

Шаблон отчёта по лабораторной работе №5

Создание и процесс обработки программ на языке ассемблера NASM

Татьяна Алексеевна Коннова

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы : Программа Hello world!	6
4	Транслятор NASM	8
5	Расширенный синтаксис командной строки NASM	9
6	Компоновщик LD	10
7	Запуск исполняемого файла	12
8	Задание для самостоятельной работы	13
9	Выводы	16

Список иллюстраций

3.1	Создание каталога	6
3.2	Переход в папку lab05	6
3.3	Создание и открытие файла	7
3.4	Текст с последующим выводом необходимого	7
4.1	Компиляция	8
5.1	Компиляция	9
5.2	Проверка создания файлов	9
6.1	Компоновка	10
6.2	Проверка создания файлов	10
6.3	Команда ld	11
7.1	Чтение файла	12
8.1	Копирование	13
8.2	Замена вывода на Фамилию и Имя	14
8.3	Трансляция в объектный файл, компоновка	14
8.4	Переносим, коммитим	15

1 Цель работы

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

2 Задание

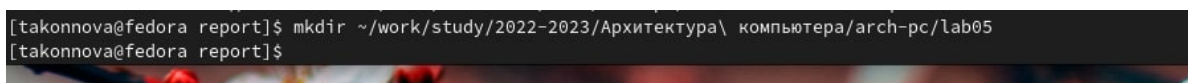
Вывести “Hello world!” и выполнить прочие действия в терминале виртуальной машины, используя ассемблер NASM.

3 Выполнение лабораторной работы :

Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран. Создаём каталог для работы с программами на языке ассемблера NASM:

`mkdir ~/.../lab05` (рис. 3.1)

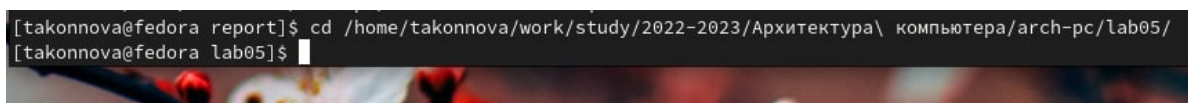


```
[takonnova@fedora report]$ mkdir ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab05
[takonnova@fedora report]$
```

Рис. 3.1: Создание каталога

Перейдём в созданный каталог

`cd ~/work/arch-pc/lab05` (рис. 3.2)



```
[takonnova@fedora report]$ cd /home/takonnova/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab05/
[takonnova@fedora lab05]$
```

Рис. 3.2: Переход в папку lab05

Создаём текстовый файл с именем hello.asm

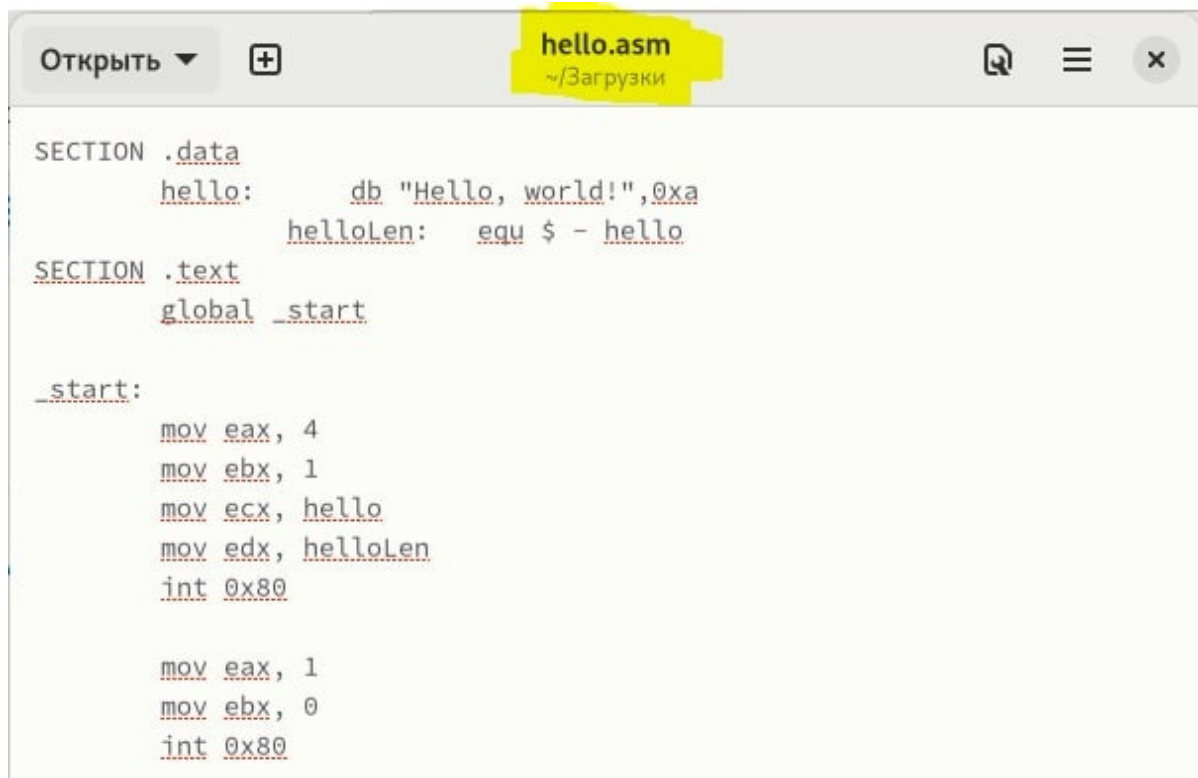
`touch hello.asm` (рис. 3.3)

откроем этот файл с помощью любого текстового редактора, например, gedit
`gedit hello.asm` (рис. 3.3)

```
[takonnova@fedora lab05]$ touch hello.asm
[takonnova@fedora lab05]$ gedit hello.asm
```

Рис. 3.3: Создание и открытие файла

и введём в него следующий текст(скриншот взят из файла шаблона данного текста, расположенного на платформе ТУИС). Скопируем и вставим его в настоящий файл: (рис. 3.4)



```
SECTION .data
    hello:      db "Hello, world!",0xa
               helloLen: equ $ - hello
SECTION .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, helloLen
    int 0x80

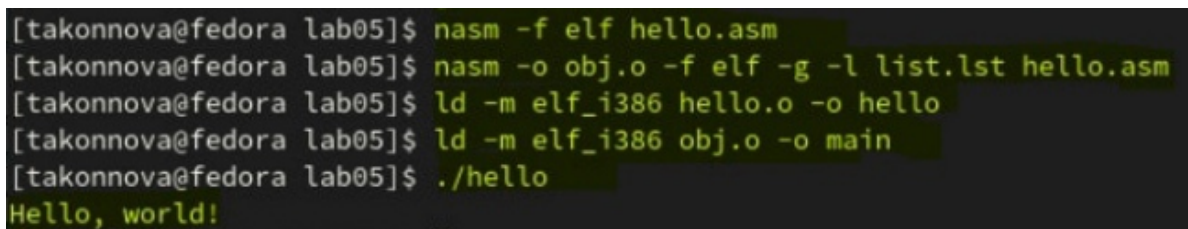
    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 3.4: Текст с последующим выводом необходимого

В отличие от многих современных высокоуровневых языков программирования, в ассемблерной программе каждая команда располагается на отдельной строке. Размещение нескольких команд на одной строке недопустимо. Синтаксис ассемблера NASM является чувствительным к регистру, т.е. есть разница между большими и малыми буквами.

4 Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:
nasm -f elf hello.asm (рис. 4.1)



```
[takonnova@fedora lab05]$ nasm -f elf hello.asm
[takonnova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[takonnova@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[takonnova@fedora lab05]$ ld -m elf_i386 obj.o -o main
[takonnova@fedora lab05]$ ./hello
Hello, world!
```

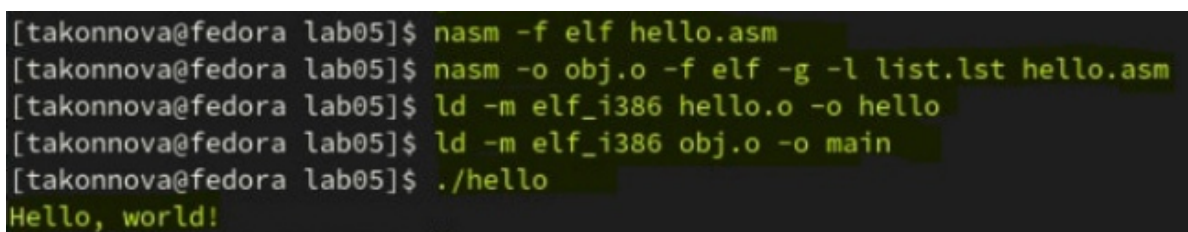
Рис. 4.1: Компиляция

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который запишется в файл hello.o.

5 Расширенный синтаксис командной строки NASM

Выполняем следующую команду:

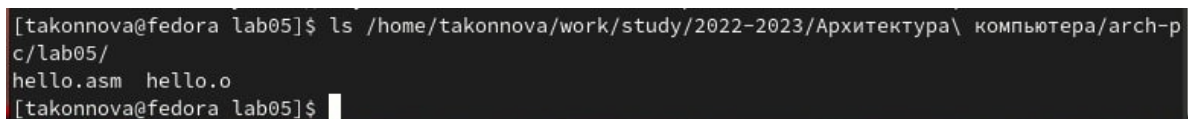
`nasm -o obj.o -f elf -g -l list.lst hello.asm` (рис. 5.1)



```
[takonnova@fedora lab05]$ nasm -f elf hello.asm
[takonnova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[takonnova@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[takonnova@fedora lab05]$ ld -m elf_i386 obj.o -o main
[takonnova@fedora lab05]$ ./hello
Hello, world!
```

Рис. 5.1: Компиляция

Компилирует исходный файл `hello.asm` в `obj.o`. С помощью команды `ls` проверяем, что файлы были созданы. (рис. 5.2)



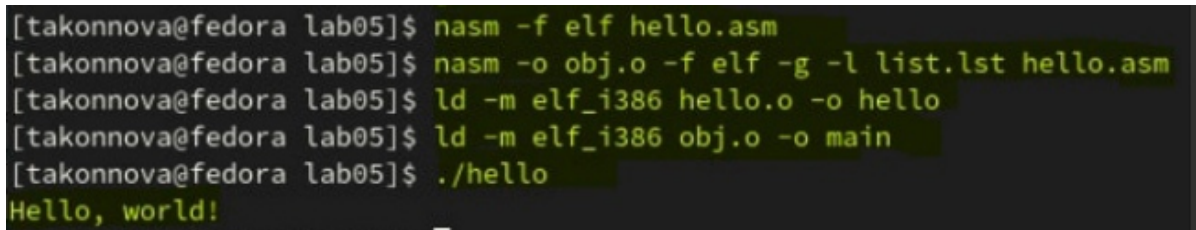
```
[takonnova@fedora lab05]$ ls /home/takonnova/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab05/
hello.asm hello.o
[takonnova@fedora lab05]$
```

Рис. 5.2: Проверка создания файлов

6 Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

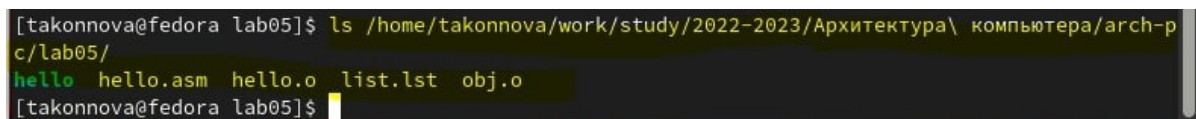
`ld -m elf_i386 hello.o -o hello` (рис. 6.1)



```
[takonnova@fedora lab05]$ nasm -f elf hello.asm
[takonnova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[takonnova@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[takonnova@fedora lab05]$ ld -m elf_i386 obj.o -o main
[takonnova@fedora lab05]$ ./hello
Hello, world!
```

Рис. 6.1: Компоновка

С помощью команды `ls` проверяем, что исполняемый файл `hello` был создан.(рис. 6.2)



```
[takonnova@fedora lab05]$ ls /home/takonnova/work/study/2022-2023/Архитектура\ компьютера/arch-p
c/lab05/
hello hello.asm hello.o list.lst obj.o
[takonnova@fedora lab05]$
```

Рис. 6.2: Проверка создания файлов

Выполняем следующую команду:

`ld -m elf_i386 obj.o -o main` (рис. 6.3)

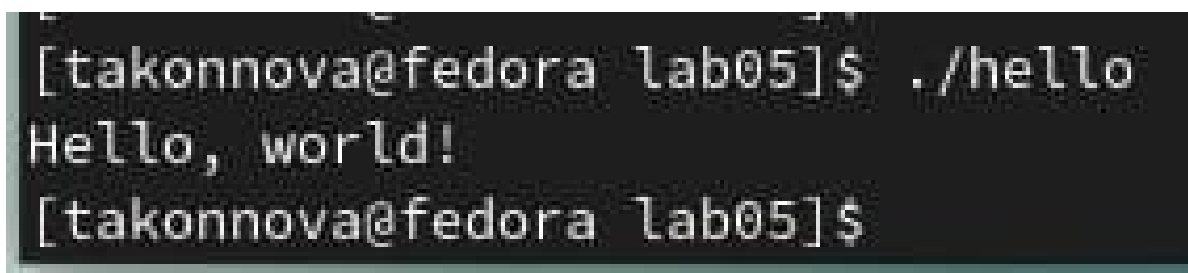
```
[takonnova@fedora lab05]$ nasm -f elf hello.asm  
[takonnova@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm  
[takonnova@fedora lab05]$ ld -m elf_i386 hello.o -o hello  
[takonnova@fedora lab05]$ ld -m elf_i386 obj.o -o main  
[takonnova@fedora lab05]$ ./hello  
Hello, world!
```

Рис. 6.3: Команда ld

7 Запуск исполняемого файла

Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге:

`./hello` (рис. 7.1)

A terminal window with a dark background and light gray text. The prompt is [takonnova@fedora lab05]\$ and the command ./hello has been entered. The output is Hello, world! followed by a new line and the prompt [takonnova@fedora lab05]\$ again.

```
[takonnova@fedora lab05]$ ./hello
Hello, world!
[takonnova@fedora lab05]$
```

Рис. 7.1: Чтение файла

8 Задание для самостоятельной работы

1. В каталоге ~/work/.../lab05 с помощью команды cp создаем копию файла hello.asm с именем lab5.asm (рис. 8.1)

A terminal window with a dark background and light-colored text. It shows two lines of commands being executed in a directory named lab05 on a Fedora system. The first command is 'cp hello.asm lab5.asm' and the second is 'gedit lab5.asm'. Both commands are preceded by the prompt '[takonnova@fedora lab05]\$'.

Рис. 8.1: Копирование

2. С помощью любого текстового редактора внесём изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с моей фамилией и моим именем. (рис. 8.2) (рис. 8.1)

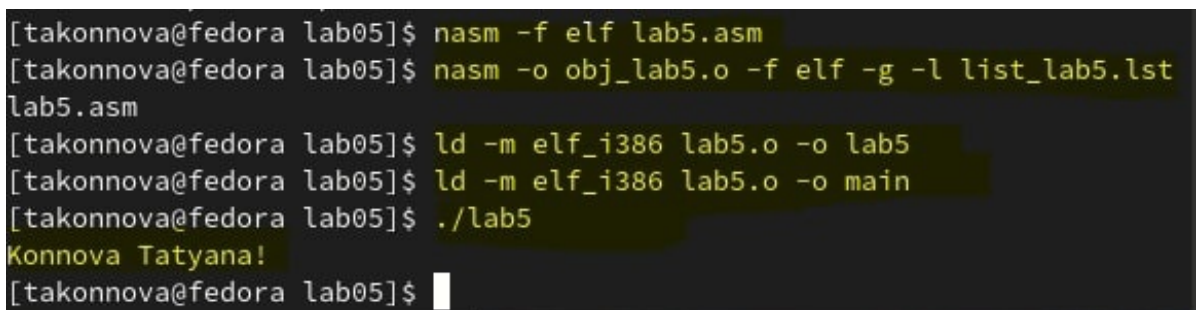


```
lab5.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05

1 SECTION .data
2     hello:      db "Konnova Tatyana!",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. 8.2: Замена вывода на Фамилию и Имя

3. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.(рис. 8.3)



```
[takonnova@fedora lab05]$ nasm -f elf lab5.asm
[takonnova@fedora lab05]$ nasm -o obj_lab5.o -f elf -g -l list_lab5.lst lab5.asm
[takonnova@fedora lab05]$ ld -m elf_i386 lab5.o -o lab5
[takonnova@fedora lab05]$ ld -m elf_i386 lab5.o -o main
[takonnova@fedora lab05]$ ./lab5
Konnova Tatyana!
[takonnova@fedora lab05]$
```

Рис. 8.3: Трансляция в объектный файл, компоновка

4. Скопируем файлы hello.asm и lab5.asm в свой локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/. Загрузим файлы на Github(рис. 8.4)

```

[takonnova@fedora lab05]$ cd /home/takonnova/work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab05
[takonnova@fedora lab05]$ git add .
[takonnova@fedora lab05]$ git commit -am 'feat(main): add files lab-5'
[master bfc9e2b] feat(main): add files lab-5
11 files changed, 66 insertions(+)
create mode 100755 labs/lab05/hello
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/hello.o
create mode 100755 labs/lab05/lab5
create mode 100644 labs/lab05/lab5.asm
create mode 100644 labs/lab05/lab5.o
create mode 100644 labs/lab05/list.lst
create mode 100644 labs/lab05/list_lab5.lst
create mode 100755 labs/lab05/main
create mode 100644 labs/lab05/obj.o
create mode 100644 labs/lab05/obj_lab5.o
[takonnova@fedora lab05]$ git push
Перечисление объектов: 17, готово.
Подсчет объектов: 100% (17/17), готово.
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (14/14), 2.97 КиБ | 1.49 МиБ/с, готово.
Всего 14 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To github.com:KONNOVAT/study_2022-2023_arh-pc.git
 17e643b..bfc9e2b master -> master

```

Рис. 8.4: Переносим, коммитим

9 Выводы

Мы наглядно освоили процедуры компиляции и сборки программ, написанных на ассем- блере NASM.