполним его с помощью команды ./lab4. В консоль вывелся наш текст “Ivan Berezchnoy” (рис. 3.12).



```
[iaberezhnoy@fedora lab04]$ nasm -f elf lab4.asm
[iaberezhnoy@fedora lab04]$ ls
binutils-0.2.dev0-py3-none-any.whl  hello.asm  lab4.asm  list.lst  obj.o
hello                               hello.o    lab4.o    main
```

Рис. 3.10: Создание объектного файла lab4.o



```
[iaberezhnoy@fedora lab04]$ ld -m elf_i386 lab4.o -o lab4
[iaberezhnoy@fedora lab04]$ ls
binutils-0.2.dev0-py3-none-any.whl  hello.asm  lab4      lab4.o    main
hello                               hello.o    lab4.asm  list.lst  obj.o
```

Рис. 3.11: Создание исполняемого файла lab4.exe

```
[iaberezhnoy@fedora lab04]$ ./lab4
Ivan Berezhnoy
```

Рис. 3.12: Запуск исполняемого файла lab4.exe

Скопируем файлы hello.asm и lab4.asm в репозиторий с лабораторной работой №4 (рис. 3.13).

```
[iaberezhnoy@fedora lab04]$ cp hello.asm /home/iaberezhnoy/work/study/2023-2024/'Архитектура компью  
тера'/arch-pc/labs/lab04  
[iaberezhnoy@fedora lab04]$ cp lab4.asm /home/iaberezhnoy/work/study/2023-2024/'Архитектура компью  
тера'/arch-pc/labs/lab04  
[iaberezhnoy@fedora lab04]$
```

Рис. 3.13: Копирование файлов

Теперь загрузим файлы на GitHub (рис. 3.14).

```

[iaberezhnoy@fedora arch-pc]$ git add .
[iaberezhnoy@fedora arch-pc]$ git commit -m "Lab4 files were added"
[master 8dfae69] Lab4 files were added
17 files changed, 51 insertions(+), 38 deletions(-)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/image1.jpg
create mode 100644 labs/lab04/report/image/image10.jpg
create mode 100644 labs/lab04/report/image/image11.jpg
create mode 100644 labs/lab04/report/image/image12.jpg
create mode 100644 labs/lab04/report/image/image13.jpg
create mode 100644 labs/lab04/report/image/image2.jpg
create mode 100644 labs/lab04/report/image/image3.jpg
create mode 100644 labs/lab04/report/image/image4.jpg
create mode 100644 labs/lab04/report/image/image5.jpg
create mode 100644 labs/lab04/report/image/image6.jpg
create mode 100644 labs/lab04/report/image/image7.jpg
create mode 100644 labs/lab04/report/image/image8.jpg
create mode 100644 labs/lab04/report/image/image9.jpg
delete mode 100644 labs/lab04/report/image/placeimg_800_600_tech.jpg
[iaberezhnoy@fedora arch-pc]$ git push
Перечисление объектов: 28, готово.
Подсчет объектов: 100% (28/28), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (22/22), готово.
Запись объектов: 100% (22/22), 444.03 Киб | 1.71 Миб/с, готово.
Всего 22 (изменений 5), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To github.com:NoisyCake/study_2023-2024_arhpc.git
7701760..8dfae69 master -> master

```

Рис. 3.14: Загрузка файлов на GitHub

## 4 Выводы

В ходе лабораторной работы мы написали простейшие программы на ассемблере NASM, поработали с транслятором NASM и компоновщиком LD и изучили расширенный синтаксис командной строки NASM.

# Список литературы

::: Архитектура ЭВМ