

# **Отчет по Лабораторной работе №4**

**дисциплина: Архитектура компьютера**

Гончарь Анастасия Александровна

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Выполнение лабораторной работы</b>	<b>7</b>
3.1 Создание программы Hello world! . . . . .	7
3.2 Работа с транслятором NASM . . . . .	8
3.3 Работа с расширенным синтаксисом командной строки NASM . . . . .	8
3.4 Работа с компоновщиком LD . . . . .	9
3.5 Запуск исполняемого файла . . . . .	9
3.6 Выполнение задания для самостоятельной работы . . . . .	9
<b>4 Выводы</b>	<b>12</b>

# Список иллюстраций

3.1 Создание каталога . . . . .	7
3.2 Переход в созданный каталог и создание файла hello.asm . . . . .	7
3.3 Открытие файла hello.asm и ввод текста . . . . .	8
3.4 Компиляция текста . . . . .	8
3.5 Компиляция файла hello.asm в obj.o . . . . .	8
3.6 Передача файла на обработку компоновщику и проверка с помощью команды ls . . . . .	9
3.7 Выполнение команды ld -m elf_i386 obj.o -o main . . . . .	9
3.8 Запуск исполняемого файла . . . . .	9
3.9 Создание копии файла . . . . .	10
3.10 Замена “Hello world!” на “Анастасия Гончарь” . . . . .	10
3.11 Компактная и запуск файла . . . . .	10
3.12 Копирование файлов в другой каталог . . . . .	11
3.13 Загрузка файлов на Github . . . . .	11

# **Список таблиц**

# **1 Цель работы**

Целью работы является освоение 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 6.Выполнение задания для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Создание программы Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM (рис. 3.1).



```
aagoncharj@dk8n80 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога

Далее переходим в созданный каталог и создаем в нем текстовый файл hello.asm (рис. 3.2).

```
aagoncharj@dk8n80 ~ $ cd ~/work/arch-pc/lab04  
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис. 3.2: Переход в созданный каталог и создание файла hello.asm

Теперь открываем созданный файл с помощью текстового редактора gedit и вводим в него необходимый текст (рис. 3.3).

The screenshot shows a terminal window with the command `aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ gedit hello.asm`. The file `hello.asm` contains the following assembly code:

```
1 SECTION .data
2     hello:    db "Hello, world!",0xa
3     helloLen: equ $ - hello
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
```

Рис. 3.3: Открытие файла hello.asm и ввод текста

## 3.2 Работа с транслятором NASM

Проводим компиляцию текста программы «Hello World», вставленного в файл `hello.asm` (рис. 3.4).

```
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ touch hello.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ gedit hello.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис. 3.4: Компиляция текста

## 3.3 Работа с расширенным синтаксисом командной строки NASM

Теперь нужно скомпилировать исходный файл `hello.asm` в `obj.o` (рис. 3.5).

```
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ touch hello.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ gedit hello.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 3.5: Компиляция файла hello.asm в obj.o

## 3.4 Работа с компоновщиком LD

Чтобы получить исполняемую программу, объектный файл передаем на обработку компоновщику, и помошью команды ls проверяем, что исполняемый файл hello был создан (рис. 3.6).

```
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 3.6: Передача файла на обработку компоновщику и проверка с помощью команды ls

Делее выполняем следующую команду: ld -m elf\_i386 obj.o -o main (рис. 3.7).

```
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst obj.o
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 3.7: Выполнение команды ld -m elf\_i386 obj.o -o main

## 3.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл, находящийся в текущем каталоге (рис. 3.8).

```
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
aaroncharj@dk8n80 ~/work/arch-pc/lab04 $ ./hello
Hello, world!
```

Рис. 3.8: Запуск исполняемого файла

## 3.6 Выполнение задания для самостоятельной работы

В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаем копию файла hello.asm с именем lab4.asm (рис. 3.9).

```
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
```

Рис. 3.9: Создание копии файла

С помощью текстового редактора gedit вносим изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моей именем и фамилией (рис. 3.10).

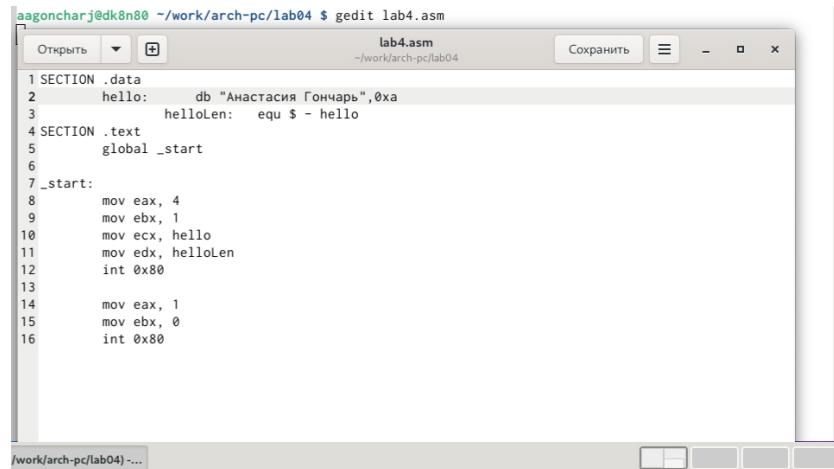


Рис. 3.10: Замена “Hello world!” на “Анастасия Гончарь”

Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рис. 3.11).

```
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
aagoncharj@dk8n80 ~/work/arch-pc/lab04 $ ./lab4
Анастасия Гончар
```

Рис. 3.11: Компактная и запуск файла

Теперь скопируем файлы hello.asm и lab4.asm в каталог ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 3.12).

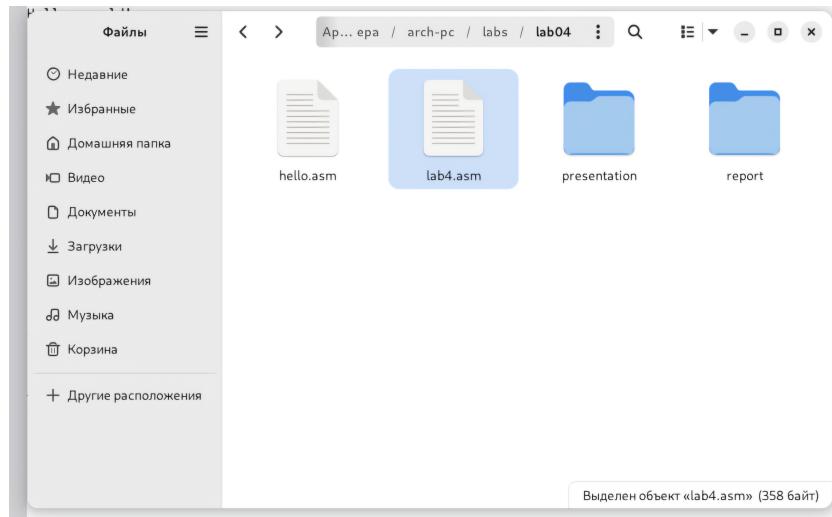


Рис. 3.12: Копирование файлов в другой каталог

Загружаем файлы на Github (рис. 3.13).

```
aagoncharj@dk8n0 ~ $ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04 $ cd ..
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs $ cd ..
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git add .
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git commit 'feat(main): add files lab-4'
error: pathspec 'feat(main): add files lab-4' did not match any file(s) known to git
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git commit -m "Add fales for lab04"
[master e98349a] Add fales for lab04
4 files changed, 32 insertions(+), 1 deletion(-)
 delete mode 100644 labs/lab02/report/_afsA6B8
 delete mode 100644 labs/lab02/report/.lock.Л02_Гончарь_отчет (копия).docx#
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/lab4.asm
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 13, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 824 байта | 824.00 КиБ/с, готово.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:aganastasii/study_2024-2025_arh-pc.git
 6ad0cba..e98349a master -> master
aagoncharj@dk8n0 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $
```

Рис. 3.13: Загрузка файлов на Github

## **4 Выводы**

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