**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

**дисциплина: Архитектура компьютеров и**

**операционные системы**

Студент: Райко М. В.

Группа: НБИбд-03–23

МОСКВА

2023 г.

**Содержание**

**1.Цель работы...................................................................................3**

**2.Задание............................................................................................4**

**3.Теоритическое введение.............................................................5**

**4.Выполнение лабораторной работы.....................................6-13**

* **4.1** Программа Hello world!..................................................**6-7**
* **4.2** Транслятор NASM..............................................................**8**
* **4.3** Расширенный синтаксис командной строки NASM.......**9**
* **4.4** Компоновщик LD.............................................................**10**
* **4.5** Запуск исполняемого файла…........................................**11**
* **4.6** Задание для самостоятельной работы.......................**12-14**

**5.Заключение..................................................................................15**

**1.Цель работы:**

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

**2.Задание:**

* **4.1** Программа Hello world!
* **4.2** Транслятор NASM
* **4.3** Расширенный синтаксис командной строки NASM
* **4.4** Компоновщик LD
* **4.5** Запуск исполняемого файла
* **4.6** Задание для самостоятельной работы

**3.Теоретическое введение:**

ЭВМ состоит из центрального процессора (ЦП), памяти и периферийных устройств, связанных общей шиной. ЦП включает арифметико-логическое устройство (выполняет операции), устройство управления (контролирует устройства) и регистры (хранят данные). В программировании на ассемблере регистры используются для операций с данными, таких как перемещение и арифметика.

ЭВМ включает периферийные устройства для хранения данных (например, жёсткие диски) и устройства ввода-вывода для взаимодействия с внешней средой. Программное управление осуществляется машинными командами, состоящими из операционной и адресной частей. Команды выполняются в командном цикле процессора, включающем формирование адреса, считывание, дешифрацию, выполнение и переход к следующей команде.

Язык ассемблера (asm)**:** это машинно-ориентированный язык низкого уровня, близкий к архитектуре ЭВМ. В отличие от языков высокого уровня, ассемблер позволяет более полный контроль над аппаратными ресурсами. Программы на ассемблере состоят из команд, понятных процессору, записанных с использованием мнемоник. Ассемблерные программы транслируются в машинный код транслятором (ассемблером), обеспечивая высокую производительность. На языке ассемблера программисты указывают инструкции, которые прямо соответствуют операциям процессора, сохраняя контроль над ресурсами.

Создание ассемблерной программы:

Написание и сохранение: Программист пишет код на ассемблере и сохраняет в файл (.asm).

Трансляция: Транслятор (например, nasm) преобразует текст программы в машинный код, создавая объектный файл (. o) и листинг (.lst).

Компоновка: Компоновщик (ld) обрабатывает объектные файлы, создавая исполняемый файл без расширения, возможно, с картой загрузки (.map).

Запуск программы: Исполняемый файл запускается. Ошибки требуют отладки с использованием отладчика, после чего цикл повторяется.

**4.Выполнение лабораторной работы:**

**4.1** Программа Hello world!:

Создайте каталог для работы с программами на языке ассемблера NASM:



(4.1.1 Создание каталога лаб4)

Перейдите в созданный каталог:



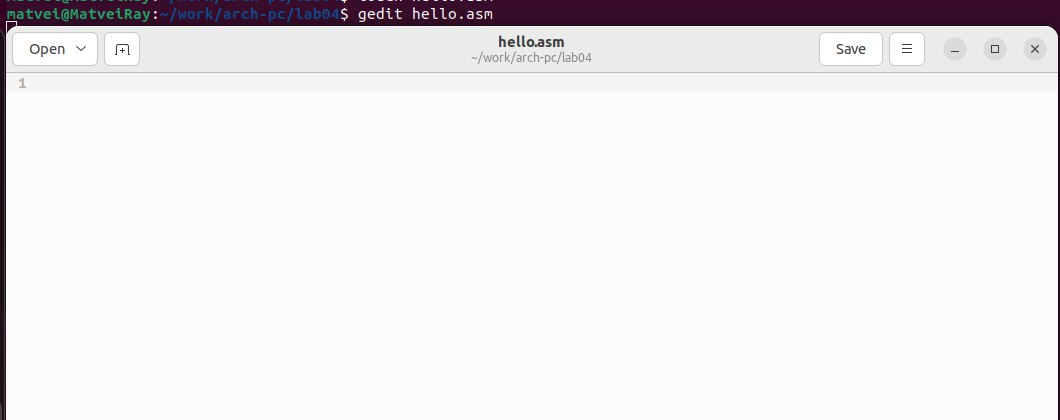
(4.1.2 Перешел в созданный каталог)

Создайте текстовый файл с именем hello.asm:



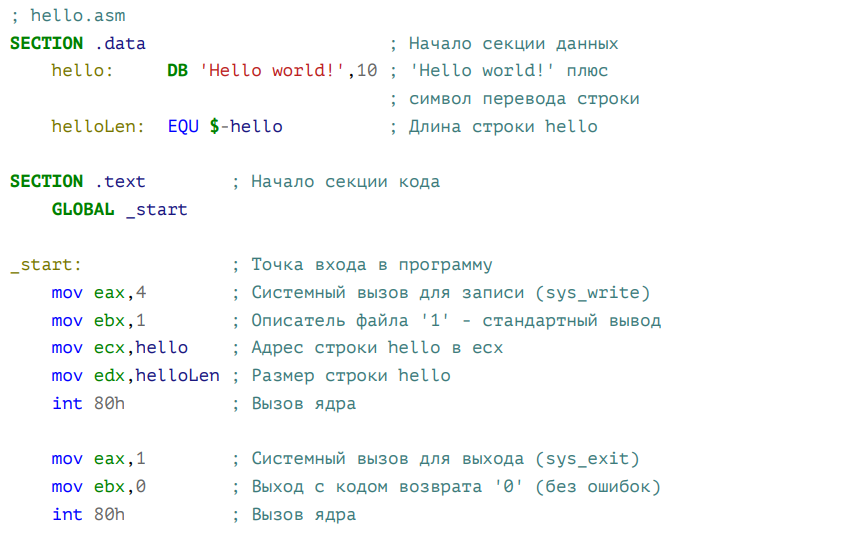
(4.1.3 Создал текстовый файл)

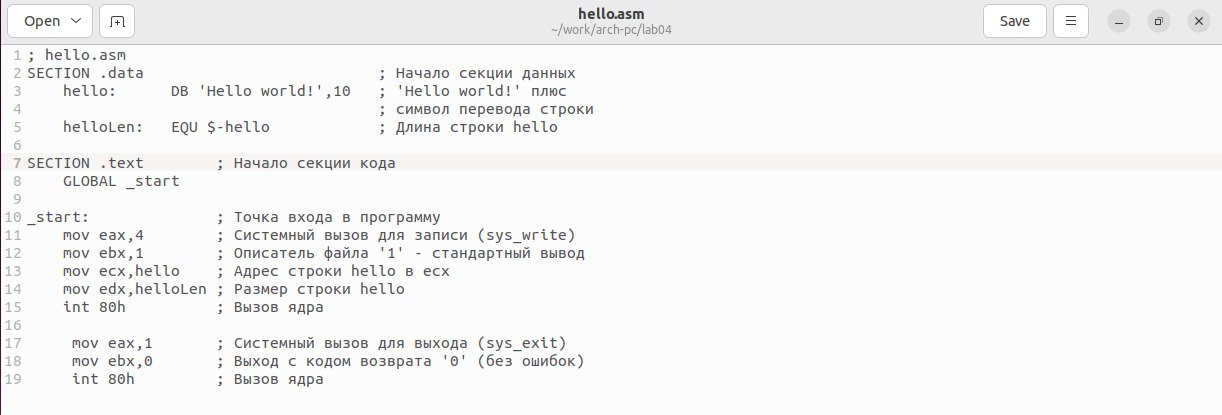
Откройте этот файл с помощью любого текстового редактора, например, gedit:



(4.1.4 Открыл текстовый файл с помщью команды gedit)

и введите в него следующий текст:



(4.1.5 Ввел синтаксис,который выводит в ассемблере программу Hello World)

**4.2** Транслятор NASM:

Для компиляции приведённого выше текста программы «Hello World» необходимо написать:



(4.2.1 Преобразование текстового файла в объективный код с помощью транслятора NASM)

С помощью команды ls проверьте, что объектный файл был создан:



(4.2.2 NASM транслировал в объективный код с названием hello.o)

**4.3** Расширенный синтаксис командной строки NASM:

Выполните следующую команду

(данная команда скомпилирует исходный файл hello.asm в obj.o (опция -o позволяет задать имя объектного файла, в данном случае obj.o), при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, будет создан файл листинга list.lst (опция -l).) :



(4.3.1 Компиляция файла)

**4.4** Компоновщик LD:

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



(4.4.1 Передаем файл компановщику,чтобы исполняемый файл Hello был создан)

С помощью команды ls проверьте, что исполняемый файл hello был создан:



(4.4.2 Утверждаемся в этом)

Выполните следующую команду:



(4.4.3 Объединение объектного файла в исполняемый файл ELF под архитектуру x86)

**4.5** Запуск исполняемого файла:

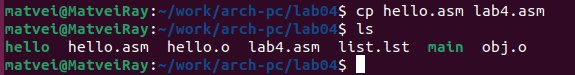
Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:



(4.5.1 Вывод программы Hello world!)

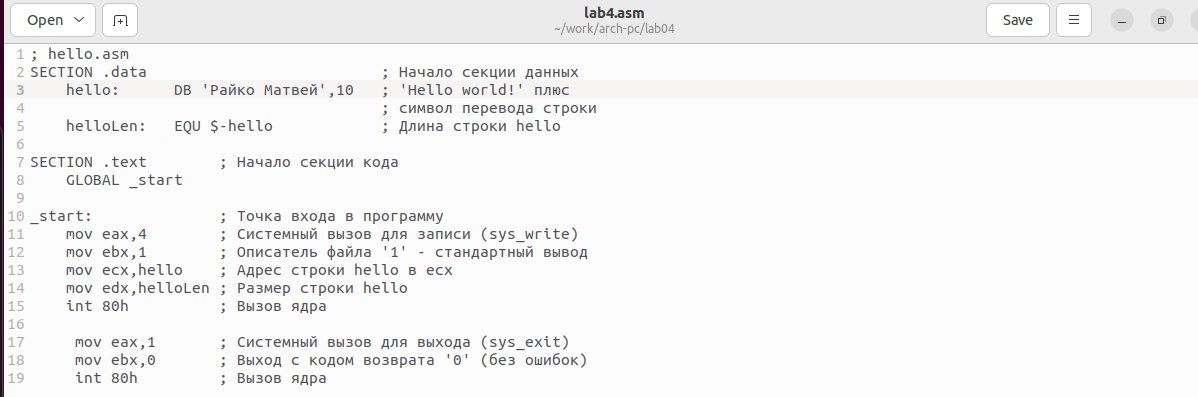
**4.6** Задание для самостоятельной работы:

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла hello.asm с именем lab4.asm



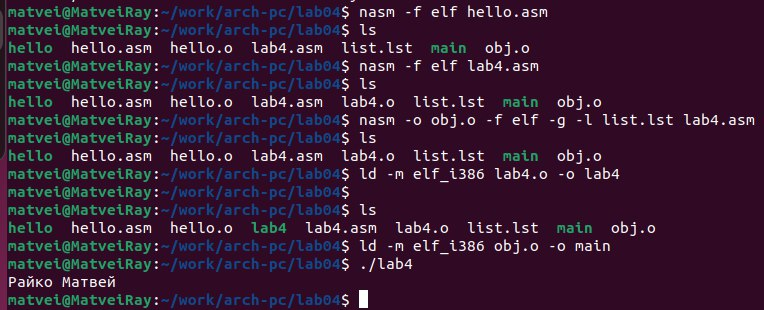
(4.6.1 Создал копию с помощью команды cp)

2. С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.



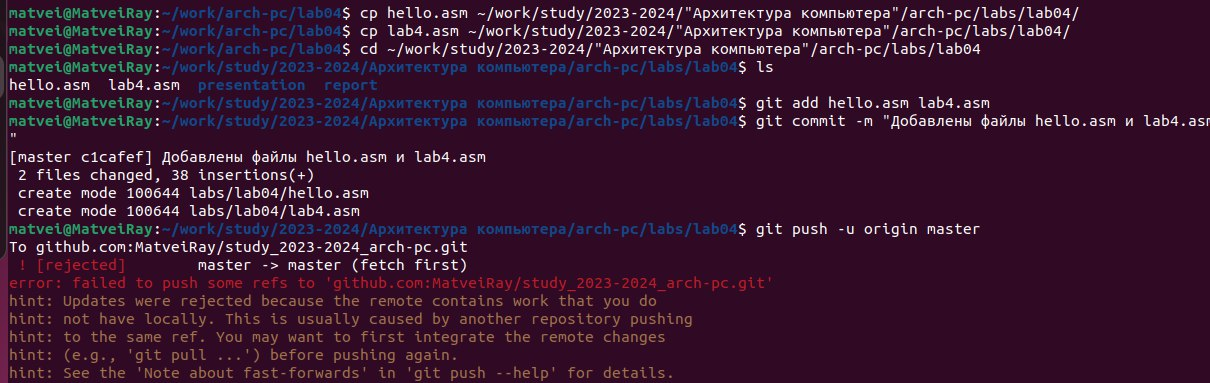
(4.6.2 Ввел свою Фамилию и свое имя в текстовый редактор)

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.

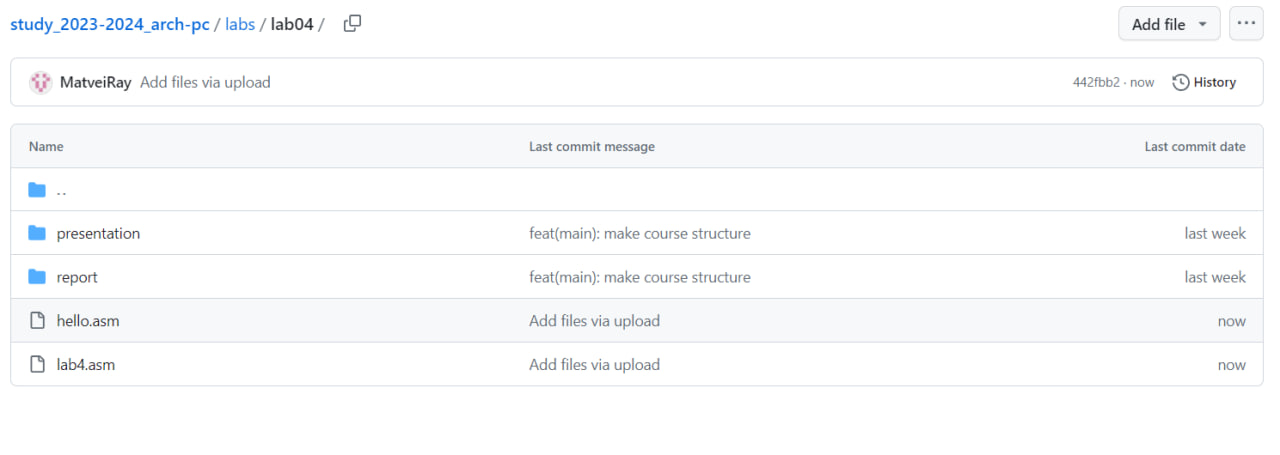


(4.6.3 Компановка объектного файла lab4 и последующий его запуск)

4. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузите файлы на Github:



(4.6.4 Скопировал файлы,загрузил в локал каталог,а после загрузил на гитхаб)



(4.6.5 Проверил на гитхабе загруженные файлы)

**5. Заключение:**

Освоение ассемблера способствует развитию навыков оптимизации кода, что приводит к повышению производительности программ. Кроме того, данное изучение открывает путь к освоению более высокоуровневых языков программирования, предоставляя более высокий уровень абстракции и удобства при написании кода. Приобретенные знания по языку ассемблера также облегчают изучение работы компилятора и понимание того, какие инструкции выполняются на более низком уровне.