Лабораторная работа No4

Архитектура компьютера

Оушен Мухаммад Ламин

Содержание

# Цель работы

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

# Задание

**1.** В каталоге ~/work/arch-pc/lab04 с помощью команды cp создать копию файла hello.asm с именем lab4.asm **2.** С помощью любого текстового редактора внести изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем. **3.** Оттранслировать полученный текст программы lab4.asm в объектный файл. Выполнить компоновку объектного файла и запустить получившийся иполняемый файл. **4.** Скопировать файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузить файлы на Github

# Теоретическое введение

# Вопросы для самопроверки :

1. **Какие основные отличия ассемблерных программ от программ на языках высокого уровня?**

* Ассемблерные программы ближе к машинному коду и требуют более детального управления оборудованием. В отличие от языков высокого уровня, таких как Python или Java, которые предоставляют абстракции и автоматизацию, ассемблерные программы требуют явного указания каждой команды и обращения к памяти. **2. В чём состоит отличие инструкции от директивы на языке ассемблера?**
* Инструкции являются командами, которые процессор выполняет(например, MOV , ADD), тогда как директивы не преобразуются в машинный код, а служат для управления компилятором(например, .data, .text ). **3. Перечислите основные правила оформления программ на языке ассемблера.**
* Каждая команда должна располагаться на отдельной строке.
* Синтаксис чувствителен к регистру, т.е. MOV и mov будут восприниматься как разные команды.
* Метки должны начинаться с буквы, знака подчеркивания или точки.
* Комментарии начинаются с ; и продолжаются до конца строки.

**4. Каковы этапы получения исполняемого файла?** - Набор текста программы в текстовом редакторе и сохранение в файл с расширением .asm. - Трансляция исходного файла в объектный код с помощью транс-лятора (например, NASM). - Компоновка объектного файла в исполняемый файл с помощью компоновщика (например, LD). - Запуск получившегося исполняемого файла. **5. Каково назначение этапа трансляции?** - Этап трансляции преобразует текст программы, написанной на ассемблере, в объектный код, который может быть использован для создания исполняемого файла. На этом этапе проверяются синтаксические ошибки и создаются объектные файлы. **6. Каково назначение этапа компоновки?** - Этап компоновки объединяет один или несколько объектных файлов, а также библиотеки в единый исполняемый файл. Компоновщик разрешает внешние ссылки и размещает код в нужных адресах памяти.

**7. Какие файлы могут создаваться при трансляции программы, какие из них создаются по умолчанию?** - При трансляции могут создаваться объектные файлы (обычно с расширением .o), файлы листинга (с расширением .lst), а также файлы с отладочной информацией. По умолчанию создается только объектный файл. **8. Каковы форматы файлов для NASM и LD?** - Для NASM: форматы могут включать elf, elf64, bin и другие, в зависимости от архитектуры и операционной системы. - Для LD: форматы включают elf\_i386 , elf\_x86\_64 и другие. # Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM.

Рис 4.1: команда mkdir

Рис 4.1: команда mkdir

Перейдем в созданный каталог.

Рис 4.2: переход в каталог

Рис 4.2: переход в каталог

Создадим текстовый файл с именем hello.asm.

Рис 4.3: создание текстового файла

Рис 4.3: создание текстового файла

Откроем этот файл с помощью любого текстового редактора, например, gedit.

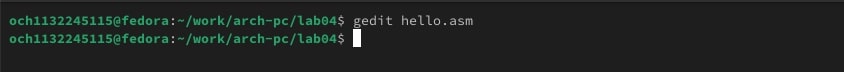


Рис 4.4: редактор gedit

Введем в него следующий текст



Рис 4.5: введение текста

Для компиляции приведённого выше текста программы «Hello World» необходимо написать (nasm -f elf hello.asm). Проверяем наличие нужных файлов с помощью команды ls.

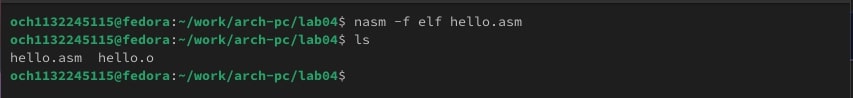


Рис 4.6: компиляция текста

Выполняем следующую команду (nasm -o obj.o -f elf -g -l list.lst hello.asm). Также проверяем наличие необходимых файлов.

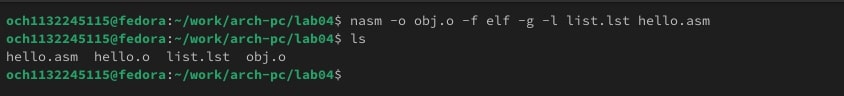


Рис 4.7: компиляция файла

Объектный файл необходимо передать на обработку компоновщику следующим образом.

Рис 4.8: паредача файла на обработку

Рис 4.8: паредача файла на обработку

Выполняем следующую команду (ld -m elf\_i386 obj.o -o main).

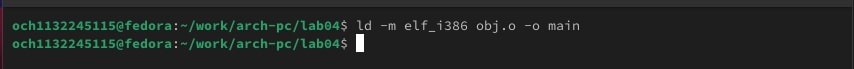


Рис 4.9: команда ld -m elf\_i386 obj.o -o main

Запустим на выполнение созданный исполняемый файл.

Рис 4.10: итог запуска

Рис 4.10: итог запуска

с помощью команды cp создайте копию файла.

Рис 4.11: hello.asm с именем lab4.asm

Рис 4.11: hello.asm с именем lab4.asm

запишем теперь свою фимилию и имя.



Рис 4.12: фимилию и имя

Запустим получившийся исполняемый файл.

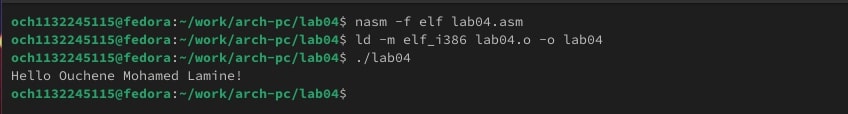


Рис 4.13: второй запуск и итог

# Выводы

Был освоен процесс компиляции и сборки программ, написанных на ассемблере NASM.