

# **Отчёт по лабораторной работе №4**

**Дисциплина: Архитектура компьютера**

Савостин Олег

# 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>9</b>
4.1	Изучение программы Hello world! . . . . .	9
4.2	Транслятор NASM . . . . .	11
4.3	Расширенный синтаксис командной строки NASM . . . . .	12
4.4	Компоновщик LD . . . . .	12
4.5	Запуск исполняемого файла . . . . .	13
4.6	Выполнение заданий самостоятельной работы . . . . .	13
<b>5</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

# Список иллюстраций

4.1	Переход в нужную папку с помощью <code>cd</code> . . . . .	9
4.2	Создание нового файла “hello.asm” . . . . .	9
4.3	Открытый ранее созданный мною файл. . . . .	10
4.4	Заполненный файл hello.asm . . . . .	11
4.5	Компиляция файла и проверка на успех совершения команды. . .	12
4.6	Компиляция файла в <code>obj.o</code> , формирование файла <code>list.lst</code> и проверка на успех совершения команды. . . . .	12
4.7	Обработка файла и проверка на наличие исполняемого файла. . .	12
4.8	Выполнение команды и проверка на наличие файла <code>main</code> . . . . .	13
4.9	Выполнение созданного исполняемого файла . . . . .	13
4.10	Копирование файла с именем <code>lab4.asm</code> . . . . .	13
4.11	Изменение текста в <code>lab4.asm</code> . . . . .	14
4.12	Трансляция из <code>asm</code> . в <code>.o</code> . . . . .	14
4.13	Вывод моих фамилии и имени в терминале. . . . .	14
4.14	Отправка файлов на Github . . . . .	15

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Задание

1. Изучение программы Hello world!
2. Транслятор NASM
3. Расширенный синтаксис командной строки NASM
4. компоновщик LD
5. Запуск исполняемого файла
6. Выполнение заданий самостоятельной работы

### 3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины(ЭВМ) являются центральный процессор, память и периферийные устройства. 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; 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. Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным

возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр.



## 4 Выполнение лабораторной работы

### 4.1 Изучение программы Hello world!

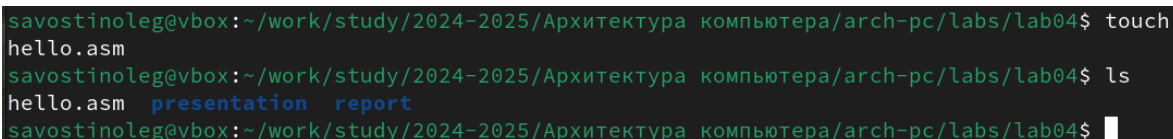
Для начала выполнения лабораторной работы, я перехожу в каталог “lab04” с помощью утилиты “cd”(рис. 4.1).



```
savostinoleg@vbox:~$ cd work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.1: Переход в нужную папку с помощью cd

Затем я создаю новый файл под названием “hello.asm”(рис. 4.2) и открываю новый файл с помощью gedit(рис. 4.3)



```
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ touch
hello.asm
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.2: Создание нового файла “hello.asm”

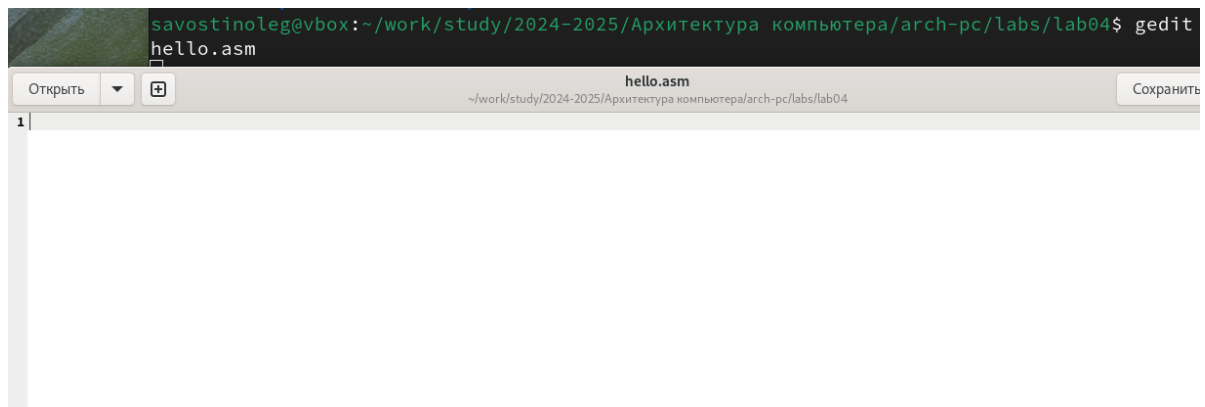
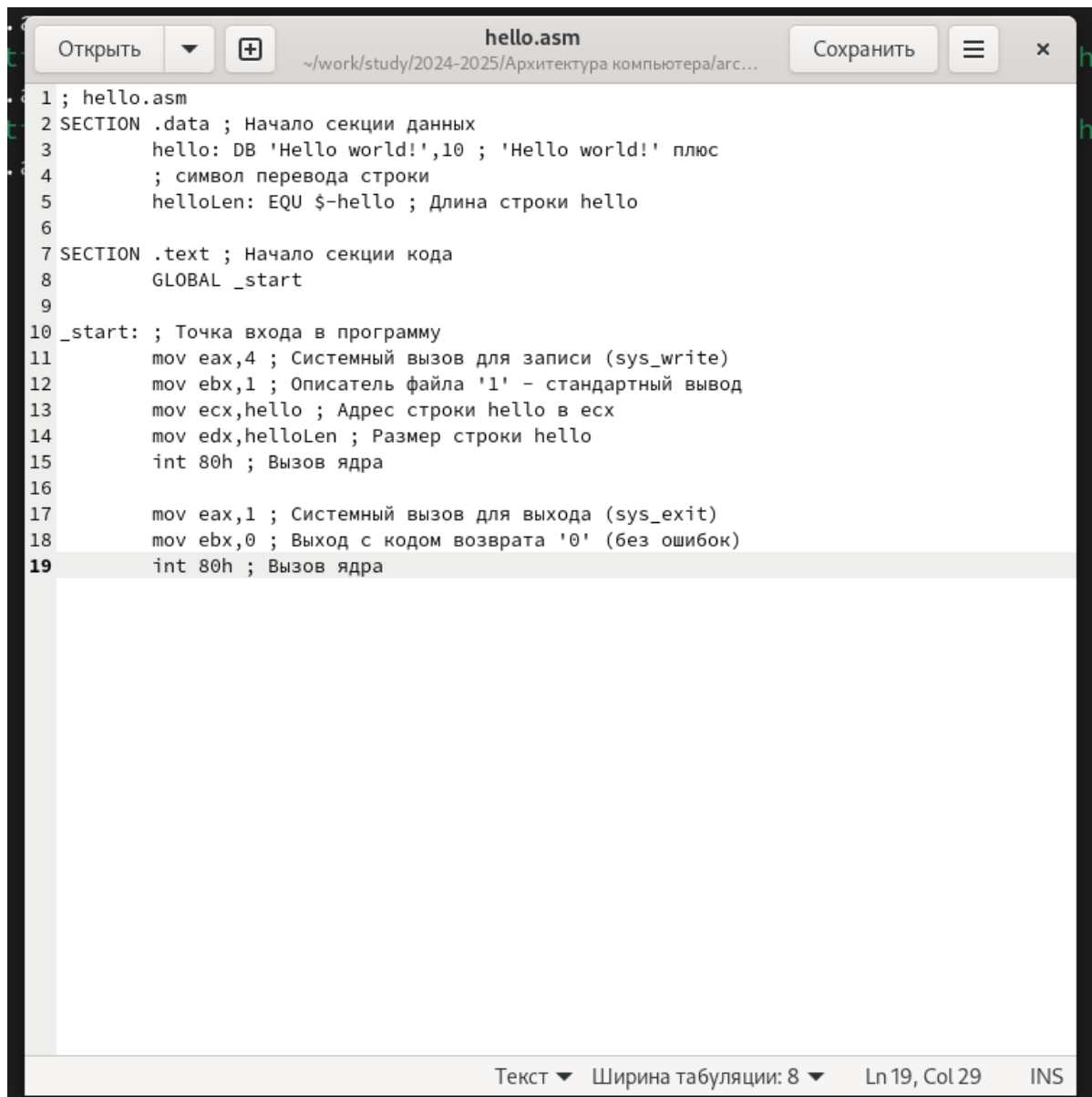


Рис. 4.3: Открытый ранее созданный мною файл.

Теперь, я ввожу текст в данный файл, который был предоставлен в файле лабораторной работы (4.4)



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Hello world!',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. 4.4: Заполненный файл hello.asm

## 4.2 Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции текста в файле “hello.asm” я ввожу команду “nasm -f elf hello.asm” (4.5)

```
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf hello.asm
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm hello.o presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.5: Компиляция файла и проверка на успех совершения команды.

Компиляция прошла успешно и как видно на изображении, появился файл hello.o

### 4.3 Расширенный синтаксис командной строки NASM

Теперь я скомпилирую исходный файл hello.asm в obj.o и кроме того, будет создан файл листинга list.lst. Затем проверяю на правильность выполнения команды.(4.6)

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

Рис. 4.6: Компиляция файла в obj.o, формирование файла list.lst и проверка на успех совершения команды.

### 4.4 Компоновщик LD

Теперь, я передаю объектный файл на обработку компоновщику с помощью команды ld -m elf\_i386 hello.o -o hello и проверяю если исполняемый файл hello был создан.(4.7)

```
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst obj.o presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.7: Обработка файла и проверка на наличие исполняемого файла.

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла и чтобы проверить это я ввожу команду: ld -m elf\_i386 obj.o -o main , после которой появляется файл main.(4.8)

```
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 obj.o -o
main
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst main obj.o presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.8: Выполнение команды и проверка на наличие файла main.

## 4.5 Запуск исполняемого файла

И теперь я запускаю на выполнение файл, вводя команду в командной строке: `./hello`, что выводит на экран Hello world!(4.9)

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

Рис. 4.9: Выполнение созданного исполняемого файла

## 4.6 Выполнение заданий самостоятельной работы

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаю копию файла `hello.asm` с именем `lab4.asm` и проверяю на его наличие(4.10)

```
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ cp hello.asm lab4.asm
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 4.10: Копирование файла с именем lab4.asm

2. С помощью программного текстового редактора вношу изменения в текст программы в файле `lab4.asm` так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем(4.11)

```

|; lab4.asm
SECTION .data
    lab4: DB 'Савостин Олег',10

    lab4len: EQU $-lab4

SECTION .text
GLOBAL _start

_start:
    mov eax,4
    mov ecx,lab4
    mov edx,lab4len
    int 80h

    mov eax,1
    mov ebx,0
    int 80h

```

Рис. 4.11: Изменение текста в lab4.asm

3. Транслирую полученный текст программы lab4.asm в объектный файл. Затем выполняю компоновку объектного файла и запускаю получившийся исполняемый файл(4.12)

```

savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf lab4.asm
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o presentation report

```

Рис. 4.12: Трансляция из asm. в .o

```

savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 lab4.o -o lab4
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 obj.o -o main
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o presentation report
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ ./lab4
Савостин Олег

```

Рис. 4.13: Вывод моих фамилии и имени в терминале.

4. Так как файлы hello.asm и lab4.asm уже находятся в нужном репозитории, я их только отправляю на Github.(4.14)

```

savostinoleg@vbox:~$ cd work/study/2024-2025/"Архитектура компьютера"/arch-pc
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m "Added files for lab04"
[master 3952f9f] Added files for lab04
10 files changed, 57 insertions(+)
create mode 100644 labs/lab03/report.zip
create mode 100755 labs/lab04/hello
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/hello.o
create mode 100755 labs/lab04/lab4
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/lab4.o
create mode 100644 labs/lab04/list.lst
create mode 100755 labs/lab04/main
create mode 100644 labs/lab04/obj.o
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 100% (19/19), готово.
Подсчет объектов: 100% (19/19), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (15/15), готово.
Запись объектов: 100% (15/15), 6.84 МиБ | 290.00 КиБ/с, готово.
Total 15 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:osavostin/study_2023-2024_arh-pc.git
   15e628e..3952f9f  master -> master
savostinoleg@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 4.14: Отправка файлов на Github

## **5 Выводы**

В ходе данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.



# **Список литературы**

1. Архитектура ЭВМ - РУДН