
Front matter

title: "Отчёт по лабораторной работе №4"

subtitle: "дисциплина: Архитектура компьютера"

author: "Абуковв Ислам Ренатович"

Generic otions

lang: ru-RU

toc-title: "Содержание"

Bibliography

bibliography: bib/cite.bib

csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

Pdf output format

toc: true # Table of contents

toc-depth: 2

lof: true # List of figures

lot: true # List of tables

fontsize: 12pt

linestretch: 1.5

papersize: a4

documentclass: scrreprt

l18n polyglossia

polyglossia-lang:

name: russian

options:

- spelling=modern

- babelshorthands=true

polyglossia-otherlangs:

name: english

l18n babel

babel-lang: russian

babel-otherlangs: english

Fonts

mainfont: IBM Plex Serif

romanfont: IBM Plex Serif

sansfont: IBM Plex Sans

monofont: IBM Plex Mono

mathfont: STIX Two Math

mainfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94

romanfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94

sansfontoptions: Ligatures=Common,Ligatures=TeX,Scale=MatchLowercase,Scale=0.94

monofontoptions: Scale=MatchLowercase,Scale=0.94,FakeStretch=0.9

mathfontoptions:

Biblatex

biblatex: true

biblio-style: "gost-numeric"

biblatexoptions:

- parenttracker=true
- backend=biber
- hyperref=auto
- language=auto
- autolang=other*
- citestyle=gost-numeric

Pandoc-crossref LaTeX customization

figureTitle: "Рис."

tableTitle: "Таблица"

listingTitle: "Листинг"

lofTitle: "Список иллюстраций"

lotTitle: "Список таблиц"

lolTitle: "Листинги"

Misc options

indent: true

header-includes:

- \usepackage[indentfirst]
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text

Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

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.

Теоретическое введение

Основными функциональными элементами любой 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 плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора 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, преобразование (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

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. Периферийные устройства в составе ЭВМ:

- устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных.
- 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.

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. 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.

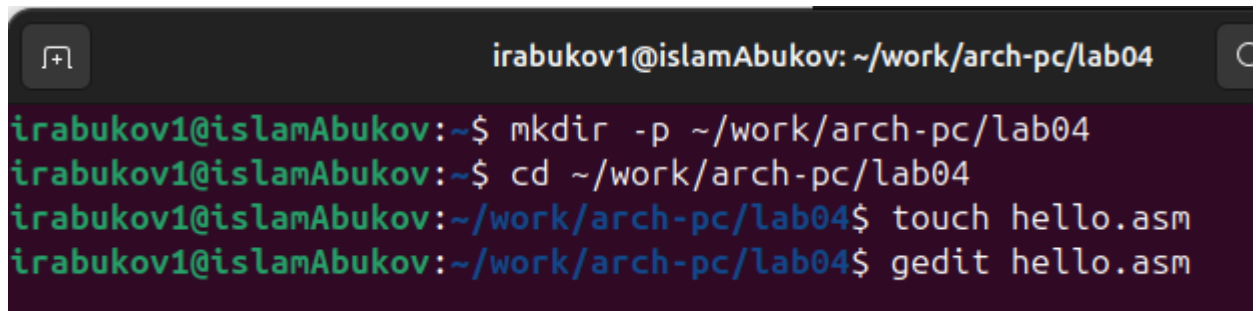
Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.

Выполнение лабораторной работы

Программа Hello world!

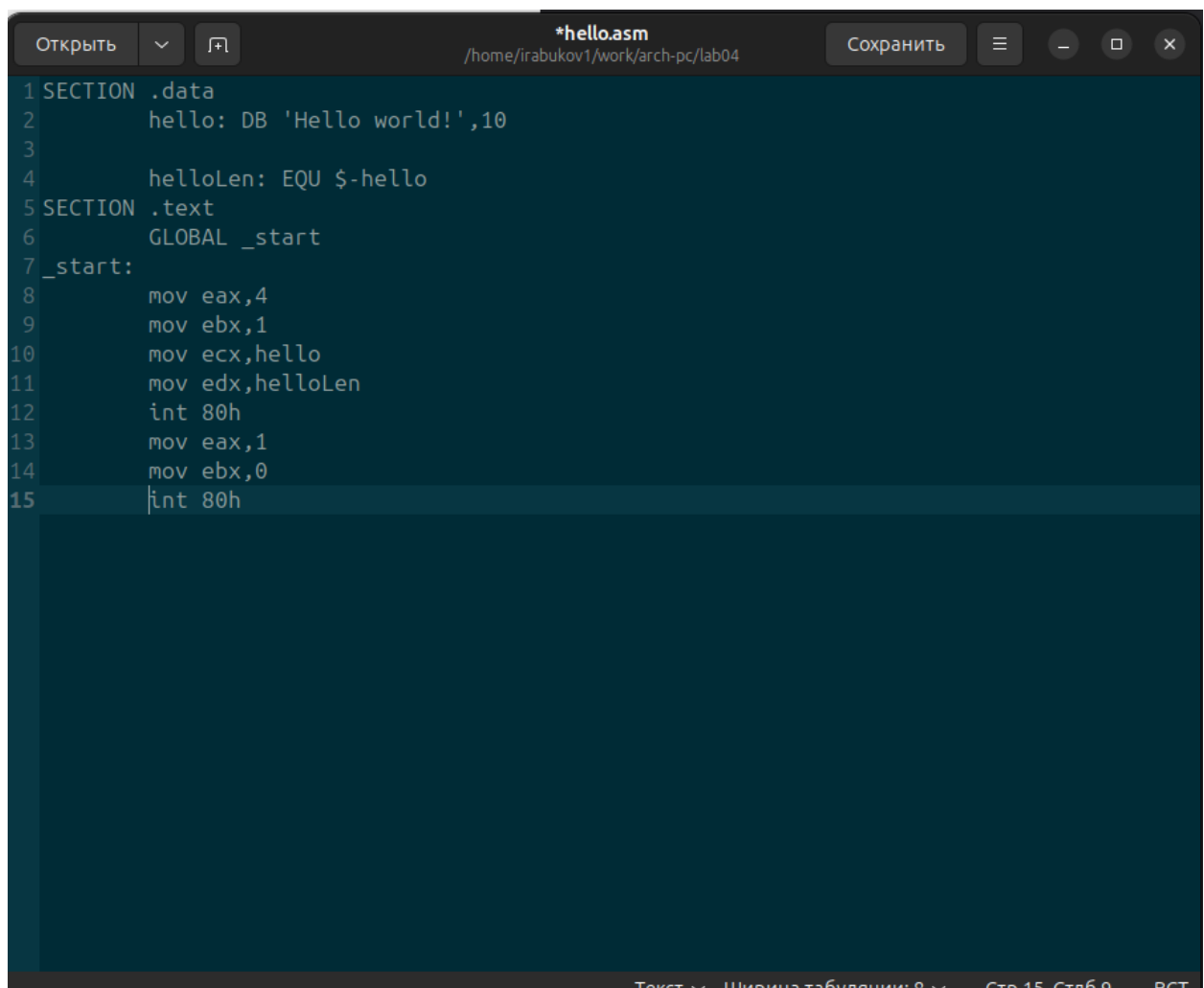
В домашней директории создаю каталог, в котором буду хранить файлы для текущей лабораторной работы.

Создаю в нем файл hello.asm, в котором буду писать программу на языке ассемблера.



```
irabukov1@islamAbukov: ~/work/arch-pc/lab04
irabukov1@islamAbukov:~$ mkdir -p ~/work/arch-pc/lab04
irabukov1@islamAbukov:~$ cd ~/work/arch-pc/lab04
irabukov1@islamAbukov:~/work/arch-pc/lab04$ touch hello.asm
irabukov1@islamAbukov:~/work/arch-pc/lab04$ gedit hello.asm
```

С помощью редактора пишу программу в созданном файле.



```
*hello.asm
/home/irabukov1/work/arch-pc/lab04
Сохранить

1 SECTION .data
2     hello: DB 'Hello world!',10
3
4     helloLen: EQU $-hello
5 SECTION .text
6     GLOBAL _start
7 _start:
8     mov eax,4
9     mov ebx,1
10    mov ecx,hello
11    mov edx,helloLen
12    int 80h
13    mov eax,1
14    mov ebx,0
15    int 80h

Текст × Ширина табуляции: 8 × Стр 15 Стлб 9 ВСТ
```

Транслятор NASM

Компилирую с помощью NASM свою программу.

```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ nasm -f elf hello.asm
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
irabukov1@islamAbukov:~/work/arch-pc/lab04$
```

Расширенный синтаксис командной строки NASM

Выполняю команду, указанную на (рис. -@fig:005), она скомпилировала исходный файл hello.asm в obj.o, расширение .o говорит о том, что файл - объектный, помимо него флаги -g -l подготовят файл отладки и листинга соответственно.

```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst
hello.asm
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
irabukov1@islamAbukov:~/work/arch-pc/lab04$
```

Компоновщик LD

Затем мне необходимо передать объектный файл компоновщику, делаю это с помощью команды ld.

Выполняю следующую команду ..., результатом исполнения команды будет созданный файл main, скомпонованный из объектного файла obj.o.

```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Запуск исполняемого файла

Запускаю исполняемый файл из текущего каталога.

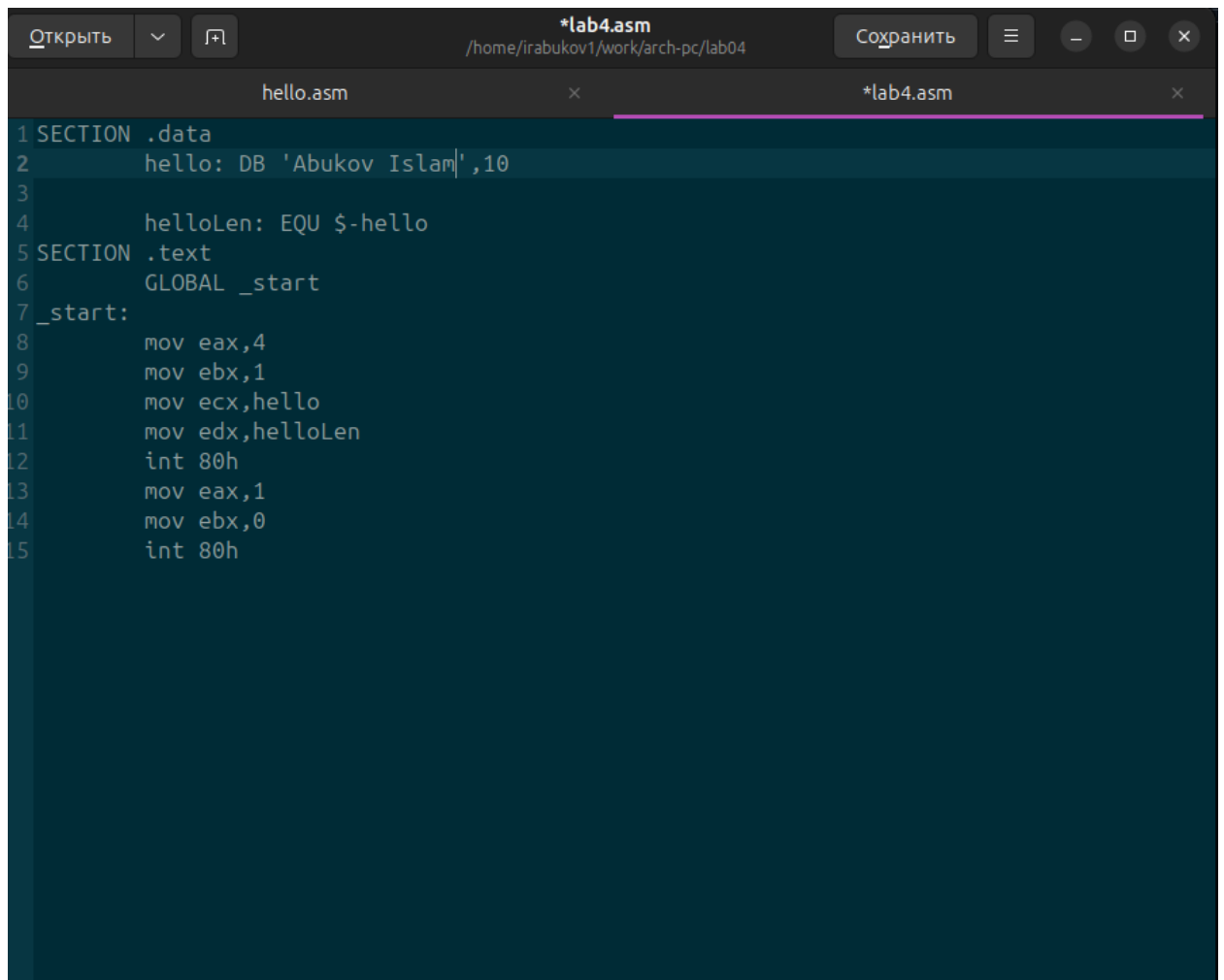
```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ./hello
Hello world!
irabukov1@islamAbukov:~/work/arch-pc/lab04$
```

Задания для самостоятельной работы

Создаю копию файла для последующей работы с ней.

```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
irabukov1@islamAbukov:~/work/arch-pc/lab04$
```

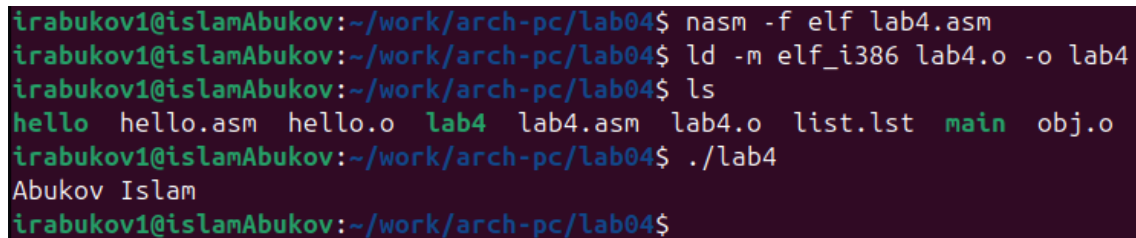
Редактирую копию файла, заменив текст на свое имя и фамилию.



The screenshot shows a code editor with two tabs: 'hello.asm' and '*lab4.asm'. The '*lab4.asm' tab is active and contains the following assembly code:

```
1 SECTION .data
2     hello: DB 'Abukov Islam',10
3
4     helloLen: EQU $-hello
5 SECTION .text
6     GLOBAL _start
7 _start:
8     mov eax,4
9     mov ebx,1
10    mov ecx,hello
11    mov edx,helloLen
12    int 80h
13    mov eax,1
14    mov ebx,0
15    int 80h
```

Транслирую копию файла в объектный файл, компоную и запускаю.



```
irabukov1@islamAbukov:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
irabukov1@islamAbukov:~/work/arch-pc/lab04$ ./lab4
Abukov Islam
irabukov1@islamAbukov:~/work/arch-pc/lab04$
```

Убедившись в корректности работы программы, копирую рабочие файлы в свой локальный репозиторий.

Загрузка изменений на свой удаленный репозиторий на GitHub.

```
irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04
irabukov1@islamAbukov: ~/work/arch-pc/lab04
irabukov1@islamAbukov: ~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04/
irabukov1@islamAbukov: ~/work/arch-pc/lab04$ cd ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04/
irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04$ git add .
irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04$ git status
Текущая ветка: master
Эта ветка соответствует «origin/master».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:   hello.asm
    новый файл:   lab4.asm

irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04$ git commit -m "feat(main): upload 4 lab work"
[master 80bb5b9] feat(main): upload 4 lab work
2 files changed, 30 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
irabukov1@islamAbukov: ~/work/study/2025-2026/Архитектура компьютер/arch-pc/labs/lab04$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
```