Лабораторная работа №4

Создание и процесс обработки программ на языке ассемблера NASM

Яковлева Дарья Сергеевна

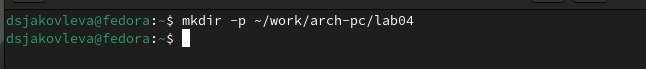
Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

# 2 Выполнение лабораторной работы

Создадим нужную директорию с помощью команды mkdir (Рис. 2.1):



Создание каталога lab4

Теперь переместимся в созданный нами каталог (Рис. 2.2):

Перемещение в созданный каталог

Перемещение в созданный каталог

Создадим файл hello с расширением .asm, в котором мы будем писать код на ассемблере (Рис. 2.3):

Создание .asm файлы

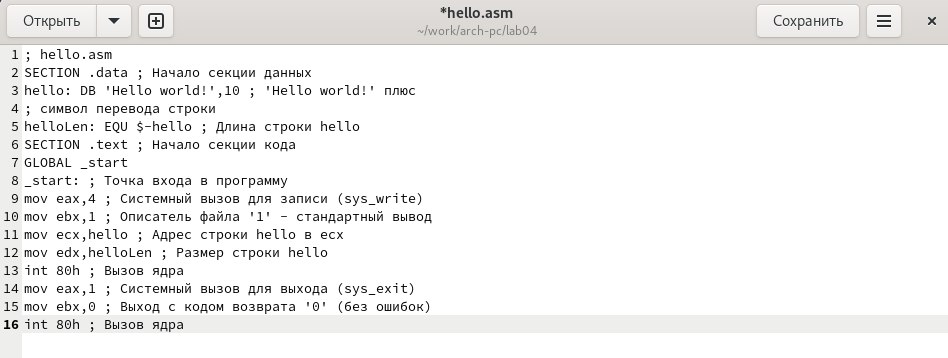
Создание .asm файлы

Для того, чтобы редактировать созданный файл, воспользуемся текстовым редактором gedit (Рис. 2.4):

Открытие созданного файла с помощью gedit

Открытие созданного файла с помощью gedit

Вставим в открытый файл следующий код (Рис. 2.5):



Редактирование файла

Теперь нам необходимо превратить наш файл в объектный. Этим занимается транслятор NASM. Введём следующую команду (Рис. 2.6):

Компиляция файла с помощью nasm

Компиляция файла с помощью nasm

Далее проверим, создался ли объектный файл с помощью команды ls (Рис. 2.7):

Проверка на успешное создание файла

Проверка на успешное создание файла

Теперь попробуем использовать полный вариант команды NASM (Рис. 2.8):

Использование команды nasm с большим количеством аргументов

Использование команды nasm с большим количеством аргументов

Проверим, создался ли файл с помощью команды ls (Рис. 2.9):

Проверка на успешное создание файлов

Проверка на успешное создание файлов

Для создания исполняемого файла необходимо использовать компоновщик ld, который соберёт объектный файл. Напишем следующую команду (Рис. 2.10):

Сборка исполняемого файла с помощью ld

Сборка исполняемого файла с помощью ld

Проверим, создался ли файл с помощью команды ls (Рис. 2.11):

Проверка на успешное создание исполняемого файла

Проверка на успешное создание исполняемого файла

Теперь соберём файл obj.o в файл main (Рис. 2.12):

Сборка исполняемого файла main из файла obj.o

Сборка исполняемого файла main из файла obj.o

Проверим, создался ли файл. Снова пропишем команду ls (Рис. 2.13):

Проверка на успешное создание исполняемого файла

Проверка на успешное создание исполняемого файла

Теперь запустим файл hello, для этого мы должны написать ./ и название файла (Рис. 2.14):

Запуск исполняемого файла

Запуск исполняемого файла

# 3 Выполнение задания для самостоятельной работы

Скопируем файл hello.asm в каталог ~/work/arch-pc/lab04 под названием lab4.asm (Рис. 3.1):

Копирование файла

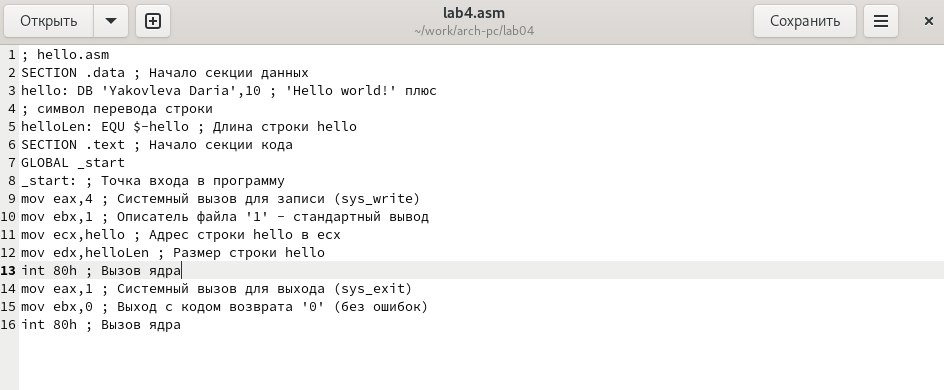
Копирование файла

Внесём изменения в скопированный файл. Для этого откроем его в gedit (Рис. 3.2):

Открытие файла для редактирования

Открытие файла для редактирования

Теперь изменим третью строчку, заменив фразу Hello world! на фамилию и имя (Рис. 3.3):



Процесс редактирования файла

Теперь скомпилируем полученный файл в объектный. Для этого воспользуемся командой nasm и укажем формат elf и нужный файл для компиляции (Рис. 3.4):

Компиляция файла в объектный

Компиляция файла в объектный

Теперь соберём полученный объектный файл. Укажем формат elf\_i386 и объектный файл для сборки (lab4.o). Укажем, что выходной файл должен быть назван lab4 (Рис. 3.5):

Сборка объектного файла в исполняемый

Сборка объектного файла в исполняемый

Убедимся в том, что сделали всё правильно. Для этого запустим собранный файл (Рис. 3.6):

Запуск собранного файла

Запуск собранного файла

Теперь скопируем файл hello.asm в каталог 4 лабораторной работы (Рис. 3.7):

Копирование файла hello.asm в каталог 4 лабораторной работы

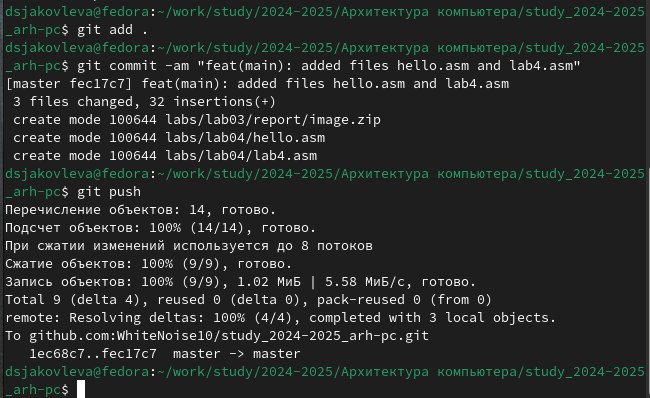
Копирование файла hello.asm в каталог 4 лабораторной работы

Эту же операцию проведём для файла lab4.asm (Рис. 3.8):

Копирование файла lab4.asm в каталог 4 лабораторной работы

Копирование файла lab4.asm в каталог 4 лабораторной работы

Теперь загрузим результат проделанной лабораторной работы на GitHub (Рис. 3.9):



Загрузка проделанной работы на GitHub

# 4 Выводы

В результате выполнения лабораторной работы появилось понимание того, как работает алгоритм создания исполняемого файла из кода на ассемблере, а также появились навыки работы с языком nasm, компиляции кода в объектный файл и сборкой исполняемых программ.