Отчёт по лабораторной работе 4

Дисциплина: архитектура компьютера

Плугин Никита

Содержание

# 1 Цель работы

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

# 2 Задания

1. Изучить основы языка Ассемблера
2. Освоить и выполнить процесс компиляции программы на Ассемблере
3. Выполнить самостоятельное задание по изменению программы

# 3 Теоретическое введение

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. Типичный формат записи команд NASM имеет вид:

[метка:] мнемокод [операнд {, операнд}] [; комментарий]

Здесь мнемокод — непосредственно мнемоника инструкции процессору, которая являетсяобязательной частью команды. Операндами могут быть числа,данные, адреса регистров или адреса оперативной памяти. Метка — это идентификатор, с которым ассемблер ассоциирует некоторое число, чаще всего адрес в памяти. Т.о. метка перед командой связана с адресом данной команды.

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

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

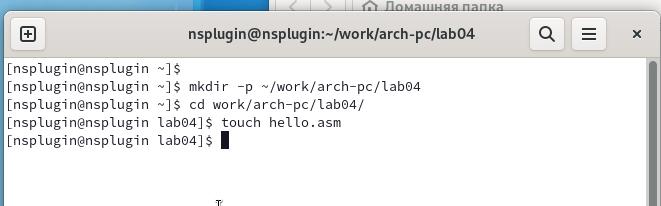


Figure 1: Создание файла для программы

1. Открыл файл и написал код программы по заданию.(рис. [[2](#fig:002)])

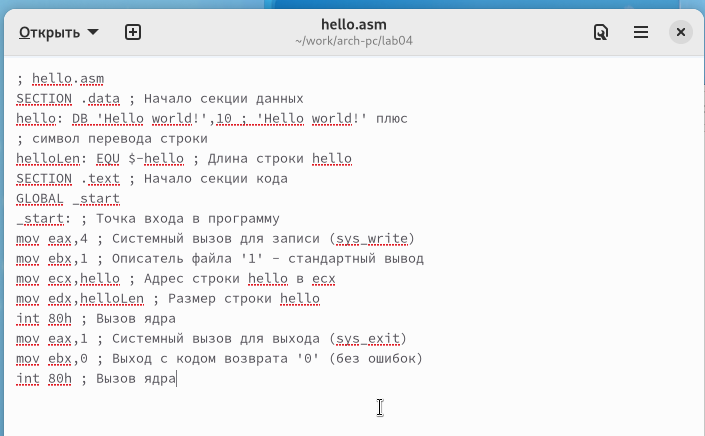


Figure 2: Программа hello.asm

В отличие от многих современных высокоуровневых языков программирования, в ассемблерной программе каждая команда располагается на отдельной строке. Размещение нескольких команд на одной строке недопустимо. Синтаксис ассемблера NASM является чувствительным к регистру, т.е. есть разница между большими и малыми буквами.

## 4.2 Транслятор NASM

1. Транслировал файл командой nasm c опцией -f. (рис. [[3](#fig:003)])

Ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат elf64 позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто elf.

Получился объектный файл hello.o

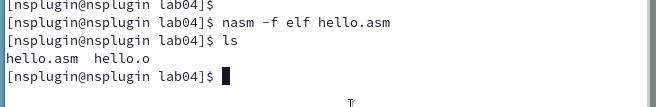


Figure 3: Трансляция программы

### 4.2.1 Расширенный синтаксис командной строки NASM

1. Транслировал файл командой nasm с дополнительными опциями : -o, -g, -l (рис. [[4](#fig:004)])

Опция -o позволяет задать имя объектного файла. Опция -g добавляет отладочную информацию. Опция -l создает файл листинг.

Получился файл листинга list.lst, объектный файл obj.o, в программу добавилась отладочная информация.

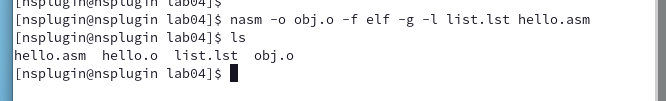


Figure 4: Трансляция программы с дополнительными опциями

## 4.3 Компоновщик LD

1. Выполнил компоновку командой ld и получил исполняемый файл. (рис. [[5](#fig:005)])

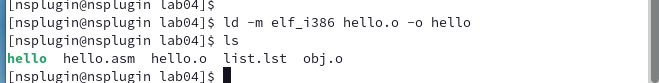


Figure 5: Компоновка программы

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла.

1. Еще раз выполнила компоновку для объектного файла obj.o и получил исполняемый файл main. (рис. [[6](#fig:006)])

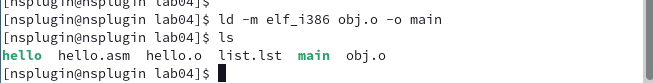


Figure 6: Компоновка второй программы

### 4.3.1 Запуск исполняемого файла

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

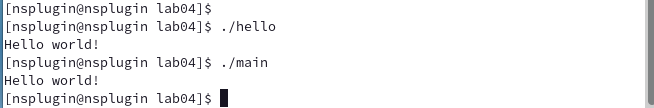


Figure 7: Запуск программ

## 4.4 Задание для самостоятельной работы

1. Скопировал программу в файл lab4.asm. (рис. [[8](#fig:008)])

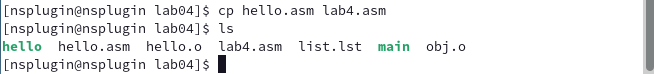


Figure 8: Копирование файла

1. Изменил сообщение Hello world на свое имя. (рис. [[9](#fig:009)])

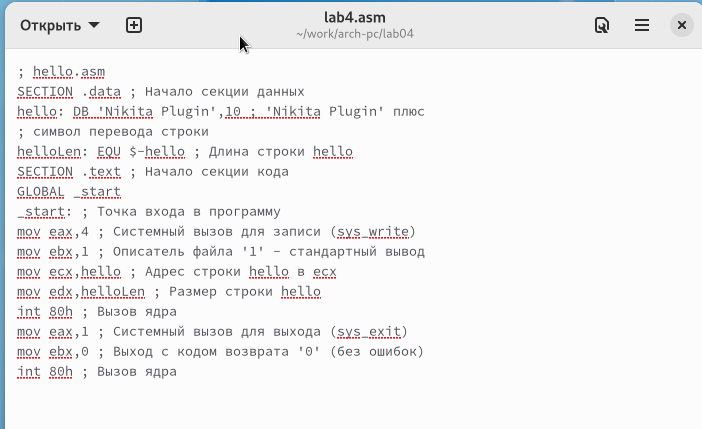


Figure 9: Программа lab4.asm

1. Оттранслировал полученный текст программы lab4.asm в объектный файл. Выполнил компоновку объектного файла и запустил получившийся исполняемый файл. (рис. [[10](#fig:010)])

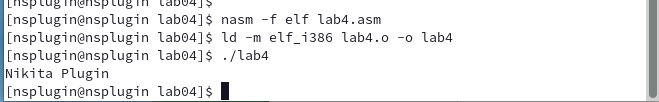


Figure 10: Проверка программы lab4.asm

1. Загрузил файлы на github.

# 5 Выводы

Освоили процесс компиляции и сборки программ, написанных на ассемблере nasm.

# 6 Источники

1. [Архитектура ЭВМ](https://esystem.rudn.ru/course/view.php?id=108)