

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: *Архитектура компьютера*

Студент: Лопатченко Полина Андреевна

Группа: НКАбд-04-25

МОСКВА

2025 г.

Оглавление

1. Цель работы.....	4
2. Задания.....	5
3. Теоретическое введение.....	6
4. Выполнение лабораторной работы	8
4.1 Программа Hello world!	8
4.2 Транслятор NASM.....	9
4.3 Расширенный синтаксис командной строки NASM	10
4.4 Компоновщик LD	10
4.5 Запуск исполняемого файла	11
4.6 Задания для самостоятельной работы.....	11
5. Выводы	13
6. Список литературы.....	14

Список иллюстраций

Рисунок 1 Создание рабочей директории	8
Рисунок 2 Создание .asm файла	8
Рисунок 3 Редактирование файла	9
Рисунок 4 Компиляция программы	9
Рисунок 5 Возможности синтаксиса NASM.....	10
Рисунок 6 Отправка файла компоновщику	10
Рисунок 7 Создание исполняемого файла	10
Рисунок 8 Запуск программы.....	11
Рисунок 9 Создание копии.....	11
Рисунок 11 Проверка работоспособности скомпонованной программы.....	12

1. Цель работы

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

2. Задания

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 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 объема, 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.

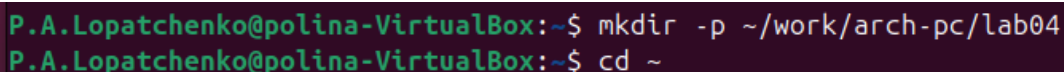
Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса данных, которые участвуют в выполнении 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) — машинноориентированный 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 Программа *Hello world!*

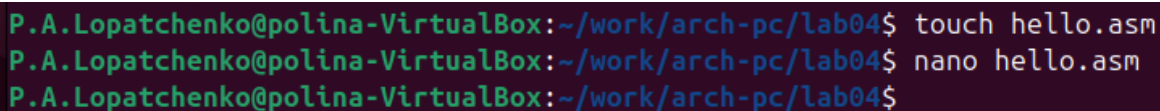
В домашней директории создаю каталог, в котором буду хранить файлы для текущей лабораторной работы. (Рисунок 1)



```
P.A.Lopatchenko@polina-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
P.A.Lopatchenko@polina-VirtualBox:~$ cd ~
```

Рисунок 1 Создание рабочей директории

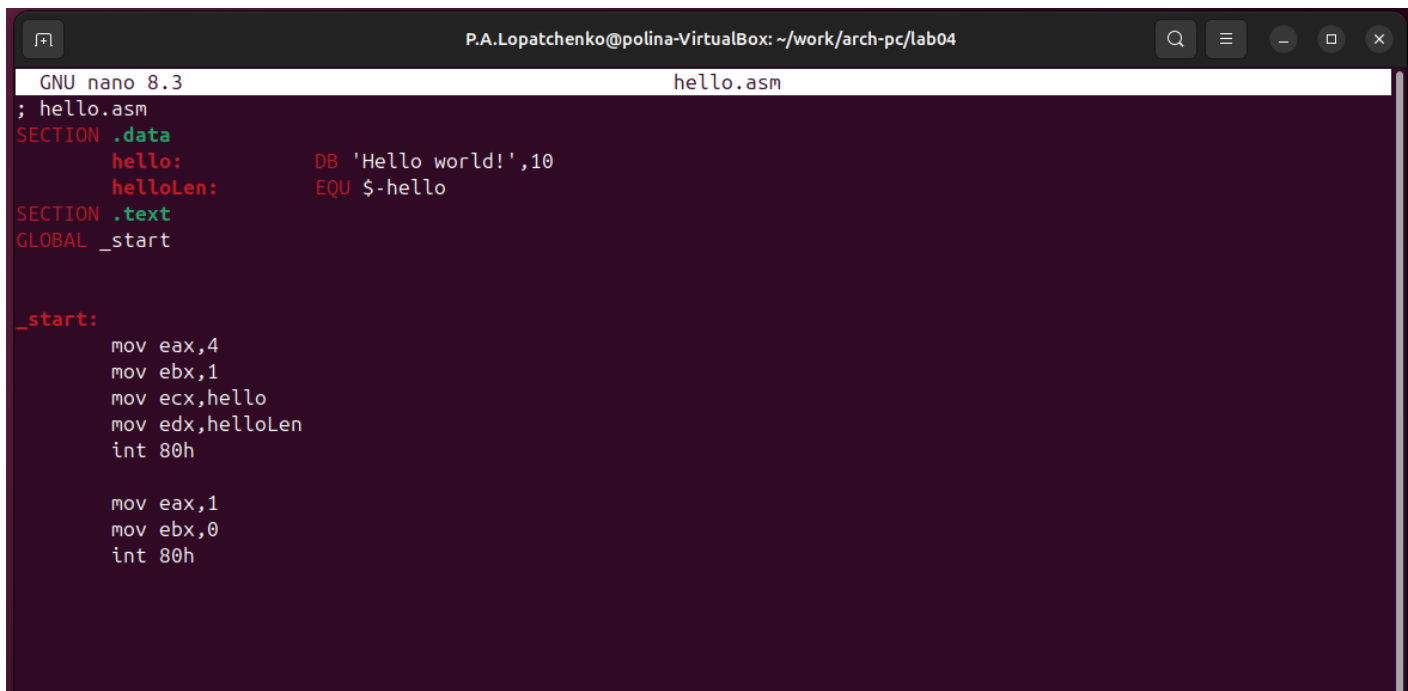
Создаю в нем файл `hello.asm`, в котором буду писать программу на языке ассемблера. (Рисунок 2)



```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nano hello.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 2 Создание `.asm` файла

С помощью редактора пишу программу в созданном файле (Рисунок 3)



```
GNU nano 8.3 hello.asm
; hello.asm
SECTION .data
    hello:      DB 'Hello world!',10
    helloLen:    EQU $-hello
SECTION .text
GLOBAL _start

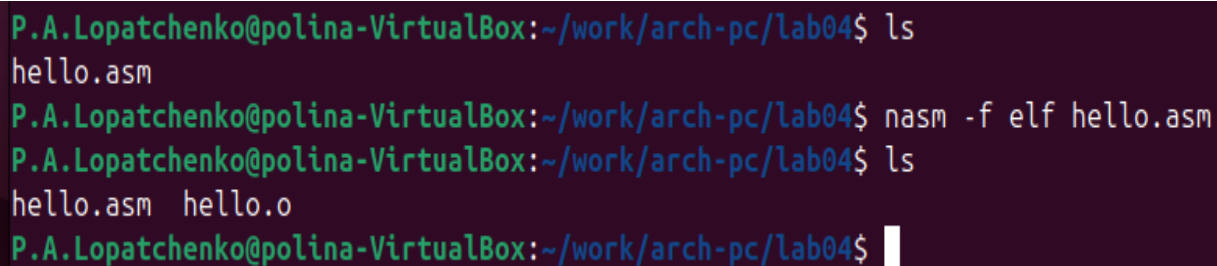
_start:
    mov eax,4
    mov ebx,1
    mov ecx,hello
    mov edx,helloLen
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рисунок 3 Редактирование файла

4.2 Транслятор NASM

Компилирую с помощью NASM свою программу. (Рисунок 4)



```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 4 Компиляция программы

4.3 Расширенный синтаксис командной строки NASM

Выполняя команду, указанную на (Рисунок 5), 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.

```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm hello.o list.lst obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 5 Возможности синтаксиса NASM

4.4 Компоновщик LD

Затем мне необходимо передать объектный файл компоновщику, делаю это с помощью команды `ld`. (Рисунок 6)

```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 6 Отправка файла компоновщику

Выполняя следующую команду ..., результатом исполнения команды будет созданный файл `main`, скомпонованный из объектного файла `obj.o`. (Рисунок 7)

```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 7 Создание исполняемого файла

4.5 Запуск исполняемого файла

Запускаю исполняемый файл из текущего каталога. (Рисунок 8)

```
hello hello.oasm hello.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 8 Запуск программы

4.6 Задания для самостоятельной работы

Создаю копию файла для последующей работы с ней. (Рисунок 9)

```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 9 Создание копии

Редактирую копию файла, заменив текст на свое имя и фамилию (Рисунок 10)

```
GNU nano 8.3 lab4.asm *
; hello.asm
SECTION .data
    hello:      DB 'Lopatchenko Polina',10
    helloLen:   EQU $-hello
SECTION .text
GLOBAL _start

_start:
    mov eax,4
    mov ebx,1
    mov ecx,hello
    mov edx,helloLen
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рисунок 10 Редактирование копии

Транслирую копию файла в объектный файл, компоную и запускаю (Рисунок 11)

```
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nano lab4.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o main
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$ ./lab4
Lopatchenko Polina!
P.A.Lopatchenko@polina-VirtualBox:~/work/arch-pc/lab04$
```

Рисунок 11 Проверка работоспособности скомпонованной программы

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

5. Выводы

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

6. Список литературы

1. <https://esystem.rudn.ru/mod/page/view.php?id=1030505>
2. https://github.com/tasavenkova/study_2025-2026_arh-pc
3. https://esystem.rudn.ru/pluginfile.php/2089084/mod_resource/content/0/Лабораторная%20работа%20№4.%20Создание%20и%20процесс%20обработки%20программ%20на%20языке%20ассемблера%20NASM.pdf