Отчёт по лабораторной работе №5.

Дисциплина: архитектура компьютера

Тимофеева Екатерина Николаевна

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассем- блере 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 Теоретическое введение

Основными функциональными элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора входят следующие устройства:

1. арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти;  
2. устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера;  
3. регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в каче- стве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические операции) данных хранящихся в регистрах. Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита. В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего используются при написании программ):  
4. RAX, RCX, RDX, RBX, RSI, RDI — 64-битные  
5. EAX, ECX, EDX, EBX, ESI, EDI — 32-битные  
6. AX, CX, DX, BX, SI, DI — 16-битные  
7. AH, AL, CH, CL, DH, DL, BH, BL — 8-битные  
  
Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных.

# 4 Выполнение лабораторной работы

Создаём каталог для работы с программами на языке ассемблера NASM, переходим в созданный каталог, создаём в текущем каталоге пустой текстовый файл hello.asm, открываем созданный файл в текстовом редакторе. (рис. 1)

![Рис. 1: Создание текстового файла](data:application/octet-stream;base64,)

Рис. 1: Создание текстового файла

Введём в него текст.(рис. 2)

![Рис. 2: Написание текста](data:application/octet-stream;base64,)

Рис. 2: Написание текста

Превращаем текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM. Далее проверяем правильность выполнения команды с помощью утилиты ls; действительно, создан файл.(рис. 3)

![Рис. 3: Переход от текста к объектному коду](data:application/octet-stream;base64,)

Рис. 3: Переход от текста к объектному коду

Вводим команду, которая скомпилирует файл hello.asm в файл obj.o при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа будет создвн файл листинга. Проверяем правильность выполнения команды.(рис. 4)

![Рис. 4: Компиляция файла](data:application/octet-stream;base64,)

Рис. 4: Компиляция файла

Передаём объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл. Ключ -о задает имя создаваемого исполняемого файла. Далее проверяем с помощью утилиты ls правильность выполнения команды.(рис. 5)

![Рис. 5: Получение исполняемого файла](data:application/octet-stream;base64,)

Рис. 5: Получение исполняемого файла

Выполняем следующую команду и запускаем созданный исполняемый файл.(рис. 6)

![Рис. 6: Запуск исполняемого файла](data:application/octet-stream;base64,)

Рис. 6: Запуск исполняемого файла

# 5 Задания для самостоятельной работы

Создаём в текущем каталоге копию файла hello.asm с именем lab5.asm (рис. 7)

![Рис. 7: Создание копии](data:application/octet-stream;base64,)

Рис. 7: Создание копии

С помощью текстового редактора открываем файл lab5.asm и вносим изменения в программу так, чтобы она выводила мои имя и фамилию.(рис. 8)

![Рис. 8: Внесение изменений](data:application/octet-stream;base64,)

Рис. 8: Внесение изменений

Компилируем текст программы в объектный файл, передаём объектный файл lab5.o на обработку компоновщику LD, чтобы получить исполняемый файл lab5 и запускаем исполняемый файл.(рис. 9)

![Рис. 9: Запуск файла](data:application/octet-stream;base64,)

Рис. 9: Запуск файла

Добавляем файлы на GitHub.

# 6 Выводы

Здесь кратко описываются итоги проделанной работы.

# Список литературы

Демидова А.В. “Лабораторная работ №5” - методический материал