Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Будник Александра Олеговна

Содержание

# 1 Цель работы

Освоить процедуры компиляции и сборки программ, которые написаны на ассемблере NASM.

# 2 Теоретическое введение

Основные функциональные элементы любой ЭВМ - ценральный процессор (ЦП), память, периферийные устройства. В состав ЦП входят арифметико-логическое устройство (АЛУ),устройство управления (УУ) и регистры. Периферийные устройства можно разделить на устройства внешней памяти и устройства ввода-вывода.

Язык ассемблера - это машинно-ориентированный язык низкого уровня. Считается, что он наиболее приближен к архитектуре ЭВМ.Ассемблер - это специальная программа транслятор.

Процесс создания и обработки программы на языке ассемблера: 1. Текст программы набирается в текстовом редакторе (файл с расширением .asm). 2. При помощи NASM (транслятор) текст превращается в объектный код. 3. При помощи компоновщика ld код обрабатывается, создается исполняемый файл. 4. Производится запуск программы

# 3 Выполнение лабораторной работы

При помощи команды mkdir созаю каталог для работы с программами на языке ассемблера NASM. Перехожу в него через команду cd (рис. ??).

Создание каталога

Создание каталога

Создаю файл hello.asm командой touch и открываю его в текстовом редакторе (рис. ??).

Создание файла

Создание файла

В текстовом редакторе ввожу будущий код программы, соблюдая синтаксис языка (рис. ??).



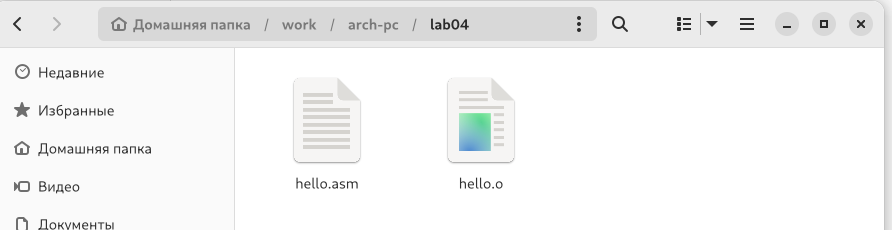
Код программы

При помощи транслятора NASM превращаю текст в объектный код, вводя следующую команду (рис. ??).

Трансляция

Трансляция

Проверяю, что файл расширения .o с кодом создан (рис. ??).



Проверка

Выполняю следующую последовательность команд, чтобы узнать расширенный синтаксис командной строки. Она компилирует файл hell.asm в obj.o, так чтобы формат выходного файла будет elf, в него будут включены символы для отладки и создан файл листинга. При помощи команды ls проверяю, все ли файлы были созданы (рис. ??).

Расширенный синтаксис

Расширенный синтаксис

С помощью компоновщика LD обрабатываю объектный файл и получаю исполняемый файл (рис. ??).

Линковка

Линковка

Ввожу аналогичную последовательность команд с другими исходными данными (рис. ??). При этом исполняемый файл будет называться main. Объектный файл, из которого он собран, называется obj.o.

Линковка другого файла

Линковка другого файла

Запускаю получившуюся программу. Она выводит текст “Hello World!” (рис. ??).

Запуск программы

Запуск программы

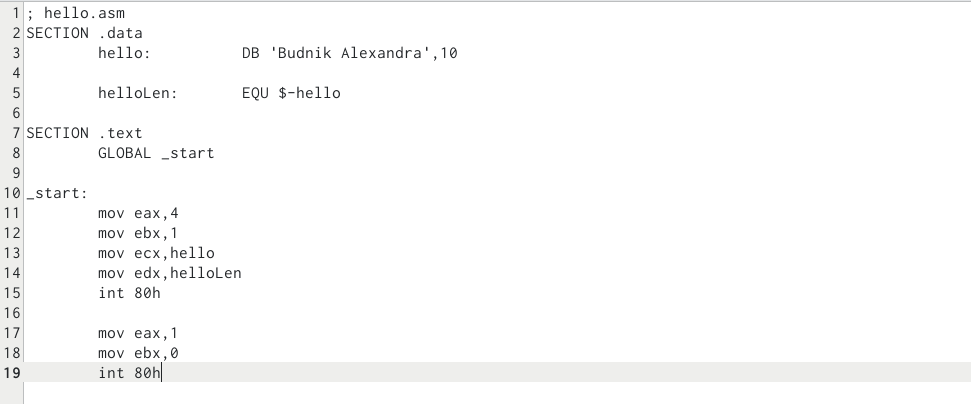
# 4 Выполнение заданий для самостоятельной работы

В каталоге создаю копию файла hello.asm с названием lab4.asm при помощи команды cp (рис. ??).

Копирование

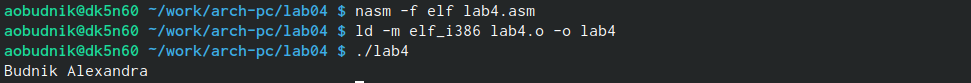
Копирование

Открываю получившийся файл в текстовом редакторе и изменяю код так, чтобы при запуске программа выводила моё имя и фамилию (рис. ??).



Изменение кода

Произвожу трансляцию текста в объектный файл, затем его линковку (компоновку). Получаю исполняемый файл. Запускаю программу (рис. ??).



Трансляция, линковка, запуск

Все получившиеся файлы копирую в локальный репозиторий и загружаю на GitHub.

# 5 Выводы

По итогам выполнения лабораторной работы №4 я научилась работать с машинно-ориентированным языком низкого уровня ассемблера NASM и создала первые две прораммы.