Создание и процесс обработки программ на языке ассемблера NASM

Лабораторная работа №4

Губайдуллина Софья Романовна

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задание

Порядок выполнения лабораторной работы:

1. Программа Hello world!
2. Расширенный синтаксис командной строки NASM
3. Компоновщик LD
4. Запуск исполняемого файла
5. Задание для самостоятельной работы

# 3 Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать.

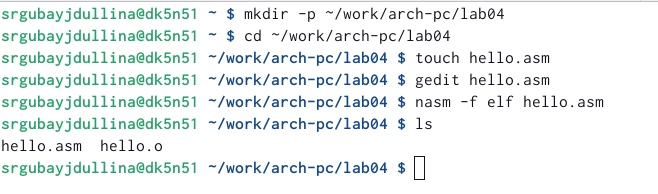
В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы. Программа состоит из машинных команд, которые указыва- ют, какие операции и над какими данными (или операндами), в какой последовательности необходимо выполнить. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер.

Наиболее распространёнными ассемблерами для архитектуры x86 являются: • для DOS/Windows: Borland Turbo Assembler (TASM), Microsoft Macro Assembler (MASM) и Watcom assembler (WASM); • для GNU/Linux: gas (GNU Assembler), использующий AT&T-синтаксис, в отличие от большинства других популярных ассемблеров, которые используют Intel-синтаксис.

В процессе создания ассемблерной программы можно выделить четыре шага: набор текста, трансляция, компановка (линковка) и запуск программы. Для создания программ на языке ассемблера обычно пользуются утилитами командной строки.

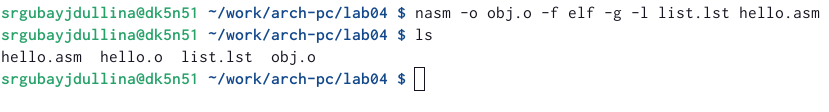
# 4 Выполнение лабораторной работы

1. Для того, чтобы проделать лабораторную работу, мне необходимо создать каталог для работы с программами на языке ассемблера NASM. Перехожу в созданный каталог. При помощи утилиты touch создаю в каталоге текстовый файл с именем hello.asm и открываю при помощи gedit. Ввожу нужный текст.



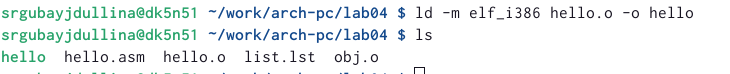
Создание каталога. Создание тестового файла в каталоге и его компляция

1. Зная, что NASM превращает текст программы в объектный код, ввожу команду nasm, которая скомпилирует исходный файл hello.asm в obj.o. Тут же проверяю правильность введенной операции при помощи ls (рис.2)



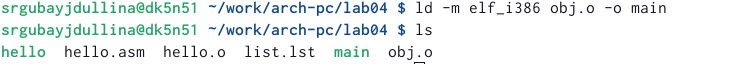
Компляция файла hello.asm

1. Для того, чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику ld. Сразу проверяю правильность выполненной операции (рис.3).



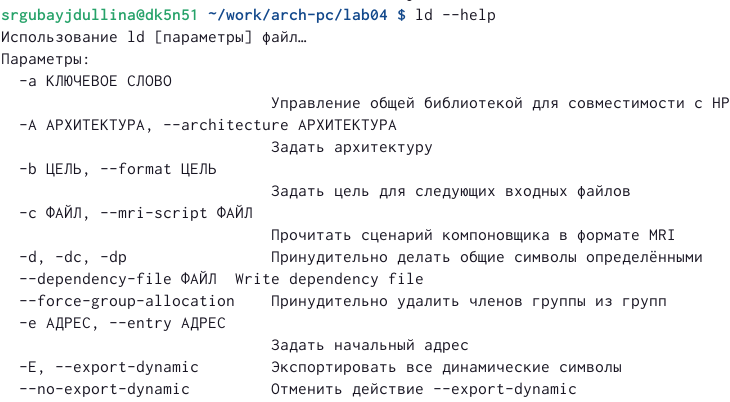
Компановка файла hello.o

Выполняю следующую команду для компановки файла obj.o (рис.4)



Компановка файла obj.o

Также при помощи ld –help я изучила формат командной строки LD (рис.5)



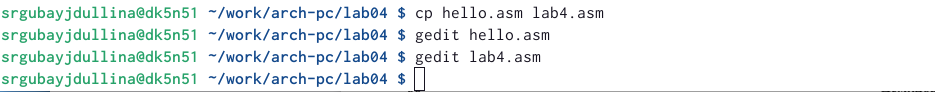
Формат командной строки ld

1. Набрав в командной строке ./hello я запускаю на выполнение созданный исполняемый файл, находящийся в текущем каталоге (рис.6)

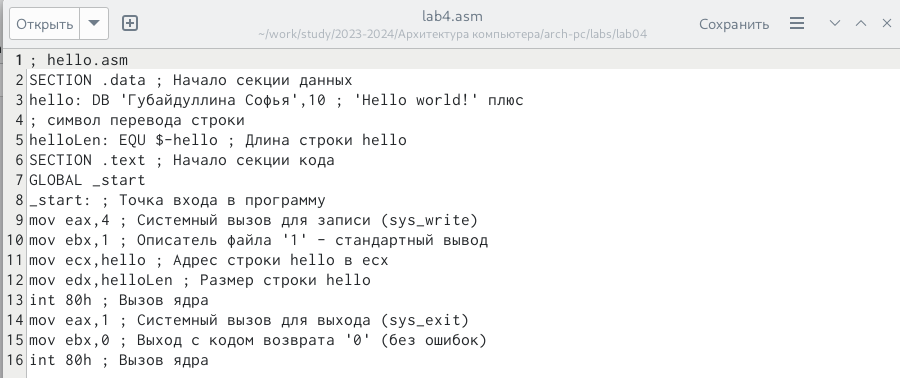
Hello World!

Hello World!

1. Далее приступаю к выполнению самостоятельной работы. Для этого в новом каталоге ~/work/arch-pc/lab04 с помощью команды cp создаю копию файла hello.asm с именем lab4.asm (рис.7). При помощи gedit по заданию я меняю текст программы в lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем. (рис.8)



Копирование файла hello.asm с именем lab4.asm

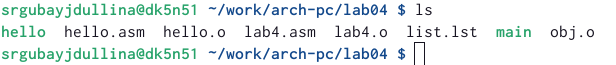


Редактирование в gedit

После мне необходимо транслировать полученный текст программы lab4.asm в объектный файл (рис.9) (рис.99).

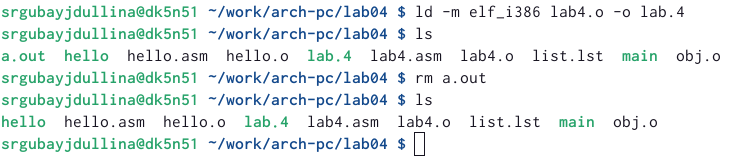
Трансляция

Трансляция

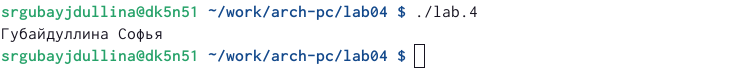


Проверка выполненной трансляции

Компаную его и запускаю получившийся исполняемый файл при помощи изученных в ходе лабораторной работы операций (рис.10) (рис.11)

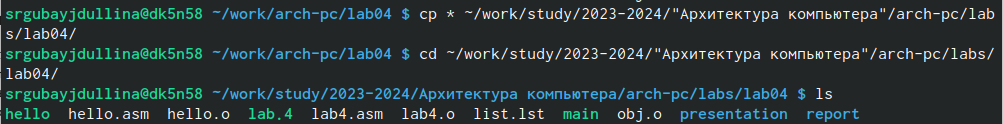


Компановка

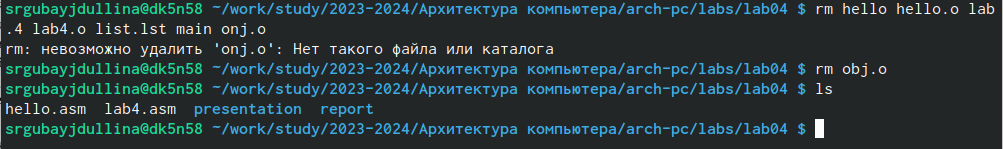


Запуск исполняемого файла

В завершении копирую файлы hello.asm и lab4.asm в свой локальный репозиторий GitHub в привычный каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ (рис.12), удаляя при этом остальные файлы, которые больше мне не пригодятся при помощи утилиты rm (рис.13)

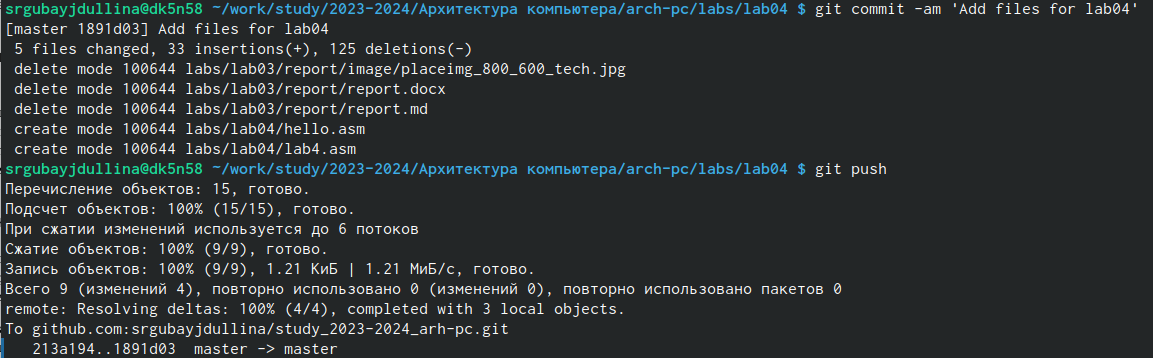


Копирование файлов в нужный каталог



Удаление ненужных скопированных файлов

Полученные файлы загружаю в свой репозиторий GitHub (рис.14)



Загрузка файлов в GitHub

# 5 Выводы

В ходе выполнения лабораторной работы я успешно освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.