Лабораторная работа№4

Дисциплина:Архитектура компьютера

Тойчубекова Асель Нурлановна

Содержание

# 1 Цель работы

Целью работы является освоение процедуры компиляции и сбоки программ, написанных на ассемблере NASM.

# 2 Задание

* Понять основные принципы работы компьютера:

-работа АЛУ;  
-устройство управления;  
-работа регистров;  
-устройство внешней памяти;  
-устройства ввода-вывода.

* Познакомиться с ассемблером и языком ассемблера:

-как создавать и обрабатывать программы на языке ассемблер;  -общие понятия о языке ассемблера NASM;  
-работа транслятора NASM;  
-работа командной строки NASM;  
-работа комповщика;  
-как запустить исполняемый файл.

* Написать программу для вывода на экран “Hello world”.
* Задание для самостоятельной работы:

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 исполняемый файл.
4. Скопировать файлы hello.asm и lab4.asm в наш локальный репозиторий в ката- лог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузить файлы на Github.

# 3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства, их взаимодействие осуществляется через общую шину, к которой они подключены. В состав центрального процессора входят:

* арифметико-логическое устройство(АЛУ);
* устройство управления(уу);
* регистры.

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое на- прямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент.

В состав ЭВМ также вхождят периферийные устройства:

* устройство внешней памяти;
* устройства ввода-вывода;

В основе вычислительного процесса ЭВМ лежит принцип программного управления, то компьютер решает задачу последовательно, следуя прогрвмме, записанная ввиде кодов. При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. В самом общем виде 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 (англ. «Assembler») — это низкоуровневый язык программирования, который представляет собой промежуточное звено между машинным кодом и высокоуровневыми языками программирования. Он используется для написания программ, которые управляют компьютером или другими устройствами на более низком уровне, непосредственно взаимодействуя с аппаратным обеспечением. Код, написанный на этом языке, обычно сохраняется с помощью расширения ASM.

Программы на ассемблере пишутся в виде набора мнемонических инструкций, каждая из которых соответствует определенной команде процессора. Эти инструкции затем транслируются (ассемблируются) в машинный код — набор двоичных чисел, которые понимает центральный процессор и выполняет соответствующие операции. Для каждой архитектуры существует свой ассемблер и свой язык ассемблера.

Ниже приведена РИС.1.1 с примерами кода, соответствующими инструкциями на ассемблере и их описание.

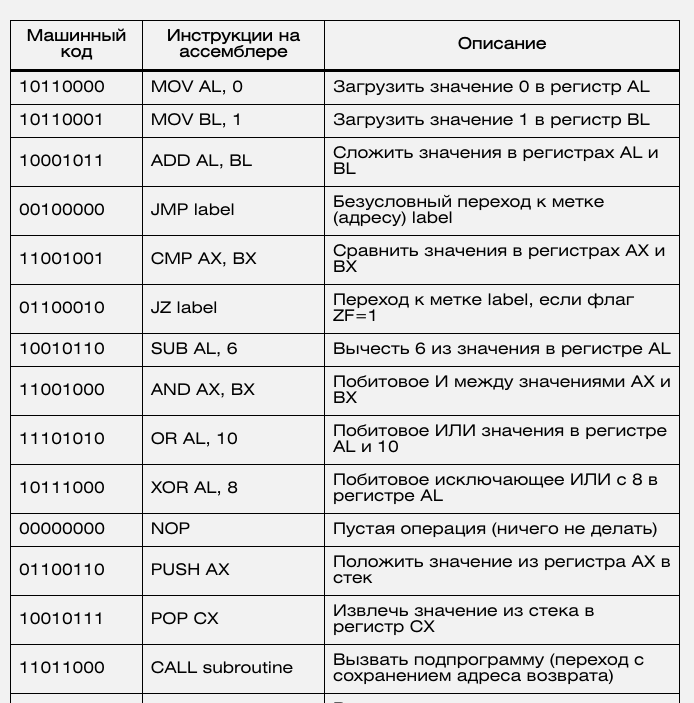


Figure 1: РИС.1.1 Пример кода языка ассемблера

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.

Типичный формат записи команд NASM имеет вид: [метка: мнемокод [операнд {, операнд}] [; комментарий]

В процессе создания ассемблерной программы можно выделить четыре шага:

* набор текста в соответствующий файл;
* трансляция текста а=прграммы с помощью транслятора,наприме nasm;
* компоновка или линковка;
* запуск программы.

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Для начала создадим каталог для работы с программами на языке ассемблера NASM, затем перейдем в этот каталог.(РИС.2.1)

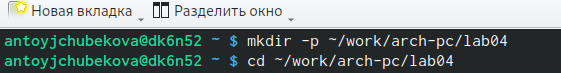


Figure 2: РИС.2.1 Создание каталога

Создадим текстовый файл c именем hello.asm c помощью touch, затем откроем этот файл с помощью редактора report и напишем в него программу для вывода на экран Hello world!(РИС2.2\_1) И (РИС.2.2\_2)



Figure 3: РИС.2.2\_1 Создание файла hello.asm

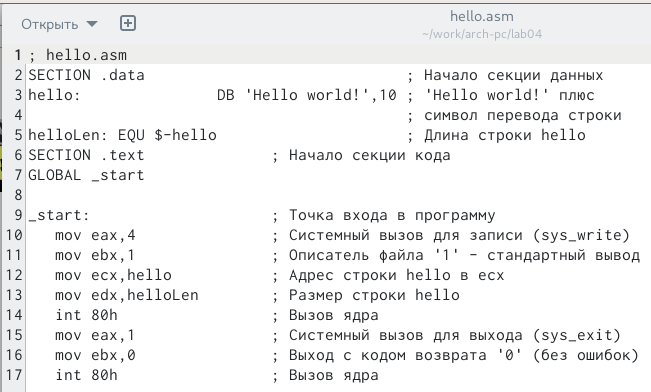


Figure 4: РИС.2.2\_2 Написание программы в файле

## 4.2 Транслятор NASM

Дальше с помощью команды nasm -f elf hello.asm компилируем текст программы для вывода Hello world! в объективный код в новом файле hello.o, в формате ELF. Используя команду ls мы вмдим, что файл был создан.(РИС.3.1)

Figure 5: РИС.3.1 Компиляция текста программы

Figure 5: РИС.3.1 Компиляция текста программы

## 4.3 Расширенный синтаксис командной строки NASM

Введем команду для компиляции файла hello.asm в файл obj.o, при этом формат выходного файла будет elf и в него будут включены символы для откладки (опции-g), также будет создан файл листинга list.lst(опции-l). С помощью команды ls мы удостоверились, что все файлы были правильно созданы.(РИС.4.1)

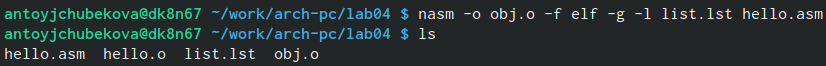


Figure 6: РИС.4.1 Компиляция текста программы

## 4.4 Компоновщик LD

Дальше передадим объективный файл hello.o на обработку комповщику LD используя команду “d -m elf\_i386 hello.o -o hello”. Дальше проверяем корректность выполнения команды,введя команду ls.(РИС.5.1)

Figure 7: РИС.5.1 Передача объективного файла на обработку комповщику ld

Figure 7: РИС.5.1 Передача объективного файла на обработку комповщику ld

Дальше введем команду “d -m elf\_i386 obj.o -o main”. Исполняемый файл будет иметь имя main, так как после ключа -о было задано значение main. Проверим наличие файлов командой ls.Объективный файл, из которого собран этот испоняемый файл, имеет имя obj.o. (РИС.5.2)

Figure 8: РИС.5.2 Передача объективного файла на обработку комповщику ld с заданным именем

Figure 8: РИС.5.2 Передача объективного файла на обработку комповщику ld с заданным именем

## 4.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл hello и види, что она успешно запустиласью(РИС.6.1)

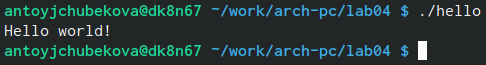


Figure 9: РИС.6.1 Запуск исполняемого файла

## 4.6 Задание для самостоятельной работы

Для начала в каталоге ~/work/arch-pc/lab04 с помощью команды cp создадим копию файла hello.asm с именем lab4.asm. С помщью ls проверим их наличие(РИС.7.1)

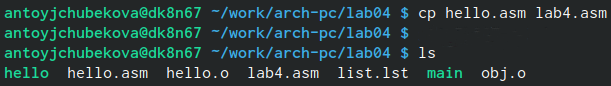


Figure 10: РИС.7.1 Создание копии файла с другим именем

С помощью текстового редактора gedit вношу изменения в текст программы в файле lab4.asm так, чтобы на экран выводилась мое фамилие и имя.(РИС.7.2)

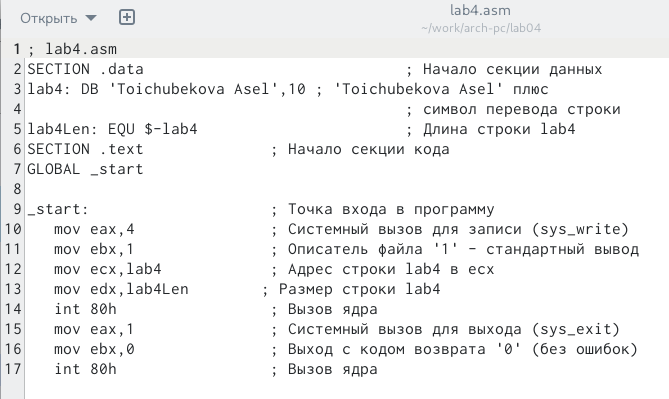


Figure 11: РИС.7.2 Изменение программы

Оттранслируем полученный текст программы lab4.asm в объектный файл. С помощью команды ls мы видим, что файл lab4.o был создан.(РИС.7.3)

Figure 12: РИС.7.3 Компиляция программы в объективный файл

Figure 12: РИС.7.3 Компиляция программы в объективный файл

Далее выполним компоновку объектного файла, с компоновщиком ld. Введя команду ls можно заметить, что исполняемый файл lab4 существует.(РИС.7.4)

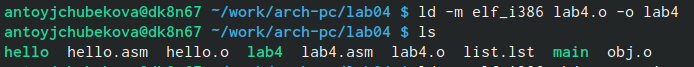


Figure 13: РИС.7.4 Компоновка объективного файла

Запустим получившийся исполняемый файл.(РИС.7.5)

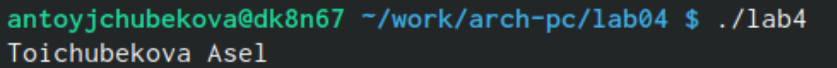


Figure 14: РИС.7.5 Запуск исполняемого файла

Скопируем файлы hello.asm и lab4.asm в наш локальный репозиторий в каталог- ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Перейдя в наш репозиторий и с командой ls видим, что все скопировалось.(РИС.7.6)

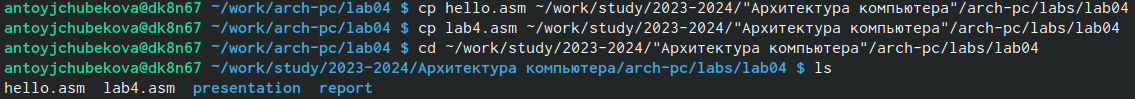


Figure 15: РИС.7.6 Копирование файлов

Загрузим файлы на Github и проверим их наличие.(РИС.7.7) И (РИС.7.8).

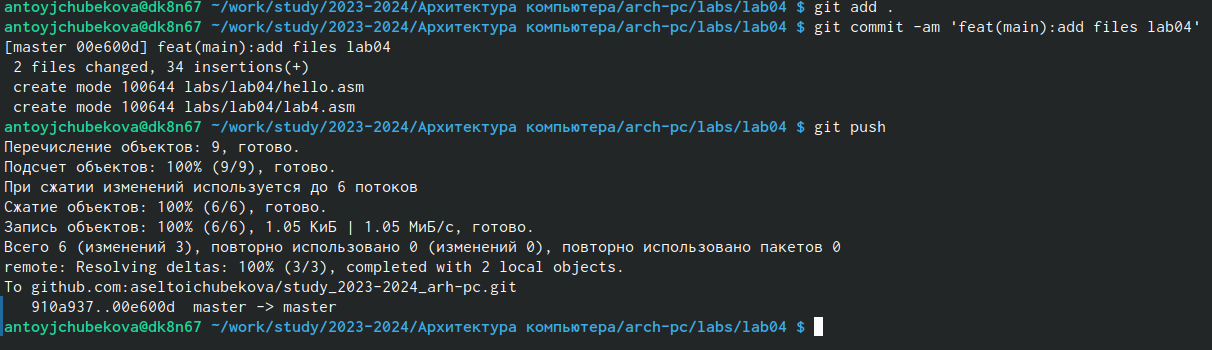


Figure 16: РИС.7.7 Загрузка файлов в github

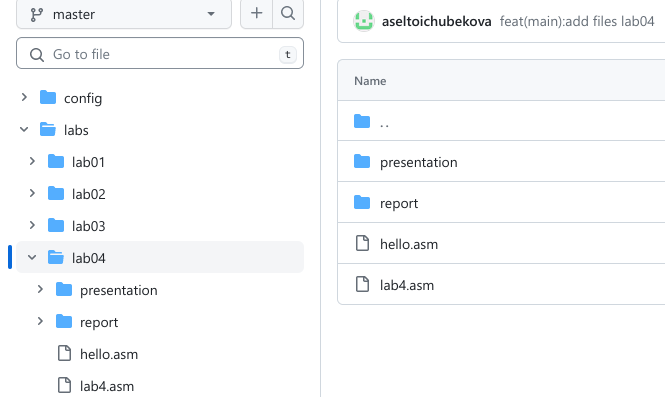


Figure 17: РИС.7.8 Проверка загрузки файлов в github

# 5 Выводы

В ходе выполнения лабораторной работы №4 я поняла основные принципы компьютера, познакомилась с понятием ассемблер и с языком ассемблера NASM, освоила процедуры комиляции и сборки программ, написанных на ассемблере NASM. Пользуясь полученными знаниями написала программу на языке ассемблера NASM, выводящяя на экран фразу “Hello world!” и мое имя и фамилие.

# Список литературы

-https://esystem.rudn.ru/course/view.php?id=4975  
-https://skillbox.ru/media/code/chto-takoe-assembler/  
-https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source