Лабораторная работа №5

Дисциплина: Архитектура компьютера

Кондратьев Арсений Вячеславович

03.10.2022

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

# 2 Выполнение лабораторной работы

1. Создал текстовый файл с именем hello.asm и ввел в него данный текст(рис.1)

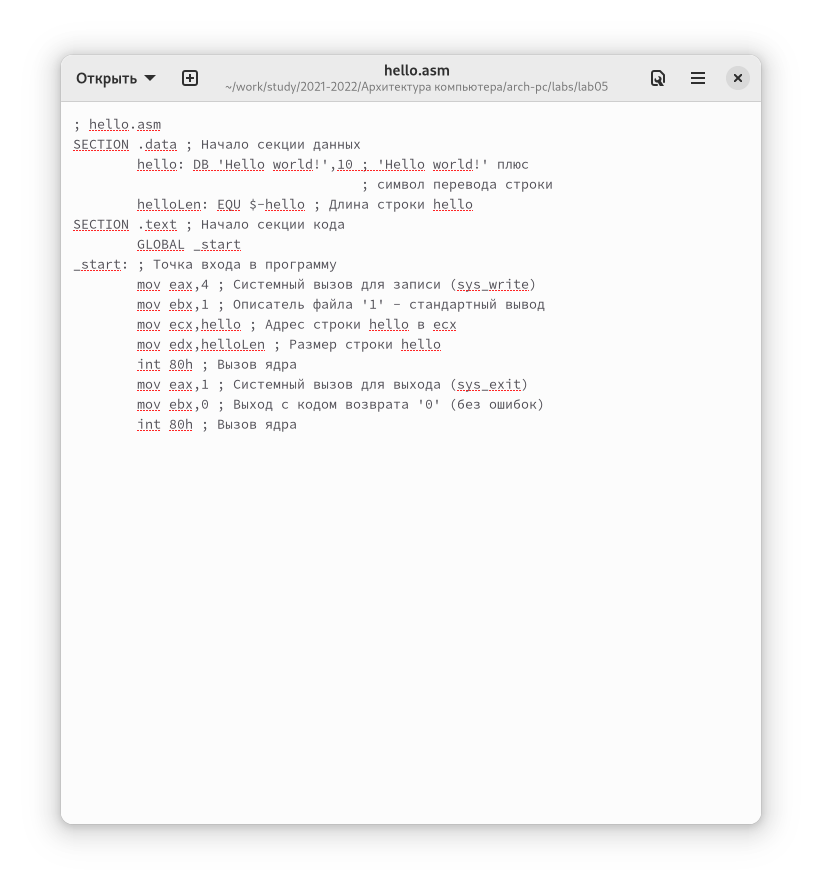


Figure 1: Рис. 1

1. Скомпилировал приведённый выше текст программы «Hello World»(рис.2)

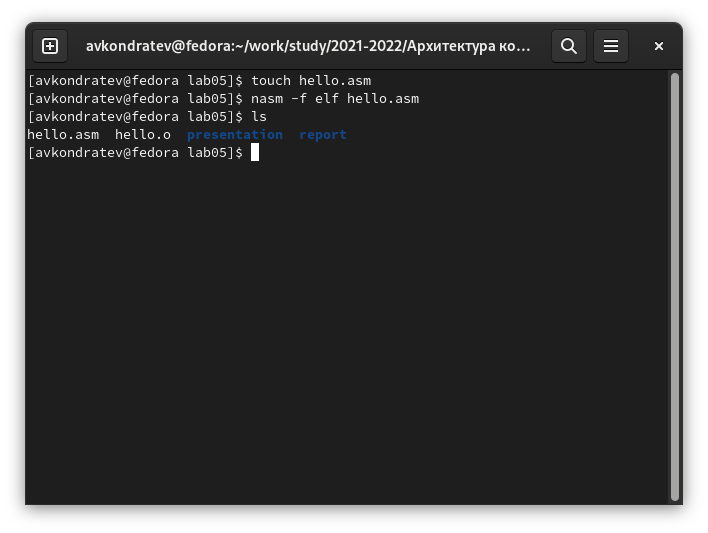


Figure 2: Рис. 2

1. Скомпилировал исходный файл hello.asm в obj.o и создал файл листинга list.lst(рис.3)

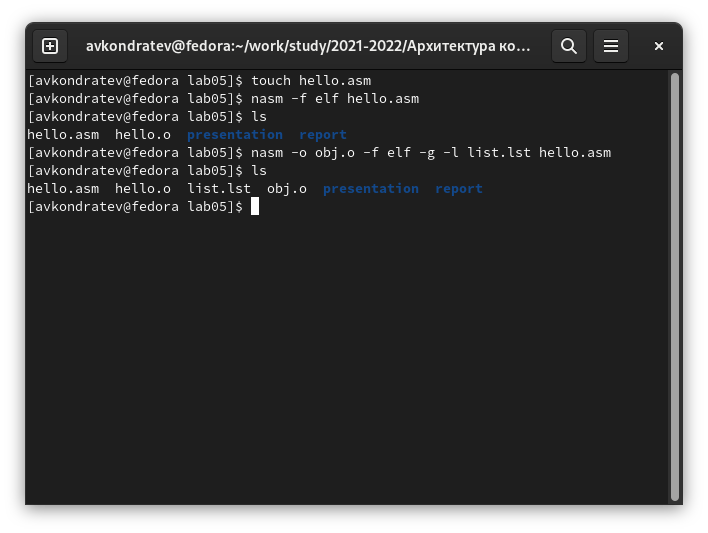


Figure 3: Рис. 3

1. Получил исполняемую программу(рис.4)

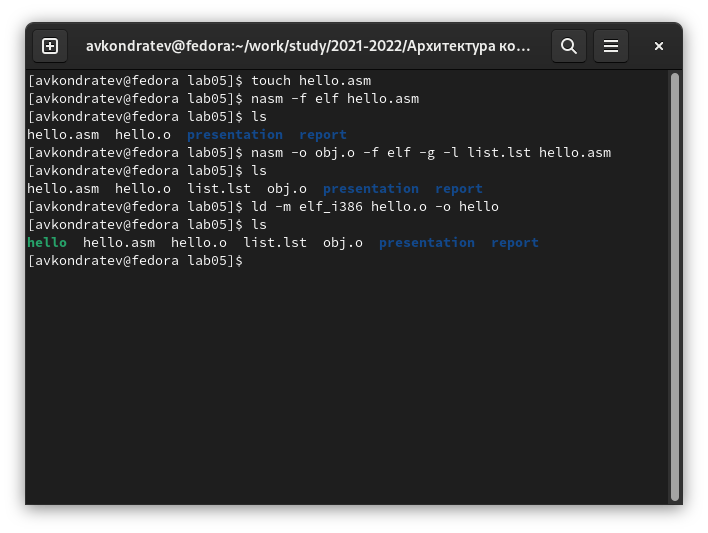


Figure 4: Рис. 4

1. Получил исполняемую программу, указав имя main.(рис.5)

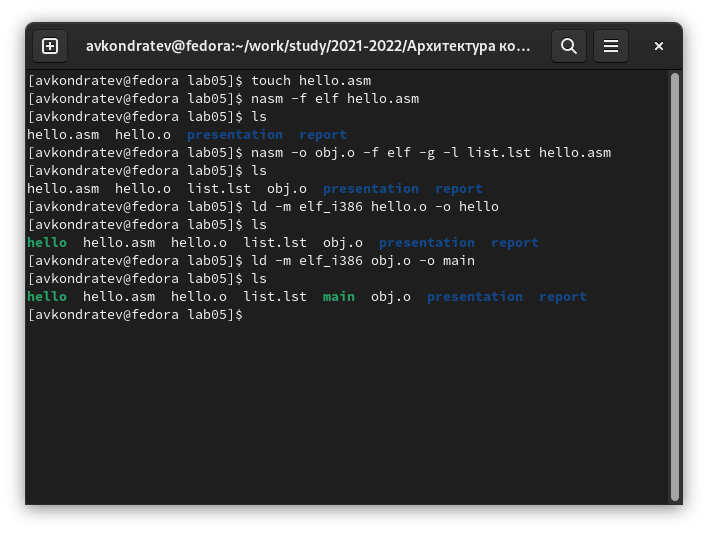


Figure 5: Рис. 5

1. Запустил на выполнение созданный исполняемый файл(рис.6)

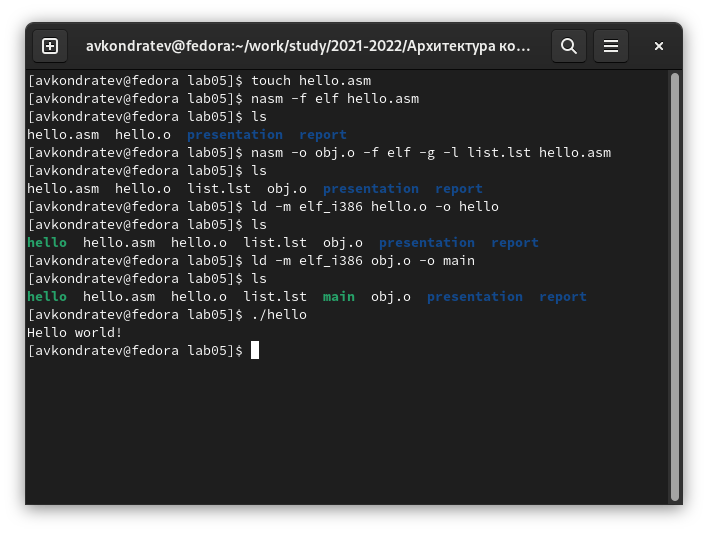


Figure 6: Рис. 6

1. Создал копию с именем lab5 и изменил код, чтобы выводились мое имя и фамилия(рис.7)

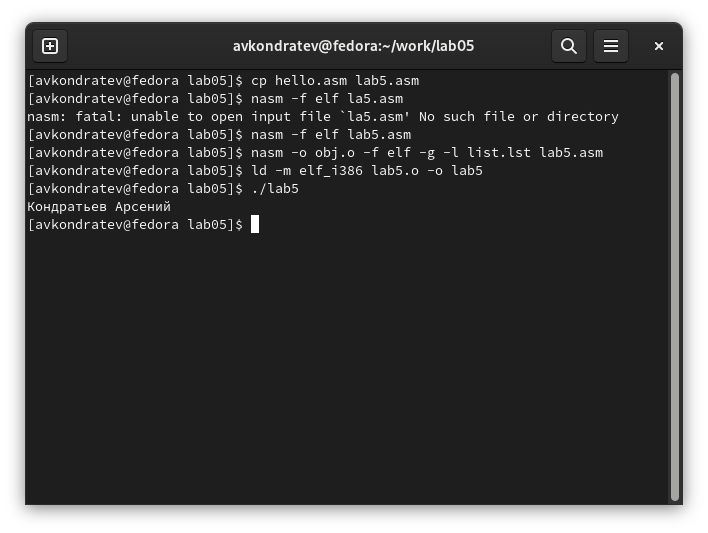


Figure 7: Рис. 7

# 3 Выводы

Я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM

# 4 Контрольные вопросы

1. 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
2. директивы не переводящиеся непосредственно в машинные команды, а управляющие работой транслятора
3. Набор текста программы в текстовом редакторе и сохранение её в отдельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип asm.

* Трансляция — преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst.
* Компоновка или линковка — этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map.
* Запуск программы. Конечной целью является работоспособный исполняемый файл. Ошибки на предыдущих этапах могут привести к некорректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При нахождении ошибки необходимо провести коррекцию программы, начиная с первого шага

1. набор текста программы, трансляция, компановка
2. преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором
3. этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл
4. NASM не запускают без параметров. 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
5. nasm - .o; ld - исполняемая программа