Лабораторная работа №4

НКАбд-02-23

Выборнов Дмитрий Валерьевич

# 1 Цель работы

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

# 2 Задание

1. Создание программы Hello world!
2. Использование транслятора NASM.
3. Работа с расширенным синтаксисом командной строки NASM.
4. Использование компоновщика LD.
5. Запуск исполняемого файла.
6. Задания для самостоятельной работы.

# 3 Теоретическое введение

**Язык ассемблера** (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня,таких как C/C++, Perl, Python и пр. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — **машинные коды**. До появления языков ассемблера программистам приходилось писать программы, используя только лишь машинные коды, которые были крайне сложны для запоминания, так как представляли собой числа, записанные в двоичной или шестнадцатеричной системе счисления. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — **Ассемблер**. Программы, написанные на языке ассемблера, не уступают в качестве и скорости программам, написанным на машинном языке,так как транслятор просто переводит мнемонические обозначения команд в последовательности бит (нулей и единиц). Используемые мнемоники обычно одинаковы для всех процессоров одной архитектуры или семейства архитектур (среди широко известных — мнемоники процессоров и контроллеров x86, ARM, SPARC, PowerPC,M68k). Таким образом для каждой архитектуры существует свой ассемблер и, соответственно, свой язык ассемблера. В нашем курсе будет использоваться ассемблер NASM (Netwide Assembler). 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 Создание программы Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в него:

Первый шаг.

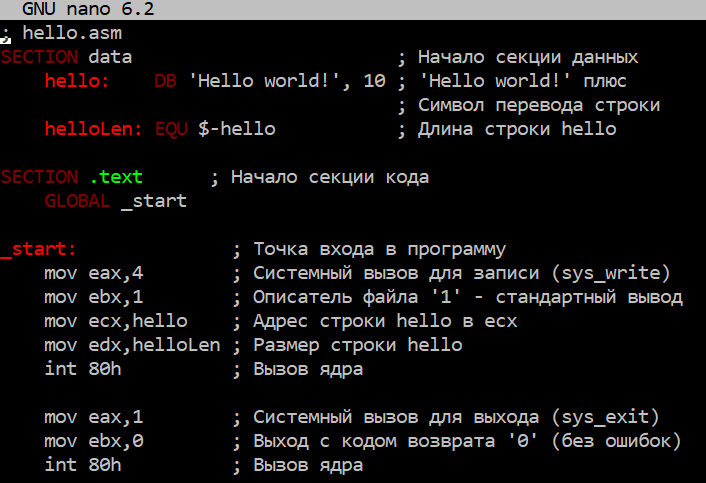
Первый шаг.

Создаю текстовый файл с именем hello.asm и открываю его при помощи nano:

Второй шаг.

Второй шаг.

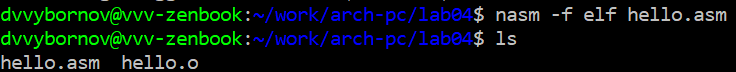
Ввожу в файл текст необходимой программы:



Третий шаг.

## 4.2 Использование транслятора NASM.

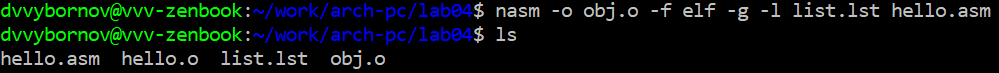
Ввожу команду для использования транслятора NASM и проверяю, что все нужные мне вайлы были созданы:



Четвёртый шаг.

## 4.3 Работа с расширенным синтаксисом командной строки NASM.

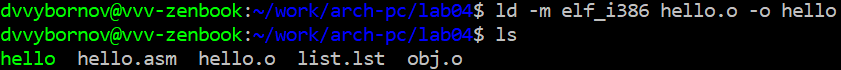
Выполняю команду, которая скомпилирует файл Hello.asm в obj.o и создаст файл листинга, и проверяю, что все нужные файлы созданы:



Пятый шаг.

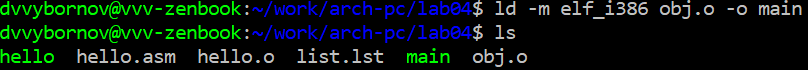
## 4.4 Использование компоновщика LD.

Использую команду ld для создания исполняемого файла Hello и проверяю, что он был создан:



Шестой шаг.

Выполняю команду, которая создаст файл main из объектного файла obj.o:



Седьмой шаг.

## 4.5 Запуск исполняемого файла.

Запускаю созданный исполняемый файл:

Восьмой шаг.

Восьмой шаг.

## 4.6 Задания для самостоятельной работы

### 4.6.1 №1

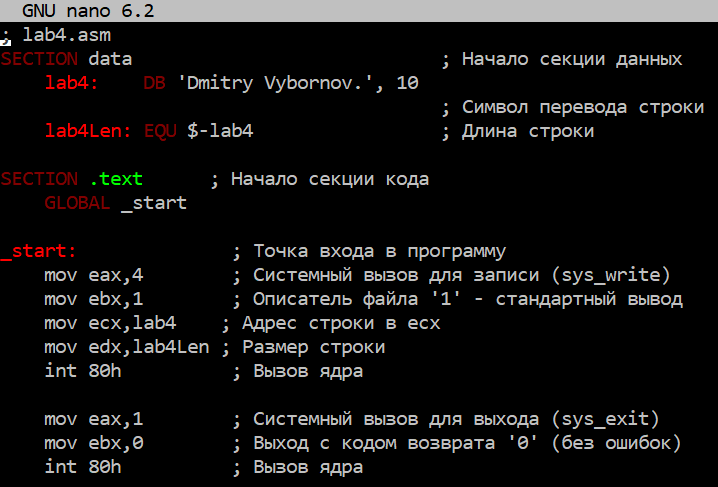
С помощью команды cp создаю копию файла hello.asm с именем lab4.asm:

Первое задание.

Первое задание.

### 4.6.2 №2

Вношу необходимые изменения в файл lab4.asm:



Второе задание.

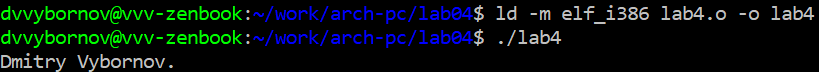
### 4.6.3 №3

Транслирую полученный текст программы в объектный файл:

Первый шаг третьего задания.

Первый шаг третьего задания.

Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл:



Второй шаг третьего задания.

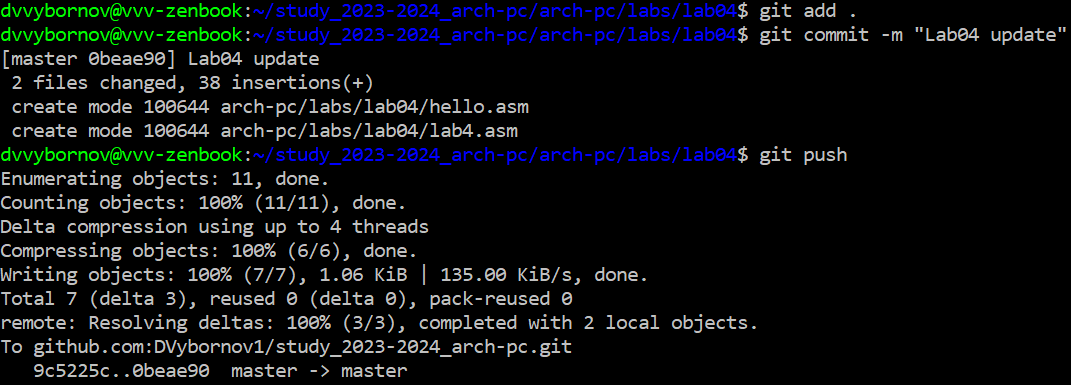
### 4.6.4 №4

Копирую файлы Hello.asm и lab4.asm в нужный репозиторий:

Первый шаг четвёртого задания.

Первый шаг четвёртого задания.

Загружаю файлы на Github:



Второй шаг четвёртого задания.

# 5 Выводы

Выполнив эту лабораторную работу, я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.