Отчёт по лабораторной работе № 4

Создание и процесс обработки программ на языке ассемблера NASM

Мальянц Виктория Кареновна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Создание программы Hello world!

Создаю каталог для работы с программами на языке ассемблера Nasm с помощью команды mkdir -p (рис. 1).

Рис. 1: Создание каталога

Рис. 1: Создание каталога

Перехожу в созданный каталог с помощью команды cd (рис. 2).

Рис. 2: Переход в созданный каталог

Рис. 2: Переход в созданный каталог

Создаю текстовый файл с именем hello.asm с помощью команды touch (рис. 3).

Рис. 3: Создание текстового файла hello.asm

Рис. 3: Создание текстового файла hello.asm

Открываю этот файл с помощью текстового редактора gedit (рис. 4) (рис. 5).

Рис. 4: Открытие файла hello.asm в текстовом редакторе gedit через терминал

Рис. 4: Открытие файла hello.asm в текстовом редакторе gedit через терминал

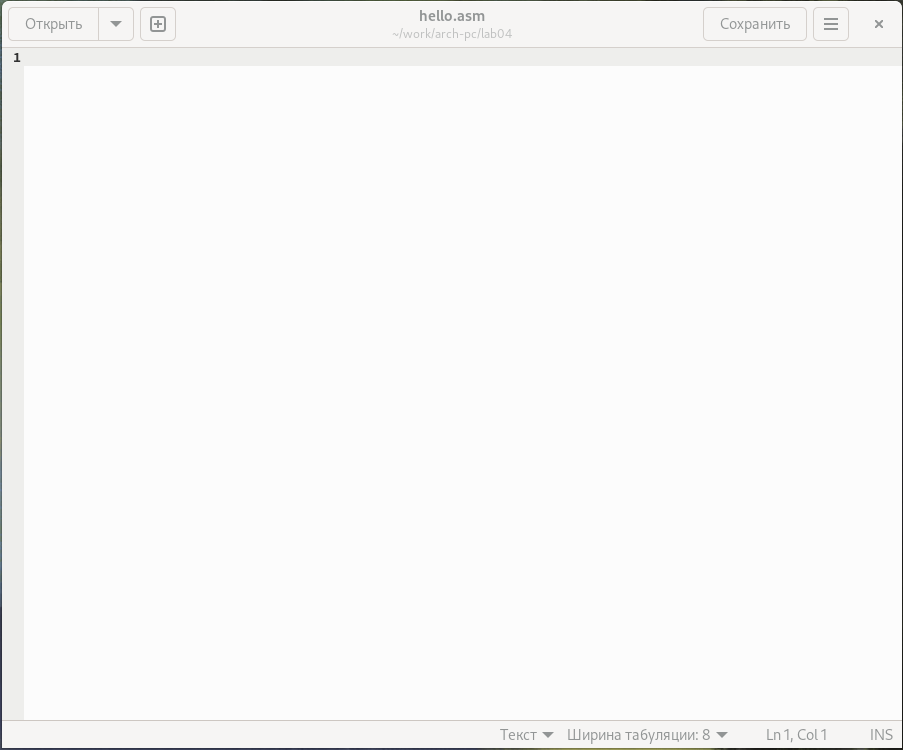


Рис. 5: Окно текстового редактора gedit

Заполняю файл, вставляя в него программу для вывода “Hello World!” (рис. 6).

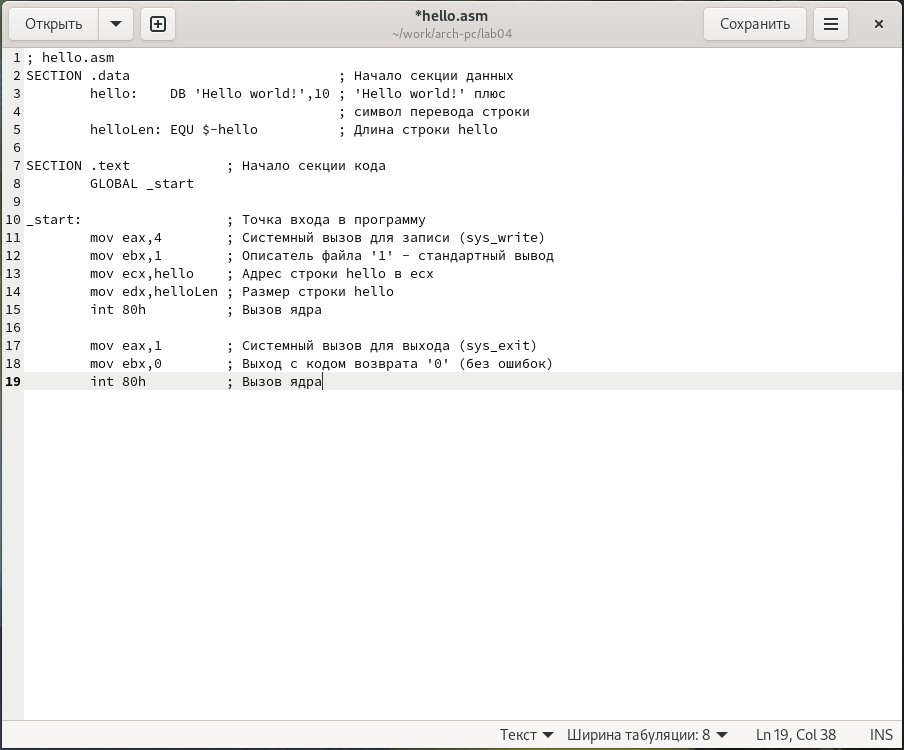


Рис. 6: Заполнение файла

## 3.2 Работа с транслятором NASM

Преобразовываю текст программы “Hello World!” в объектный код с помощью транслятора NASM. Для этого использую команду nasm -f elf hello.asm, ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Проверяю, что объектный файл был создан с помощью команды ls. Объектный файл имеет имя hello.o (рис. 7).

Рис. 7: Компиляция текста программы

Рис. 7: Компиляция текста программы

## 3.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует исходный файл hello.asm в obj.o, при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g) и будет создан файл листинга list.lst (опция -l). С помощью команды ls проверяю, что файлы были созданы (рис. 8).

Рис. 8: Компиляция текста программы

Рис. 8: Компиляция текста программы

## 3.4 Работа с компоновщиком LD

Передаю объектный файл на обработку компоновщику LD, чтобы получить программу hello. Ввожу команду ld -m elf\_i386 obj.o -o main. Ключ -o задает имя создаваемого исполняемого файла. С помощью команды ls проверяю, что файл hello был создан (рис. 9).

Рис. 9: Передача объектного файла на обработку компоновщику

Рис. 9: Передача объектного файла на обработку компоновщику

Выполняю следующую программу (рис. 10). Исполняемый файл будет иметь имя main. Объектный файл, из которого собран исполняемый файл, имеет имя obj.o.

Рис. 10: Передача объектного файла на обработку компоновщику

Рис. 10: Передача объектного файла на обработку компоновщику

## 3.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 11).

Рис. 11: Запуск исполняемого файла

Рис. 11: Запуск исполняемого файла

## 3.6 Выполнение заданий для самостоятельной работы

С помощью команды cp создаю копию файла hello.asm с именем lab4.asm в каталоге ~/work/arch-pc/lab04 (рис. 12).

Рис. 12: Создание копии файла

Рис. 12: Создание копии файла

С помощью текстового редактора gedit открываю файл lab4.asm. В нем вношу изменения в программу так, чтобы вместо Hello world! на экран выводилась строка с моими именем и фамилией (рис. 13) (рис. 14).

Рис. 13: Открытие файла lab4.asm в текстовом редакторе gedit через терминал

Рис. 13: Открытие файла lab4.asm в текстовом редакторе gedit через терминал

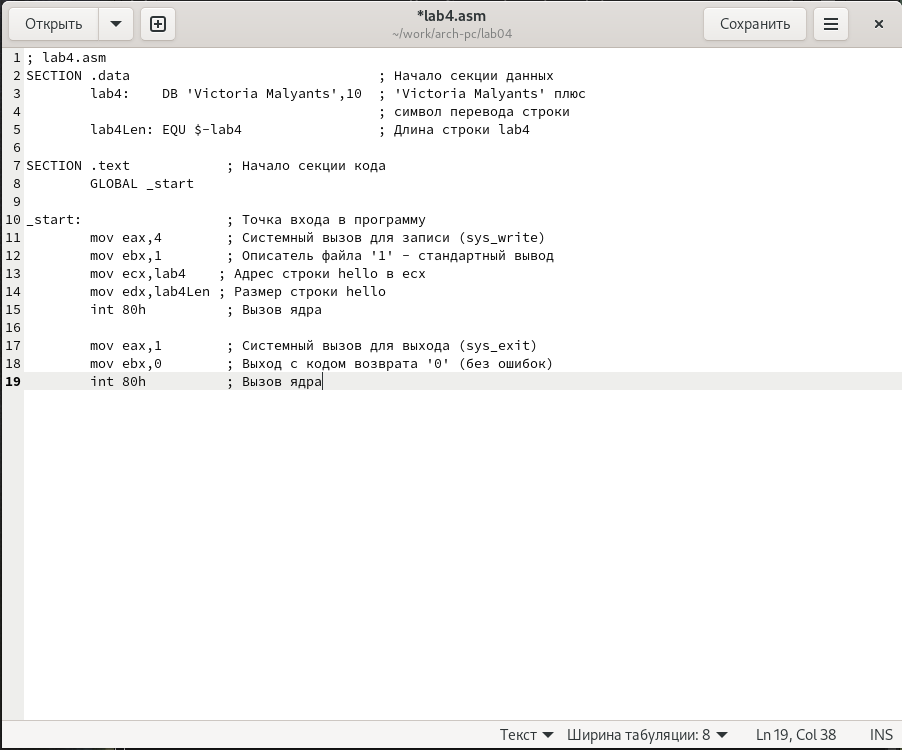


Рис. 14: Изменение программы

Компилирую текст программы в объектный файл. Проверяю с помощью команды ls, что файл lab4.o создан (рис. 15).

Рис. 15: Компиляция текста программы

Рис. 15: Компиляция текста программы

Передаю объектный файл lab4.0 на обработку компоновщику LD, чтобы получить исполняемый файл lab4. С помощью команды ls проверяю, что файл lab4 был создан (рис. 16).

Рис. 16: Компиляция текста программы

Рис. 16: Компиляция текста программы

Запускаю исполняемый файл lab4 (рис. 17).

Рис. 17: Запуск исполняемого файла

Рис. 17: Запуск исполняемого файла

Копирую файл hello.asm в каталог ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 18).

Рис. 18: Создание копии файла в каталоге

Рис. 18: Создание копии файла в каталоге

Копирую файл lab4.asm в каталог ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 19).

Рис. 19: Создание копии файла в каталоге

Рис. 19: Создание копии файла в каталоге

Перехожу в каталог ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис. 20).

Рис. 20: Переход в каталог

Рис. 20: Переход в каталог

С помощью команды ls убеждаюсь в том, что файлы hello.asm и lab4.asm скопированы в каталог (рис. 21).

Рис. 21: Файлы hello.asm и lab4.asm в каталоге

Рис. 21: Файлы hello.asm и lab4.asm в каталоге

Добавляю изменения на Github с помощью команды git add . (рис. 22).

Рис. 22: Добавление файлов

Рис. 22: Добавление файлов

Сохраняю изменения на Github с помощью команды git commit -m (рис. 23).

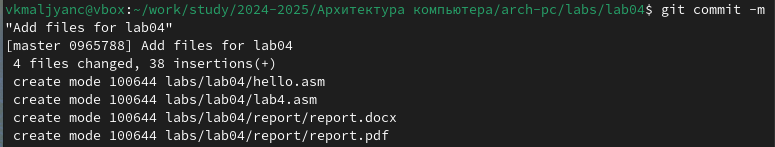


Рис. 23: Сохранение изменений

Отправляю все произведенные изменения локального дерева в центральный репозиторий (рис. 24).

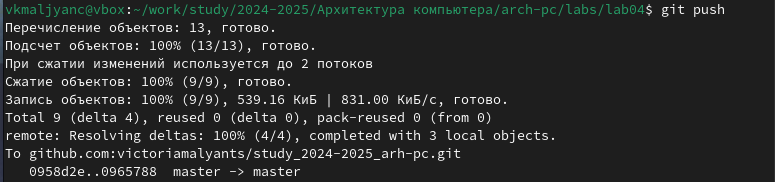


Рис. 24: Отправка файлов

# 4 Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.