

# **Отчет по лабораторной работе №4**

**Дисциплина: архитектура компьютера**

Учаева Алёна Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Создание программы Hello world! . . . . .	9
4.2	Работа с транслятором NASM . . . . .	11
4.3	Работа с расширенным синтаксисом командной строки NASM . . .	11
4.4	Работа с компоновщиком LD . . . . .	12
4.5	Запуск исполняемого файла . . . . .	13
4.6	Выполнение заданий для самостоятельной работы . . . . .	13
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	Создание пустого файла . . . . .	9
4.2	Открытие файла в текстовом редакторе . . . . .	10
4.3	Заполнение файла . . . . .	10
4.4	Компиляция текста программы . . . . .	11
4.5	Компиляция текста программы . . . . .	12
4.6	Передача объектного файла компоновщику . . . . .	12
4.7	Передача объектного файла компоновщику . . . . .	12
4.8	Запуск исполняемого файла . . . . .	13
4.9	Создание копии файла . . . . .	13
4.10	Изменение программы . . . . .	13
4.11	Компиляция текста программы . . . . .	14
4.12	Передача файла компоновщику . . . . .	14
4.13	Запуск исполняемого файла . . . . .	14

# 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 заданий для самостоятельной работы

### 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. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, пре-

образование (арифметические или логические операции) данных хранящихся в регистрах. Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита. В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего используются при написании программ):

- RAX, RCX, RDX, RBX, RSI, RDI — 64-битные
- EAX, ECX, EDX, EBX, ESI, EDI — 32-битные
- AX, CX, DX, BX, SI, DI — 16-битные
- AH, AL, CH, CL, DH, DL, BH, BL — 8-битные (половинки 16-битных регистров).

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных. В состав ЭВМ также входят периферийные устройства, которые можно разделить на:

- устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных (жёсткие диски, твердотельные накопители, магнитные ленты);
- устройства ввода-вывода, которые обеспечивают взаимодействие ЦП с внешней средой.

В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы. Программа состоит из машинных команд, которые указывают, какие операции и над какими данными (или операндами), в какой последовательности необходимо выполнить. Набор машинных команд определяется устройством конкретного процессора. Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса

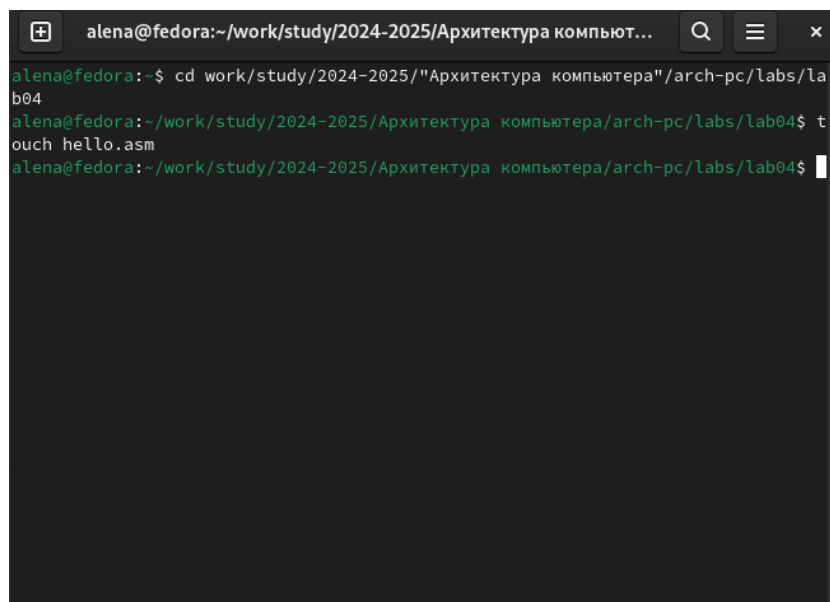
данных, которые участвуют в выполнении данной операции. При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. В самом общем виде он заключается в следующем: 1. формирование адреса в памяти очередной команды; 2. считывание кода команды из памяти и её дешифрация; 3. выполнение команды; 4. переход к следующей команде. Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.



## 4 Выполнение лабораторной работы

### 4.1 Создание программы Hello world!

Перемещаюсь в каталог, в котором буду работать и создаю пустой текстовый файл `hello.asm` с помощью команды `touch` (рис. 4.1).

A screenshot of a terminal window with a dark background. The window title is "alena@fedora:~/work/study/2024-2025/Архитектура компьют...". The terminal shows the following commands and output:

```
alena@fedora:~$ cd work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ touch hello.asm
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.1: Создание пустого файла

Открываю созданный файл в текстовом редакторе (рис. 4.2).

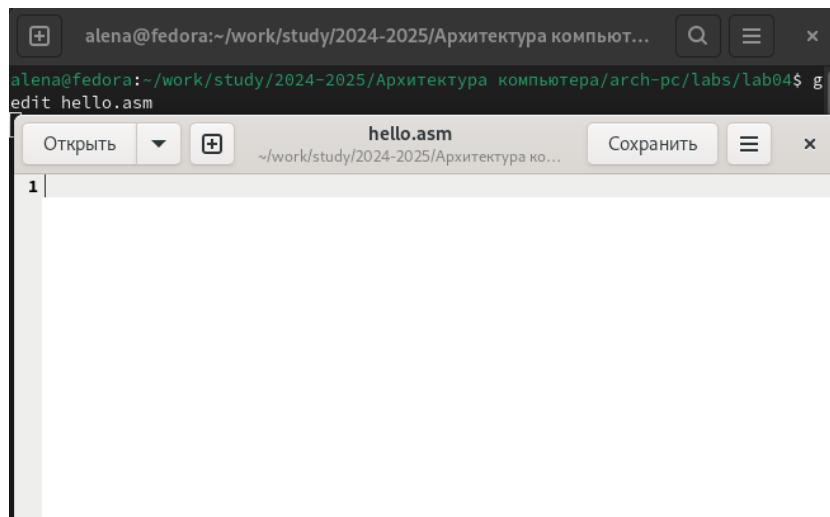


Рис. 4.2: Открытие файла в текстовом редакторе

Заполняю файл вставляя в него программу для вывода Hello world! (рис. 4.3).

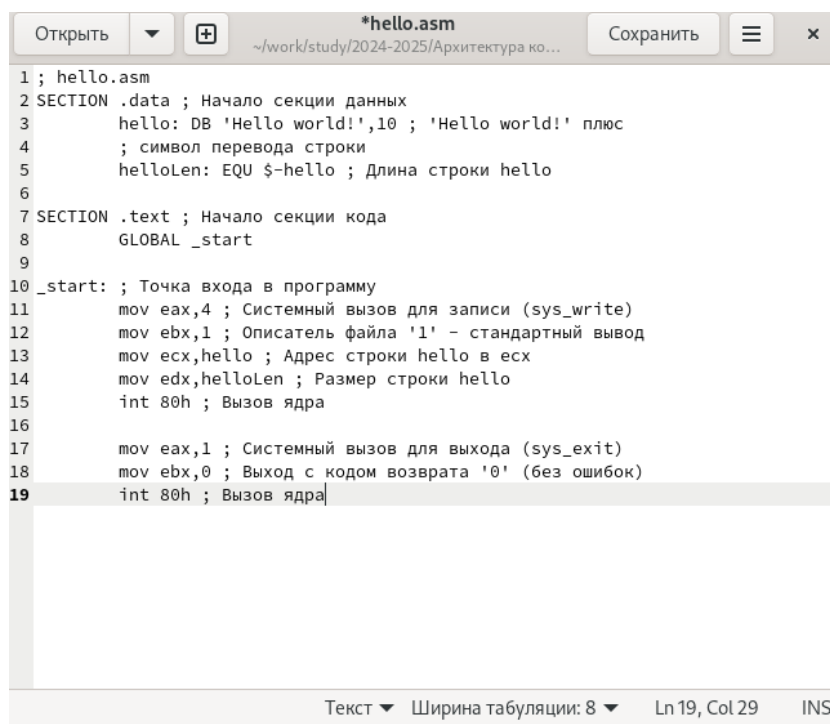
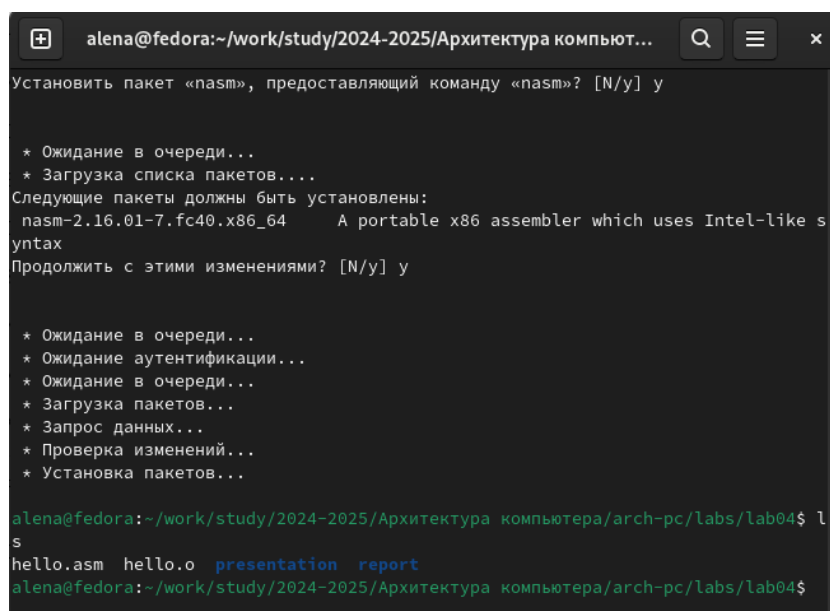


Рис. 4.3: Заполнение файла

## 4.2 Работа с транслятором NASM

Превращаю текст программы “Hello world!” в объектный код с помощью транслятора NASM и с помощью команды `ls` проверяю, что объектный файл был создан (рис. 4.4).



```
alena@fedora:~/work/study/2024-2025/Архитектура компьют...
Установить пакет «nasm», предоставляющий команду «nasm»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
nasm-2.16.01-7.fc40.x86_64      A portable x86 assembler which uses Intel-like s
yntax
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ l
s
hello.asm hello.o presentation report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.4: Компиляция текста программы

## 4.3 Работа с расширенным синтаксисом командной строки NASM

Далее ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o`, при этом в файл будут включены символы для отладки и будет создан файл листинга `list.lst`, затем проверяю корректность выполненных действий (рис. 4.5).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера...

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
nasm-2.16.01-7.fc40.x86_64    A portable x86 assembler which uses Intel-like synt
ax
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm hello.o presentation report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ nasm
-o obj.o -f elf -g -l list.lst hello.asm
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm hello.o list.lst obj.o presentation report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.5: Компиляция текста программы

## 4.4 Работа с компоновщиком LD

Передаю объектный файл на обработку компоновщику LD, чтобы получить исполняемый файл hello и проверяю корректность, с помощью команды ls (рис. 4.6).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_
i386 hello.o -o hello
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst obj.o presentation report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.6: Передача объектного файла компоновщику

Выполняю команду (рис. 4.7). Исполняемый файл будет иметь имя main, так как после ключа -o было задано значение main. Объектный файл из которого собран этот исполняемый файл, имеет имя obj.o.

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_
i386 obj.o -o main
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst main obj.o presentation report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.7: Передача объектного файла компоновщику

## 4.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 4.8).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ./hello
Hello world!
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.8: Запуск исполняемого файла

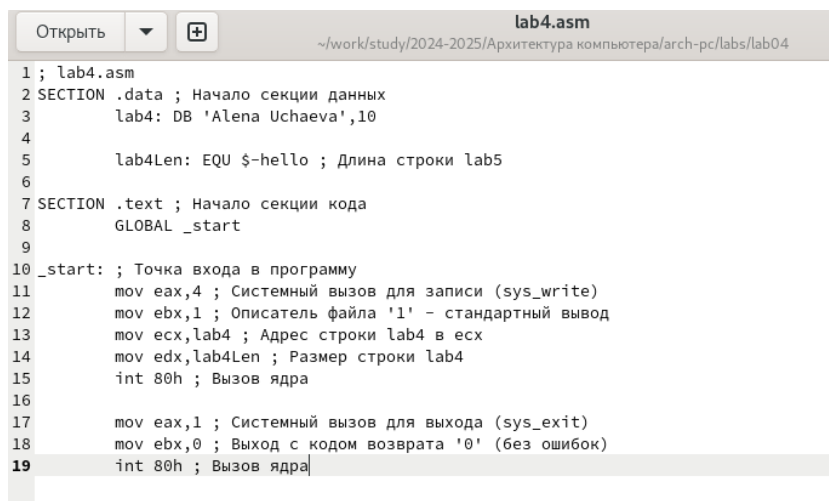
## 4.6 Выполнение заданий для самостоятельной работы

С помощью команды `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab4.asm` (рис. 4.9).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ cp hello.asm lab4.asm
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.9: Создание копии файла

С помощью текстового редактора открываю файл `lab4.asm` и вношу изменения в программу, чтобы она выводила мои имя и фамилию (рис. 4.10).



```
lab4.asm
1 ; lab4.asm
2 SECTION .data ; Начало секции данных
3     lab4: DB 'Alena Uchaeva',10
4
5     lab4Len: EQU $-hello ; Длина строки lab5
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,lab4 ; Адрес строки lab4 в ecx
14     mov edx,lab4Len ; Размер строки lab4
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. 4.10: Изменение программы

Компилирую текст программы в объектный файл и проверяю корректность действий (рис. 4.11).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютер...
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ n
asm -f elf lab4.asm
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ l
s
hello      hello.o  lab4.o  main  presentation
hello.asm  lab4.asm  list.lst  obj.o  report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.11: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. 4.12).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютер...
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ n
asm -f elf lab4.asm
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ l
s
hello      hello.o  lab4.o  main  presentation
hello.asm  lab4.asm  list.lst  obj.o  report
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ l
d -m elf_i386 lab4.o -o lab4
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ l
s
hello      hello.o  lab4.asm  list.lst  obj.o      report
hello.asm  lab4     lab4.o    main      presentation
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.12: Передача файла компоновщику

Запускаю исполняемый файл lab4, на экран выводятся мои имя и фамилия (рис. 4.13).

```
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ./lab4
Alena Uchaeva
alena@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.13: Запуск исполняемого файла

## **5 Выводы**

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

# **Список литературы**

1.Архитектура ЭВМ