

# **Отчет по лабораторной работе №4**

Иванова Анастасия Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>9</b>
<b>4</b>	<b>Вывод</b>	<b>11</b>

# Список иллюстраций

2.1	Создание каталога и переход в него . . . . .	6
2.2	Создание текстового файла и изменение в текстовом редакторе . . . . .	6
2.3	Ввод текста . . . . .	7
2.4	Превращение текста в объектный код . . . . .	7
2.5	Компиляция исходного файла . . . . .	7
2.6	Передача на обработку компоновщику . . . . .	8
2.7	Переход в каталог курса . . . . .	8
2.8	Проверка содержимого папки . . . . .	8
2.9	Запуск исполняемого файла . . . . .	8
3.1	Копирование файла hello.asm . . . . .	9
3.2	Внесение изменений в файл . . . . .	9
3.3	Выполнение команд . . . . .	10
3.4	Копирование файлов в репозиторий . . . . .	10
3.5	Загрузка файлов на Github . . . . .	10

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

### 1. Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран. Создадим каталог для работы с программами на языке ассемблера NASM и перейдем в созданный каталог [рис.@fig-001]):

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

Рисунок 2.1: Создание каталога и переход в него

Создадим текстовый файл с именем hello.asm и откроем этот файл с помощью текстового редактора gedit[рис.@fig-002]):

```
asivanova@anastasia:~/work/arch-pc/lab04$ touch hello.asm
asivanova@anastasia:~/work/arch-pc/lab04$ gedit hello.asm
```

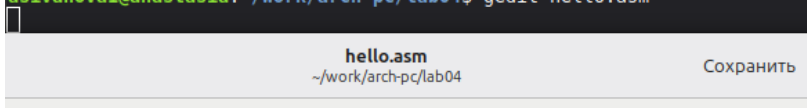
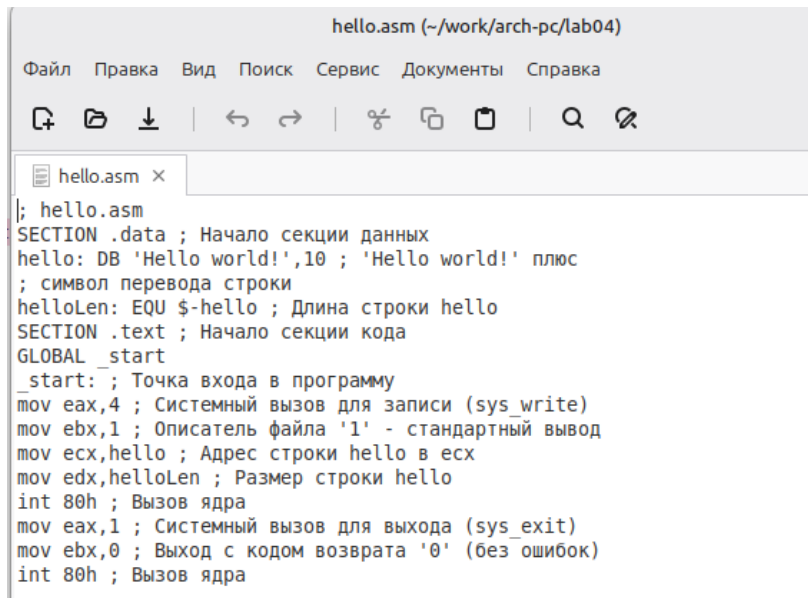


Рисунок 2.2: Создание текстового файла и изменение в текстовом редакторе

Введем в него необходимый текст рис. 2.3):

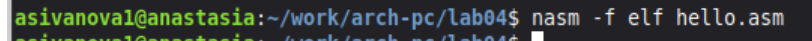


```
hello.asm (~work/arch-pc/lab04)
Файл  Правка  Вид  Поиск  Сервис  Документы  Справка
[Иконки]
hello.asm x
; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Hello world!',10 ; 'Hello world!' плюс
; символ перевода строки
helloLen: EQU $-hello ; Длина строки hello
SECTION .text ; Начало секции кода
GLOBAL _start
_start: ; Точка входа в программу
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,hello ; Адрес строки hello в ecx
mov edx,helloLen ; Размер строки hello
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h ; Вызов ядра
```

Рисунок 2.3: Ввод текста

## 2.Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции приведённого выше текста программы «Hello World» напомним рис. 2.4):



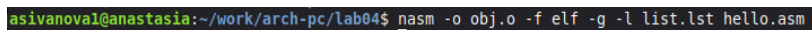
```
asivanova1@anastasia:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

Рисунок 2.4: Превращение текста в объектный код

Транслятор преобразовал текст программы из файла hello.asm в объектный код, который записался в файл hello.o.

## 3.Расширенный синтаксис командной строки NASM

Выполним следующую команду рис. 2.5):



```
asivanova1@anastasia:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рисунок 2.5: Компиляция исходного файла

Данная команда скомпилировала исходный файл hello.asm в obj.o, при этом формат выходного файла стал elf, в который включены символы для отладки.

## 4.Компоновщик LD

Чтобы получить исполняемую программу, объектный файл, передадим на обработку компоновщику рис. 2.6):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
```

Рисунок 2.6: Передача на обработку компоновщику

Ключ -o с последующим значением задал в данном имя создаваемого исполняемого файла.

Выполним следующую команду рис. 2.7):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
```

Рисунок 2.7: Переход в каталог курса

Проверка содержимого папки, после выполнения всех команд рис. 2.8):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
```

Рисунок 2.8: Проверка содержимого папки

## 5. Запуск исполняемого файла

Запустим созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке ./hello рис. 2.9):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рисунок 2.9: Запуск исполняемого файла



### 3 Задание для самостоятельной работы

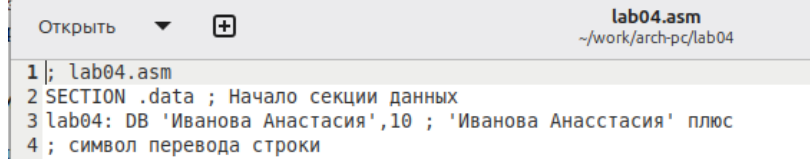
1. В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm` (рис. 3.1):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ cp ~/work/arch-pc/lab04/hello.asm ~/work/arch-pc/lab04/lab4.asm
asivanova1@anastasia:~/work/arch-pc/lab04$ ls -la ~/work/arch-pc/lab04/
итого 52
drwxrwxr-x 2 asivanova1 asivanova1 4096 окт 26 19:32 .
drwxrwxr-x 3 asivanova1 asivanova1 4096 окт 26 13:42 ..
-rwxrwxr-x 1 asivanova1 asivanova1 8668 окт 26 13:49 hello
-rw-rw-r-- 1 asivanova1 asivanova1 812 окт 26 13:46 hello.asm
-rw-rw-r-- 1 asivanova1 asivanova1 656 окт 26 13:46 hello.o
-rw-rw-r-- 1 asivanova1 asivanova1 812 окт 26 19:32 lab4.asm
-rw-rw-r-- 1 asivanova1 asivanova1 1488 окт 26 13:48 list.lst
-rwxrwxr-x 1 asivanova1 asivanova1 9096 окт 26 13:50 main
-rw-rw-r-- 1 asivanova1 asivanova1 1552 окт 26 13:48 obj.o
```

Рисунок 3.1: Копирование файла `hello.asm`

2. С помощью текстового редактора внесем изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводились фамилия и имя (рис. 3.2):

```
asivanova1@anastasia:~/work/arch-pc/lab04$ gedit lab4.asm
```



```
1; lab04.asm
2 SECTION .data ; Начало секции данных
3 lab04: DB 'Иванова Анастасия',10 ; 'Иванова Анастасия' плюс
4 ; символ перевода строки
```

Рисунок 3.2: Внесение изменений в файл

3. Оттранслируем полученный текст программы `lab4.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рис. 3.3):

```

asivanova@anastasia:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
asivanova@anastasia:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
asivanova@anastasia:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
asivanova@anastasia:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
asivanova@anastasia:~/work/arch-pc/lab04$ ./lab4
Иванова Анастасия

```

Рисунок 3.3: Выполнение команд

4. Скопируем файлы hello.asm и lab4.asm в Наш локальный репозиторий в каталог ~/work/study/2025-2026/«Архитектура компьютера»/arch-pc/labs/lab04/ и загрузим файлы на Github рис. 3.4):

```

asivanova@anastasia:~/work/arch-pc/lab04$ mv ~/work/arch-pc/lab04/lab4.asm ~/work/study_2025-2026_arh-pc/Архитектура компьютера/arch-pc/labs/lab04/
asivanova@anastasia:~/work/arch-pc/lab04$ mv ~/work/arch-pc/lab04/hello.asm ~/work/study_2025-2026_arh-pc/Архитектура компьютера/arch-pc/labs/lab04/

```

Рисунок 3.4: Копирование файлов в репозиторий

рис. 3.5):

```

asivanova@anastasia:~/work/study_2025-2026_arh-pc/Архитектура компьютера/arch-pc/labs/lab04$ git push
Перечисление объектов: 31, готово.
Подсчет объектов: 100% (31/31), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (19/19), готово.
Запись объектов: 100% (30/30), 699.43 Киб | 3.59 Миб/с, готово.
Всего 30 (изменений 1), повторно использовано 18 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:asivanova/study_2025-2026_arh-pc.git
 ad0e655..95c61cd master -> master

```

Рисунок 3.5: Загрузка файлов на Github

## 4 Вывод

Мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM.