Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров.

Ворожейкин Владимир Вячеславович

# Содержание

Цель работы…………………………………………………………………………………1

Задание………………………………………………………………………………………..1

Теоретическое введение………………………………………………………………...1

Выполнение лабораторной работы…………………………………………………..3

Создание программы Hello world!…………………………………………………….3

[Работа с транслятором NASM……………………………………………………….](#_Toc150737088)3

[Работа с расширенным синтаксисом командной строки NASM…………](#_Toc150737089)3

[Работа с компоновщиком LD………………………………………………………](#_Toc150737090)...3

[Запуск исполняемого файла…………………………………………………………](#_Toc150737091)3

[Выполнение заданий для самостоятельной работы………..………………..3Ошибка: источник перекрёстной ссылки не найден](#_Toc150737092)

[Выводы………………………………………………………………………………………](#_Toc150737093)5

[Список литературы………………………………………………………………………](#_Toc150737094)..55

# 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

Основные функциональные элементы ЭВМ (электронно-вычислительные машины) работают совместно для выполнения операций обработки информации. Ниже описаны несколько ключевых элементов и их работы. 1. Центральный процессор (ЦП) - это “мозг” компьютера. Он выполняет основные вычисления, управляет и контролирует операции в ЭВМ. ЦП состоит из нескольких частей: - Арифметико-логическое устройство (АЛУ) - выполняет арифметические и логические операции, например, сложение, вычитание, умножение, деление и сравнение. - Устройство управления - принимает команды и данные, декодирует их и управляет внутренними ресурсами ЦП, такими как регистры и память. - Регистры - это небольшие хранилища данных в ЦП. Они могут содержать команды, данные и результаты вычислений на протяжении операций. 2. Оперативная память (ОЗУ) - это тип хранилища данных, используемый для временного хранения команд и данных, которые ЦП активно использует. ОЗУ обеспечивает быстрый доступ к данным, и они могут быть прочитаны или записаны через различные адресные линии. 3. Устройства ввода/вывода (В/В) - это устройства, используемые для передачи информации между ЭВМ и внешними устройствами. Примеры устройств В/В включают клавиатуру, мышь, монитор, дисковод, принтер и т.д. Они обеспечивают ввод данных в ЭВМ и вывод результатов. 4. Арифметический логический блок (ALU) - отвечает за выполнение арифметических и логических операций, таких как сложение, вычитание, умножение, деление, сравнение и логические операции (НЕ, ИЛИ, И, XOR и т.д.). 5. Память для хранения данных (например, жесткий диск) - это основное устройство для долговременного хранения данных в ЭВМ. Он используется для сохранения операционной системы, программного обеспечения, файлов пользователя и других данных. 6. Шины - это набор проводов, которые используются для передачи данных, команд и сигналов между различными компонентами ЭВМ, такие как ЦП, ОЗУ и В/В устройства. Все эти элементы работают совместно, обмениваясь информацией и выполняя операции в соответствии с программами и командами, заданными пользователем. Они обеспечивают эффективное выполнение задач и обработку информации в ЭВМ. 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

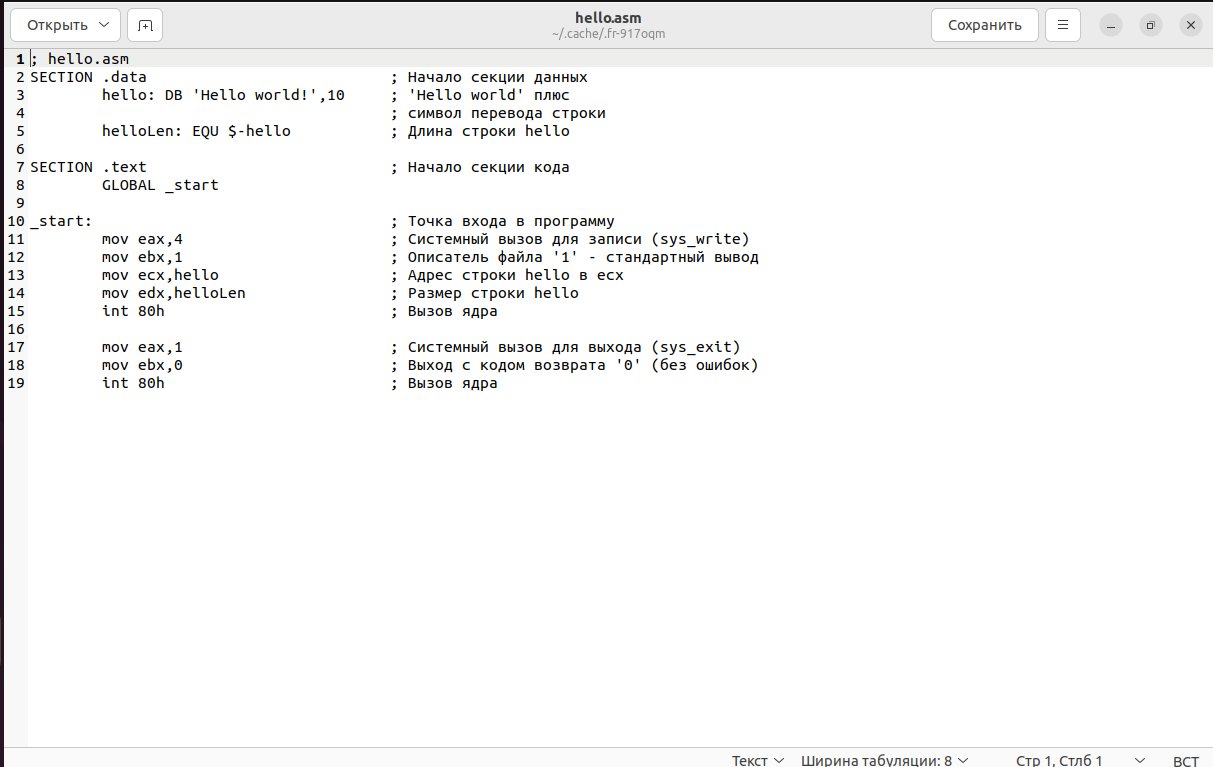
## Создание программы Hello world!

С помощью утилиты cd перемещаюсь в каталог, в котором буду работать (рис. 1), создаю в текущем каталоге пустой текстовый файл hello.asm с помощью touch.



Перемещение между директориями

Заполняю файл, вставляя в него программу для вывода “Hello world!” (рис. 2).



Заполнение файла

## Работа с транслятором NASM.

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду nasm -f elf hello.asm, ключ -f указывает транслятору nasm, что требуется создать бинарный файл в формате ELF (рис. 3).



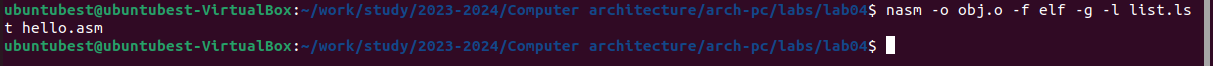
Компиляция текста программы

Далее проверяю правильность выполнения команды с помощью ls (рис.4).

Проверка правильности выполнения

## Работа с расширенным синтаксисом командной строки NASM.

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, использую ключ -o который задает имя объектному файлу, так же в файл будут включены символы для отладки (ключ -g), с помощью ключа -l будет создан файл list.lst (рис. 5). Далее проверяю с помощью ls правильность выполнения команды.



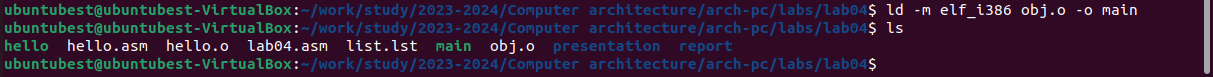
Компиляция текста программы

## Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello (рис. 6). Ключ -о задает имя создаваемого исполняемого файла. Далее проверяю с помощью ls правильность выполнения команды (рис.6)

*Компиляция текста программы*

Выполняю следующую команду (рис. 7). Исполняемый файл будет иметь имя main, т.к. после ключа -о было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o. С помощью ls проверяю правильность выполнения.



Передача объектного файла на обработку компоновщику

## Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 8).

*Запуск исполняемого файла*

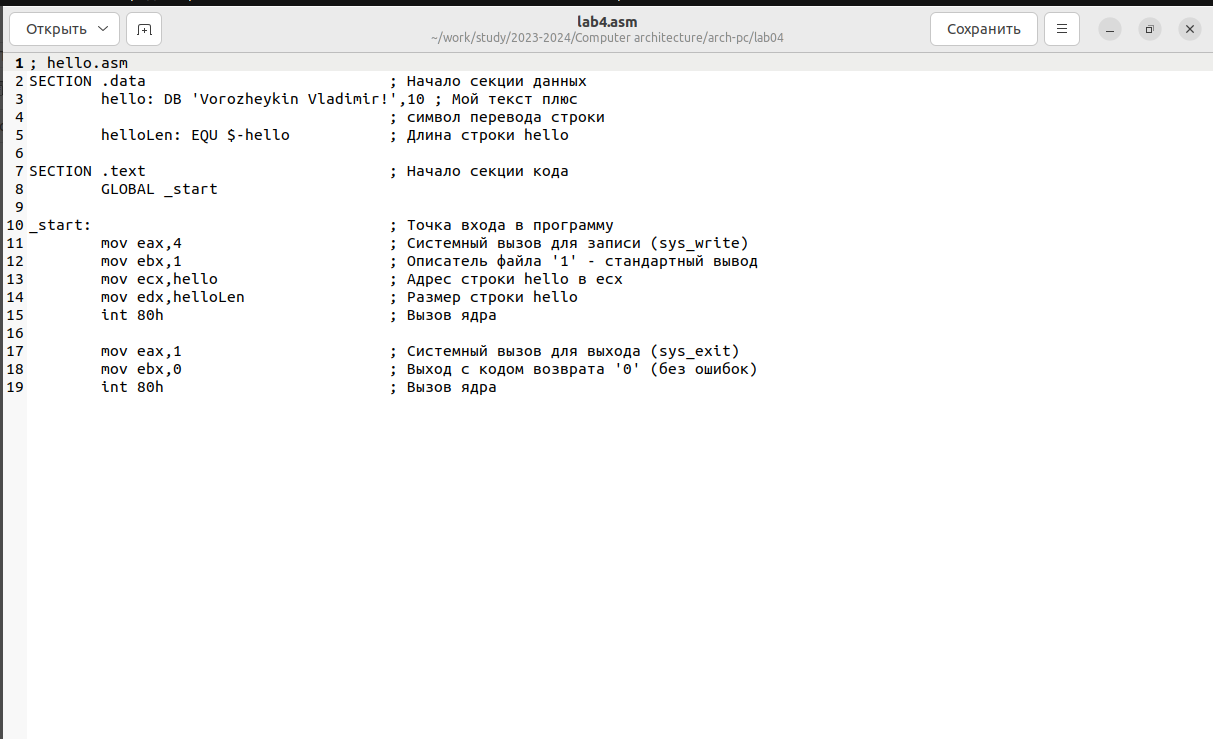
## Выполнение заданий для самостоятельной работы.

С помощью cp создаю в текущем каталоге копию файла hello.asm с именем lab4.asm (рис. 9).



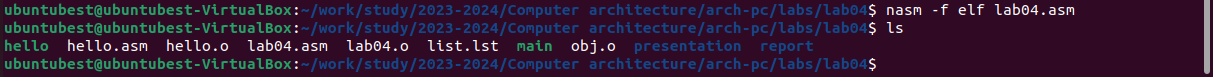
Создание копии файла

С помощью текстового редактора открываю файл lab4.asm и вношу изменения в программу, чтобы она выводила мои имя и фамилию. (рис. 10).



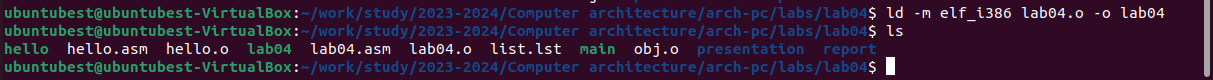
Изменение программы

Компилирую текст программы в объектный файл (рис. 11). Проверяю с помощью ls, что файл lab4.o создан.



Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. 12). С помощью ls проверяю правильность выполнения.



Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4, на экран выводятся мои имя и фамилия (рис. 13).

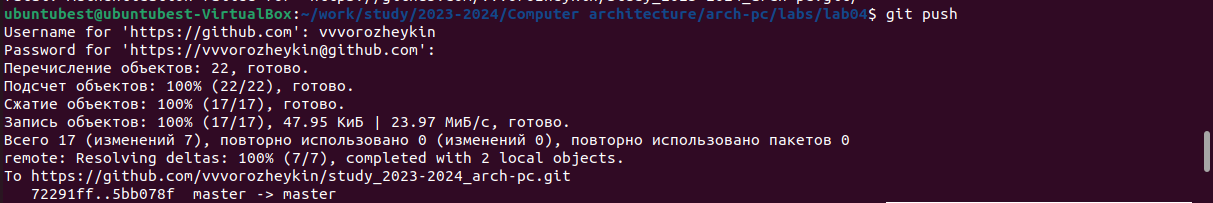
Запуск исполняемого файла

С помощью команды git add . добавляю файлы на GitHub (рис. 14).



Добавление файлов на GitHub

Отправляю файлы на сервер с помощью команды git push (рис. 15).



Отправка файлов

# Выводы

При выполнении этой лабораторной работы я освоил способы компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы

1. https://esystem.rudn.ru/pluginfile.php/1584628/mod\_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%965.pdf