Шаблон отчёта по лабораторной работе №4

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

Содержание

# 1 Цель работы

Освоить процедуру компиляции и сборки программ 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 иполняемого 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: - арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; - устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; - 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): - RAX, RCX, RDX, RBX, RSI, RDI — 64-битные - EAX, ECX, EDX, EBX, ESI, EDI — 32-битные - AX, CX, DX, BX, SI, DI — 16-битные - 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. Периферийные устройства в составе ЭВМ: - устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных. - устройства ввода-вывода, которые обеспечивают взаимодействие ЦП с внешней средой. В основе вычислительного процесса ЭВМ лежит принцип программного управ- ления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы. Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса данных, которые участвуют в выполнении данной операции. При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. Он заключается в следующем: 1. формирование адреса в памяти очередной команды; 2. считывание кода команды из памяти и её дешифрация; 3. выполнение команды; 4. переход к следующей команде. Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.

# 4 Выполнение лабораторной работы

## 4.1 Создание программы Hello World!

Создаю каталог для лабораторной работы №4

Figure 1: Создание каталога

Figure 1: Создание каталога

Перемещаюсь в каталог, в котором буду работать

Figure 2: Перемещение между директориями

Figure 2: Перемещение между директориями

Создаю пустой текстовый файл hello.asm

Figure 3: Создание файла

Figure 3: Создание файла

Открываю созданный файл в текстовом редакторе gedit

Figure 4: Открытие файла в текстовом редакторе

Figure 4: Открытие файла в текстовом редакторе

Заполняю файл, вставляя в него программу для вывода “Hello world!”

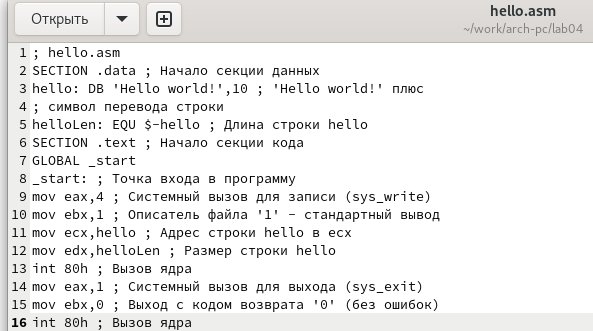


Figure 5: Заполнение файла

## 4.2 Работа с транслятором NASM

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM и проверяю правильность выполнения команды

Figure 6: Компиляция текста программы

Figure 6: Компиляция текста программы

## 4.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду для компиляции файла hello.asm в файл obj.o и проверяю правильность выполнения программы

Figure 7: Компиляция текста программы

Figure 7: Компиляция текста программы

## 4.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello, и проверяю правильность выполнения команды

Figure 8: Передача файла на обработку компоновщику

Figure 8: Передача файла на обработку компоновщику

Ввожу необходимую команду и проверяю правильность её выполнения

Figure 9: Передача файла на обработку компоновщику

Figure 9: Передача файла на обработку компоновщику

## 4.5 Запуск исполняемого файла

Запускаю на выполнение созаднный файл hello

Figure 10: Запуск исполняемого файла

Figure 10: Запуск исполняемого файла

## 4.6 Выполнение заданий для самостоятельной работы

Копирую файл hello.asm с новым именем lab4.asm и проверяю правильность копирования

Figure 11: Создании копии файла

Figure 11: Создании копии файла

С помощью текстового редактора gedit открываю файл lab4.asm и вношу изменения, чтобы программа выводила мои имя и фамилию

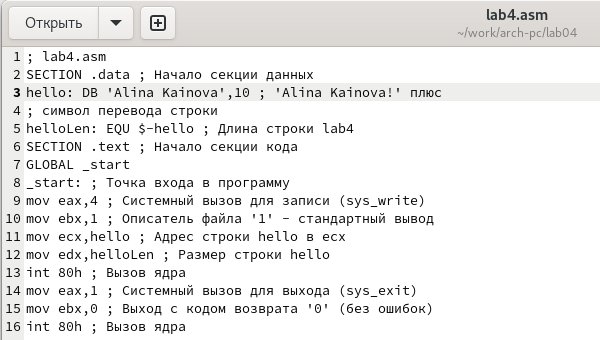


Figure 12: Изменение файла

Компилирую файл lab4.asm в объектный файл lab4.o и проверяю правильность компиляции

Figure 13: Компиляция текста программы

Figure 13: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD

Figure 14: Передача файла на обработку компоновщику

Figure 14: Передача файла на обработку компоновщику

Запускаю исполняемый файл lab4

Figure 15: Запуск исполняемого файла

Figure 15: Запуск исполняемого файла

Копирую файлы hello.asm и lab4.asm в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04

Figure 16: Копирование файлов в нужный каталог

Figure 16: Копирование файлов в нужный каталог

Проверяю правильность копирования файлов

Figure 17: Проверка копирования

Figure 17: Проверка копирования

Добавляю изменения на GitHub

Figure 18: Добавление файлов

Figure 18: Добавление файлов

Сохраняю добавленные изменения

Figure 19: Сохранение изменений

Figure 19: Сохранение изменений

Отправляю файлы на сервер

Figure 20: Отправка файлов

Figure 20: Отправка файлов

# 5 Выводы

В ходе данной лабораторной работы мы научились компилировать програмы и выполнять сборку программ, написанных на ассемблере NASM.

# Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089084/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%964.%20%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B8%20%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM.pdf