Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров

Мухина Ксения Николаевна

Содержание

# 1. Цель работы

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

# 2. Выполнение лабораторной работы

Далее описываемая работа была выполнена на виртуальной машине Oracle VirtualBox с ОС Linux Ubuntu.

1. Программа Hello world!

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

Для этого создадим каталог для работы с будущей программой и перейдём в него, создадим текстовый файл ‘hello.asm’ и отредактируем его при помощи gedit.

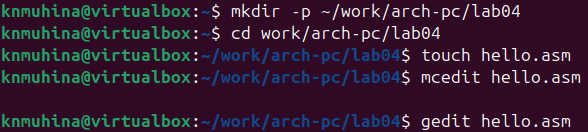


Рис. 1. Создание каталога и файла для будущей программмы.

Наш код будет выглядеть так.

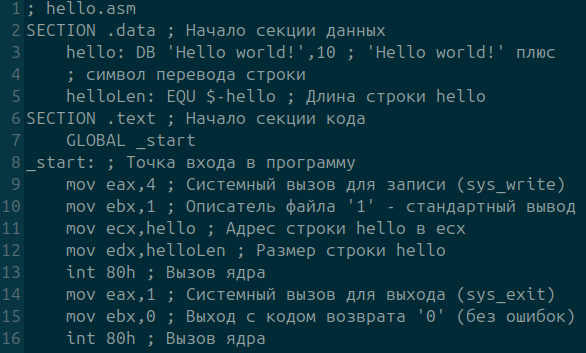


Рис. 2. Код hello.

Данный код взят из указаний по выполнению лабораторной работы; комментарии к коду указаны после символа ‘;’. Обратим внимание, что каждая команда располагается строго на отдельной строке. Также синтаксис ассемблера чувствителен к регистру, т.е. есть разница между большими и малыми буквами.

1. Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции кода программы используем команду nasm. После этого при помощи ls убедимся в успешности компиляции.

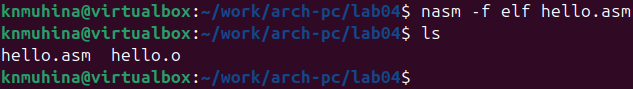


Рис. 3. Компиляция текста Hello world!

Текст был записан в виде объектного кода в созданный объектный файл под названием ‘hello.o’. Отметим, что ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. NASM всегда создаёт выходные файлы в текущем каталоге: в данном случае это наш каталог lab04.

1. Расширенный синтаксис командной строки NASM

Полный синтаксис команды nasm выглядит следующим образом:

nasm [-@ косвенный\_файл\_настроек] [-o объектный\_файл] [-f формат\_объектного\_файла] [-l листинг] [параметры…] [–] исходный\_файл

Выполним следующую команду и сразу же проверим её результат при помощи ls:

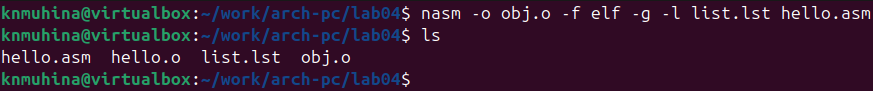


Рис. 4. Компиляция текста в файл obj.o и создание листинга list.lst.

Данная команда скомпилировала hello.asm в obj.o (-o позволила задать имя объектному файлу), при этом формат выходного файла будет elf, в который включены символы для отладки благодаря опции -g. Помимо этого был создан файл листинга list.lst при помощи опции -l.

1. Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику (команда ld).

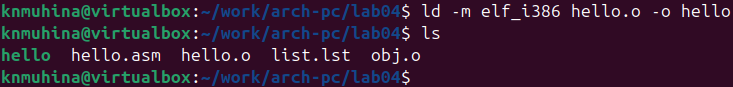


Рис. 5. Создание исполняемого файла hello.

Файл был успешно обработан, и в результате была получена программа hello.

Компоновщик не предполагает по умолчанию расширений для файлов, но принято использовать следующие:

* o: для объектных файлов;
* без расширения: для исполняемых файлов;
* map: для файлов схемы программы;
* lib: для библиотек.

Так же, как и в команде nasm, в команде ld при помощи ключа -o можно задать имя создаваемому файлу.

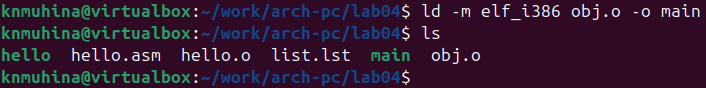


Рис. 6. Создание исполняемого файла с заданным именем.

В результате была собрана программа main из объектного файла obj.o.

4.1. Запуск исполняемого файла

Запустим созданный исполняемый файл hello.

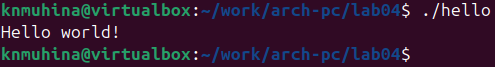


Рис. 7. Запуск исполняемого файла hello.

Терминал вывел текст ‘Hello world!’, что показывает успешное исполнение.

# 3. Задания для самостоятельной работы

Создадим копию файла hello.asm с именем lab4.asm и в соответствии с заданием внесём в него изменения с помощью gedit.

Теперь наш код будет выглядеть так:

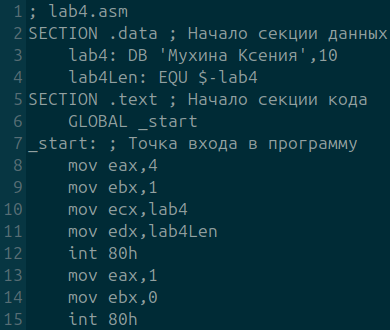


Рис. 8. Код lab4.

Оттранслируем полученный текст в объектный файл, выполним компоновку файла и запустим получившийся исполняемый файл.

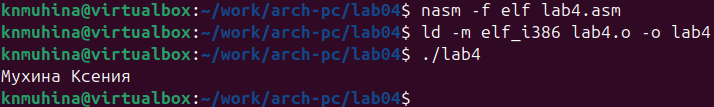


Рис. 9. Запуск программы lab4.

Загрузим файлы hello.asm и lab4.asm на GitHub.

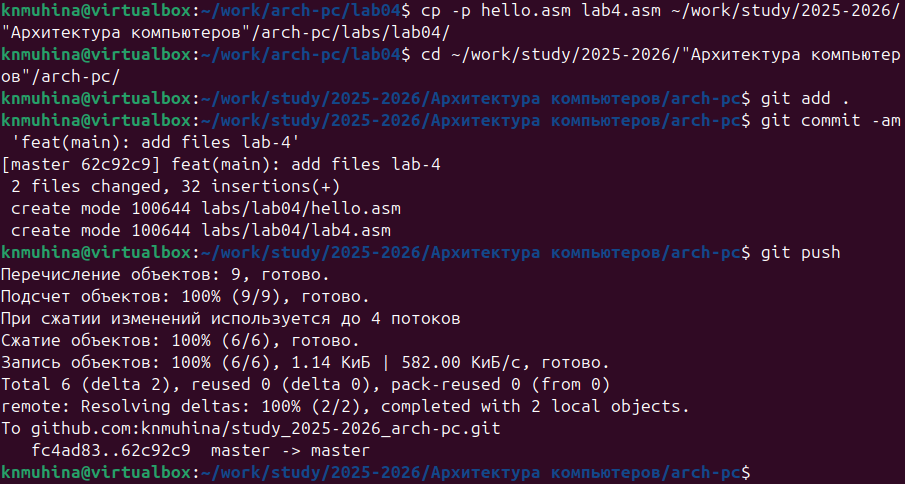


Рис. 10. Загрузка файлов на GitHub.

# 4. Выводы

В ходе выполнения лабораторной работы мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы

1. Файл [“Лабораторная работа №4. Создание и процесс обработки программ на языке ассемблера NASM.pdf”](https://esystem.rudn.ru/mod/resource/view.php?id=1030552)