Отчёта по лабораторной работе 4

Создание и процесс обработки программ на языке ассемблера NASM

Тарасова Алина НКАбд 05-23

Содержание

# 1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задание

1. Изучите программу HelloWorld и скомпилируйте ее.
2. С помощью любого текстового редактора внесите изменения в текст программы так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.
3. Скомпилируйте новую программу и проверьте ее работу.
4. Загрузите файлы на GitHub.

# 3 Теоретическое введение

NASM (англ. Netwide Assembler) – это 80x86 ассемблер, разработанный исходя из принципов переносимости и модульности. Он поддерживает широкий диапазон форматов объектных файлов, включая форматы Linuxa.out и ELF, NetBSD/FreeBSD, COFF, Microsoft 16-bit OBJ и Win32. Он способен также создавать простые бинарные файлы. Синтакс NASM максимально упрощен для понимания и похож на синакс Intel, но слегка посложнее. Он поддерживает инструкции Pentium, P6 и MMX, а также имеет макро-расширения.

NASM был создан Саймоном Тэтхемом совместно с Юлианом Холлом и в настоящее время развивается небольшой командой разработчиков на SourceForge.net. Первоначально он был выпущен согласно его собственной лицензии, но позже эта лицензия была заменена на GNU LGPL после множества проблем, вызванных выбором лицензии. Начиная с версии 2.07 лицензия заменена на «упрощённую BSD» (BSD из 2 пунктов).

NASM может работать на платформах, отличных от x86, таких как SPARC и PowerPC, однако код он генерирует только для x86 и x86-64.

NASM успешно конкурирует со стандартным в Linux- и многих других UNIX-системах ассемблером gas. Считается, что качество документации у NASM выше, чем у gas. Кроме того, ассемблер gas по умолчанию использует AT&T-синтаксис, ориентированный на процессоры не от Intel, в то время как NASM использует вариант традиционного для x86-ассемблеров Intel-синтаксиса; Intel-синтаксис используется всеми ассемблерами для под DOS/Windows, например, MASM, TASM, fasm.

В NASM используется Intel-синтаксис записи инструкций. Предложение языка ассемблера NASM (строка программы) может состоять из следующих элементов:Метка, Инструкция, Операнды, Комментарий.

Операнды отделяются между собой запятой. Перед строкой и после инструкции можно использовать любое количество пробельных символов. Комментарий начинается с точки с запятой, а концом комментария считается конец строки. В качестве инструкции может использоваться команда или псевдокоманда (директива компилятора). Если строка очень длинная, то её можно перенести на следующую, используя обратный слеш (  ), подобно тому, как это делается в языке Си.

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Создаю каталог lab04 командой mkdir, перехожу в него с помощью команды cd, создаю файл hello.asm. (рис. [[1](#fig:001)])

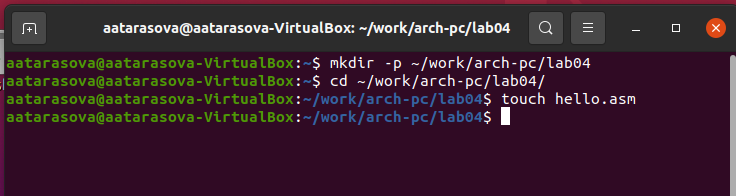


Figure 1: Создание каталога и файла

Открыла файл и написала код программы по заданию.(рис. [[2](#fig:002)])

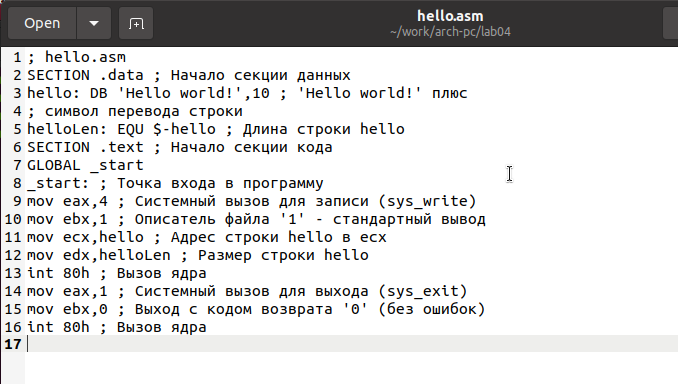


Figure 2: Программа hello.asm

## 4.2 Транслятор NASM

Транслирую файл командой nasm. Получился объектный файл hello.o (рис. [[3](#fig:003)])

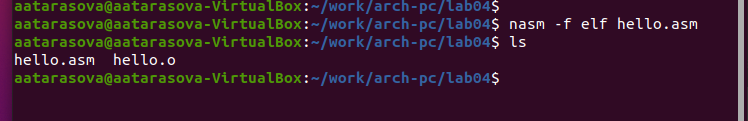


Figure 3: Трансляция hello.asm

Транслирую файл командой nasm с дополнительными опциями. (рис. [[4](#fig:004)]) Получился файл листинга list.lst, объектный файл obj.o, в программу добавилась отладочная информация.

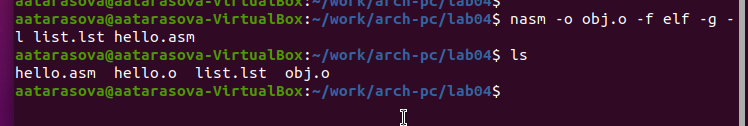


Figure 4: Трансляция hello.asm с дополнительными опциями

## 4.3 Компоновщик LD

Выполняю линковку командой ld и получил исполняемый файл. (рис. [[5](#fig:005)])

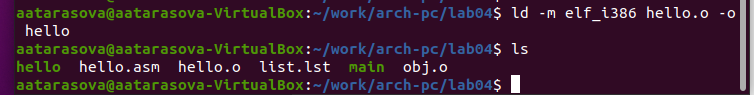


Figure 5: Линковка программы

Еще раз выполняю линковку для объектного файла obj.o и получаю исполняемый файл main.(рис. [[6](#fig:006)])

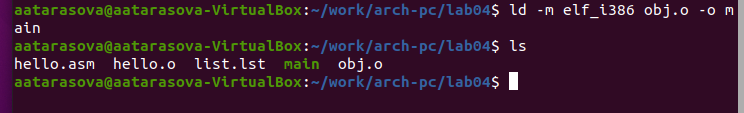


Figure 6: Линковка программы

Запускаю исполняемые файлы.(рис. [[7](#fig:007)])

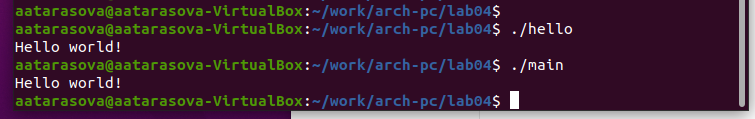


Figure 7: Запуск программ

## 4.4 Выполнение заданий для самостоятельной работы.

Копирую программу в новый файл.

Изменяю сообщение Hello world на свое имя (рис. [[8](#fig:008)]) и запускаю новую программу. (рис. [[9](#fig:009)])

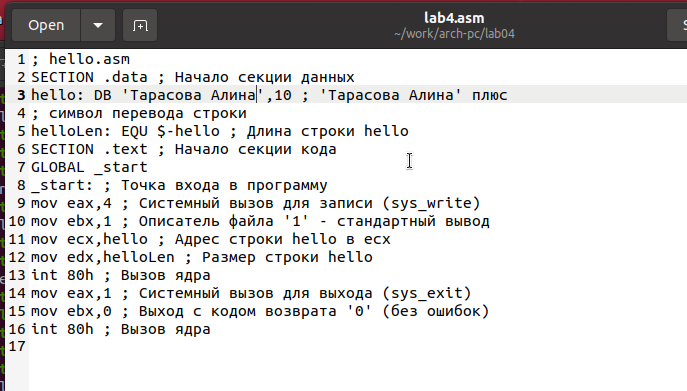


Figure 8: Код программы в файле lab4.asm

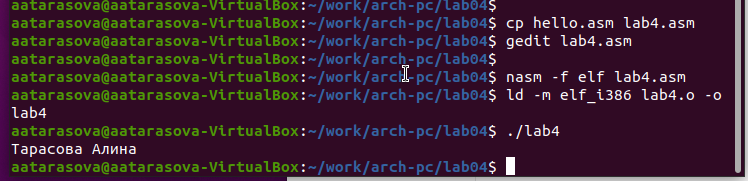


Figure 9: Запуск программы lab4.asm

# 5 Выводы

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

# Список литературы

1. Архитектура ЭВМ
2. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.