Отчёт по лабораторной работе №4

Дистиплина: архитектура компьютера

Худдыева Дженнет

Содержание

# 1 Цель работы

Цель данной лабораторной работы-освоить процедуры компиляции и сборки программ,написанных на ассембдера 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 компьютеров.В состав центрального процессора входят следующие устройства: - арифметико-логическое устройства (АЛУ) - выполняет логические и арифметические действия,необходимые для обработки информации, хранящейся в памяти. - устройства 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 ассемблера,необходимо знать,какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблера используют регистры в качестве операндов.Практически все команды представляют с собой преобразование данных хранящихся в регистрах процессоров,это например пересылка данных между регистрами или памятью,преобразование (арифметические или логические операции) данных хранящихся в регистрах.Доступ к регистрам осущечтвляется не по адресам,как к основной памяти,а по именам.каждый регистор процессора архитектуры х86 имеет свое название,состоящее из 3 или 3 букв латинского алфавита.В качестве примера приведём названия основных регистров общего назначения (именноэти регистры чаще всего используются при написании программ): - RAX,RCX,RDX,RBX,RSI,RDI-64 bit - EAX,ECX,EDX,EBX,ESI,EDI-32 bit - AX,CX,DX,BX,SI,DI-16 bit - AH,AL,CH,DH,DL,BH,BL-8 bit

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ).ОЗУ-это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процесоора,предназначенное для хранения программ и данных,с которыми процес- сор непосредственно работает в текущий момент,ОЗУ состоит из одинаковых ячеек памяти.Номер ячейки памяти - это адрес хранящихся в ней данных.Периферийные утсройства в составе ЭВМ: - устройства внешней памяти,которые предназначеныдля долговременного хранения больших объёмов данных. - устройства ввода-вывода,которые обеспечивают взаимодействие ЦП с внешной средой.

В основе вычислительного процесса ЭВМ лежит принцип программного управления.Это означает,что компьютер решает поставленную задачу как последовательность действий,записанных в виде программы.

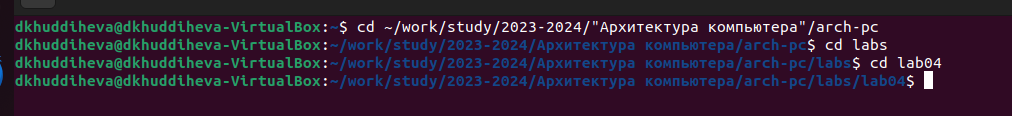
Коды команды представляют собой многозарядные двоичные комбинации из 0 и 1.В коде машинной команды можно выделить две части: операционную и адресную.В операционной части хранится код команды,которую необходимо выполнить.В адресной части хранятся данные или данных,которые участвуют в выполнении данной 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.

# 4 Выполнение лабораторной работы

## 4.1 Создание программы Hello world!

С помощью утилиты cd перемещаюсь в каталог,в котором буду работать (рис. [??]).



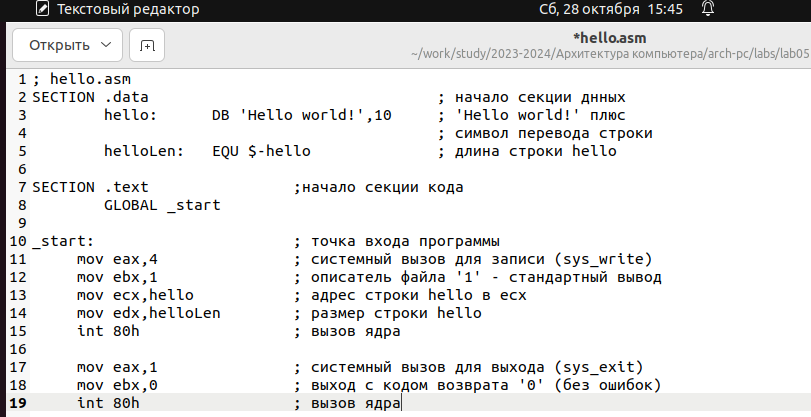
Перемещение между директориями

Создаю в текущем каталоге пустой текстовый файл hello.asm с помощью утилиты touch (рис. [??]).

Создание пустого файла

Создание пустого файла

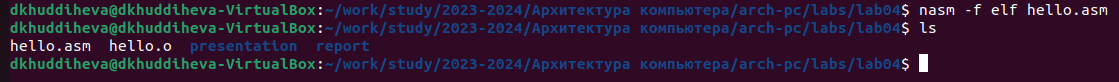
Открываю созданный файл в текстовом редакторе gedit.Заполняю файл,вставляя в него программу для вывода “Hello world!” (рис. [??]).



Заполнение файла

## 4.2 Работа с транслятором NASM

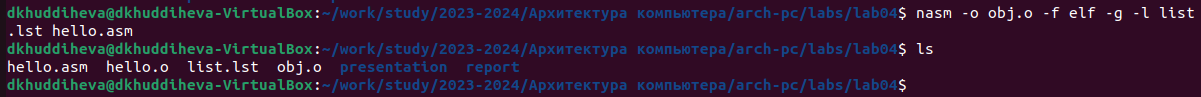
Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду -f elf hello.asm ключ -f указывает транслятору nasm,что требуется создать бинарный файл в формате ELF.Далее проверяю выполнение команды с помощью ls (рис. [??]).



Компиляция текста программы

## 4.3 Работа с расширенным синтаксисом командной строки NASM

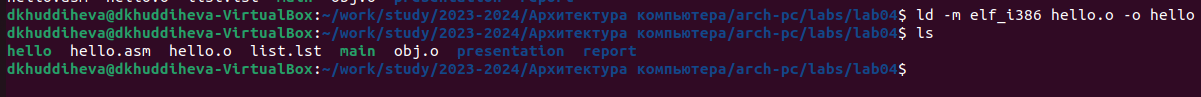
Ввожу команду,которая скомпилирует файл hello.asm в файл obj.o,при этом в файл будут включены символы для отладки (ключ-g), также с помощью ключа -l будет создан файл листинга list.lst (рис. [??]).



Компиляция текста программы

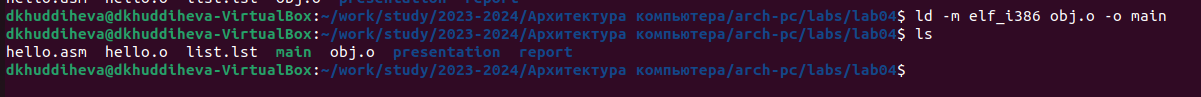
## 4.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD,чтобы получить исполняемый файл hello (рис. [??]).



Передача объектного файла на обработку компоновщику

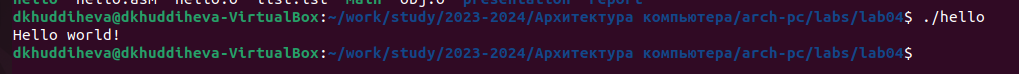
Выполняю следующую команду (рис. [??]). Исполняемый файл будет иметь имя main,после ключа -o было задано значение main. Объектный файл,из которого собран этот исполняемый файл имеет имя obj.o



Передача объектного файла на обработку компоновщику

## 4.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. [??]).



Запуск исполняемого файла

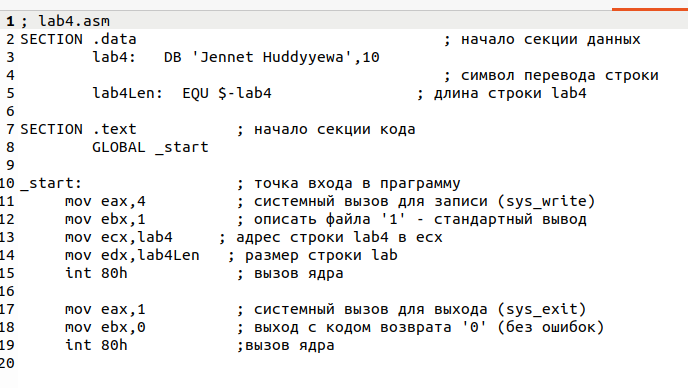
## 4.6 Выполнение заданий для самостоятельной работы

С помощью утилиты cp создаю в текущем каталоге копию файла hello.asm с именем lab4.asm (рис. [??]).

Cоздание копии файла

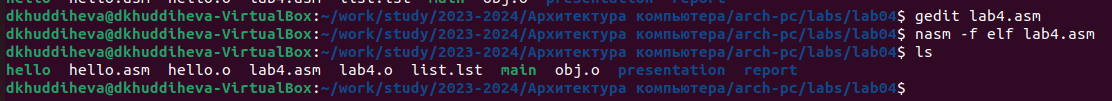
Cоздание копии файла

С помощью текстового редактора gedit открываю файл lab4.asm и вношу изменения в программу так,чтобы она выводила моё имя и фамилию (рис. [??]).



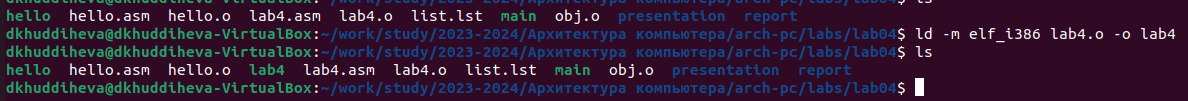
Изменение программы

Компилирую текст программы в объектный файл (рис. [??]).Проверяю с помощью утилиты ls?что файл lab5.asm создан.



Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD,чтобы получить исполняемый файл lab4 (рис. [??]).



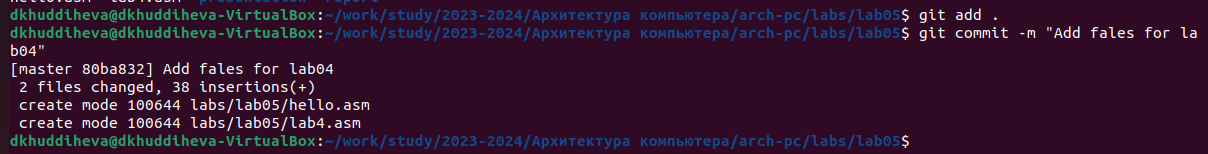
Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4,на экран действительно выводятся моё имя и фамилия (рис. [??]).

Запуск исполняемого файла

Запуск исполняемого файла

С помощью команд git add .и git commit добавляю файлы на GitHub (рис. [??]).

.Отправляю файлы на сервер с помощью команды git push.

# 5 Выводы

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