Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Наговицын Арсений Владимирович

Содержание

# 1 Цель работы

Цель данной работы: освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задание

1. Программа 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. Задание для самостоятельной работы.

# 3 Теоретическое введение

Основными функциональными элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Эти устройства взаимодействуют друг с другом через общую шину, которая соединяет их. В современных компьютерах проводники шины представлены в виде электропроводящих дорожек на материнской плате. Центральный процессор выполняет обработку информации и координирует работу всех узлов компьютера. Он состоит из нескольких устройств, включая арифметико-логическое устройство (АЛУ), которое выполняет логические и арифметические операции над данными, и устройство управления (УУ), которое обеспечивает управление и контроль всех устройств компьютера. Также в состав процессора входят регистры - сверхбыстрая оперативная память небольшого объема, используемая для хранения промежуточных результатов выполнения инструкций.

Для написания программ на ассемблере необходимо знать, какие регистры процессора существуют и как их использовать. Примерами основных регистров общего назначения в архитектуре x86 являются RAX, RCX, RDX, RBX, RSI, RDI (64-битные), EAX, ECX, EDX, EBX, ESI, EDI (32-битные), AX, CX, DX, BX, SI, DI (16-битные) и AH, AL, CH, CL, DH, DL, BH, BL (8-битные). Оперативное запоминающее устройство (ОЗУ) является важным элементом ЭВМ. Оно предназначено для хранения программ и данных, с которыми процессор работает в текущий момент. ОЗУ состоит из ячеек памяти с уникальными номерами, которые служат адресами для хранящихся данных.

Периферийные устройства включают устройства внешней памяти, предназначенные для хранения больших объемов данных, и устройства ввода-вывода, которые обеспечивают взаимодействие с внешней средой.

Вычислительный процесс ЭВМ основан на программном управлении, где задача решается последовательностью действий, записанных в программе. Коды команд представляют собой двоичные комбинации, где операционная часть содержит код выполняемой команды, а адресная часть содержит данные или адреса данных, необходимых для выполнения операции. При выполнении команды процессор выполняет командный цикл, который включает формирование адреса команды, считывание кода команды, выполнение команды и переход к следующей команде.

Язык ассемблера (assembly language) является машинно-ориентированным языком низкого уровня. NASM (Netwide Assembler) - это проект ассемблера, который поддерживает Intel-синтаксис и инструкции x86-64. NASM доступен для различных операционных систем и позволяет получать объектные файлы для этих систем.

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Перехожу в каталог, с помощью утилиты cd, и создаю папку lab04 (рис. 4.1).

fig:
Создание каталога

Рис. 4.1: Создание каталога

Создаю в текущем каталоге файл и открываю файл в текстовом редакторе (рис. 4.2).

fig:
Создание файла

Рис. 4.2: Создание файла

Заполняю файл, вставляя в него программу для вывода “Hello world!” (рис. 4.3).

fig:
Заполнение файла

Рис. 4.3: Заполнение файла

## 4.2 Работа с транслятором NASM.

Превращаю текст программы в объектный код с помощью транслятора NASM (рис. 4.4).

fig:
Компиляция текста программы

Рис. 4.4: Компиляция текста программы

## 4.3 Работа с расширенным синтаксисом командной строки NASM.

Ввожу команду, которая скомпилирует файл Hello.asm в файл obj.o (рис. 4.5).

fig:
Компиляция текста программы

Рис. 4.5: Компиляция текста программы

## 4.4 Работа с компоновщиком LD.

Передаю объектный файл hello.o на обработку компоновщику ld (рис. 4.6).

fig:
Компиляция текста программы

Рис. 4.6: Компиляция текста программы

## 4.5 Запуск исполняемого файла.

Запускаю на выполнение созданный исполняемый файл hello (рис. 4.7).

fig:
Запуск исполняемого файла

Рис. 4.7: Запуск исполняемого файла

## 4.6 Задание для самостоятельной работы.

Создаю в текущем каталоге копию файла hello.asm с именем lab4.asm и запускаю в текстовом редакторе (рис. 4.8).

fig:
Создание копии файла

Рис. 4.8: Создание копии файла

Вношу изменения в программу, чтобы она выводила мое имя и фамилию (рис. 4.9).

fig:
Изменение программы

Рис. 4.9: Изменение программы

Компилирую текст программы в объектный код с помощью транслятора NASM (рис. 4.10).

fig:
Компиляция текста программы

Рис. 4.10: Компиляция текста программы

Ввожу команду, которая скомпилирует файл lab4.asm в файл lab4.o (рис. 4.11).

fig:
Компиляция текста программы

Рис. 4.11: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD (рис. 4.12).

fig:
Компиляция текста программы

Рис. 4.12: Компиляция текста программы

Запускаю на выполнение созданный исполняемый файл lab4 (рис. 4.13).

fig:
Запуск исполняемого файла

Рис. 4.13: Запуск исполняемого файла

Копирую файлы hello.asm и lab4.asm в каталог (рис. 4.14).

fig:
Копирование файлов

Рис. 4.14: Копирование файлов

Проверяю нахождение файлов в каталоге (рис. 4.15).

fig:
Файлы в каталоге

Рис. 4.15: Файлы в каталоге

Добавляю файлы на GitHub (рис. 4.16).

fig:
Файлы в каталоге

Рис. 4.16: Файлы в каталоге

Загружаю на сервера GitHub (рис. 4.17).

fig:
Файлы в каталоге

Рис. 4.17: Файлы в каталоге

# 5 Выводы

При выполнении данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.