Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютеров

Филипьева Ксения Дмитриевна

Содержание

# 1 Цель работы

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

# 2 Задание

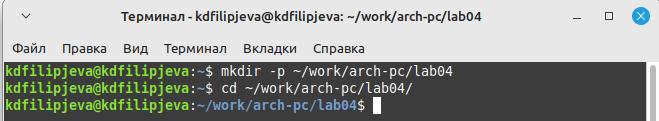
1. Выполнение лабораторной работы №4
2. Заполнение отчета по выполнению лабораторной работы №4 с помощью языка разметки Markdown
3. Выполнение заданий для самостоятельной работы

# 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, таких как C/C++, Perl, Python и пр.

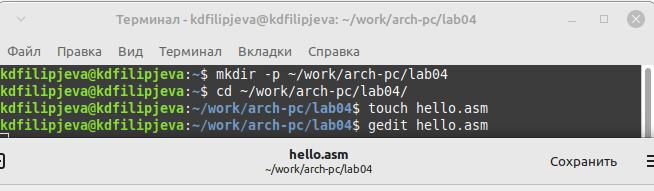
# 4 Выполнение лабораторной работы

Открываем терминал, создаем папку “lab04” в папке “work” для работы с программами на языке ассемблера NASM и переходим в нее. (рис. ??)



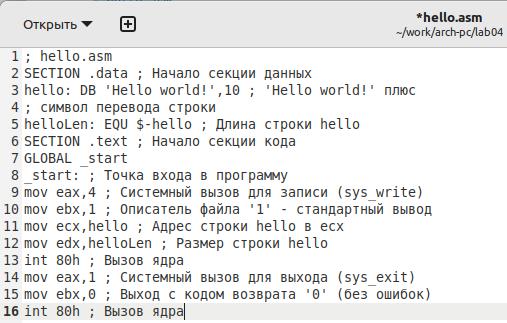
Создание каталога для работы с ассемблером

После того, как перешли в созданный нами каталог, создадим файл “hello.asm” и откроем его с помощью текстового редактора “gedit” (рис. ??)



Создание файла “hello.asm”

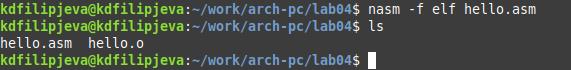
После того, как файл был открыт с помощью текстового редактора, вставим в него заранее заготовленный текст на языке ассемблера. (рис. ??)



Вставленный текст в редакторе

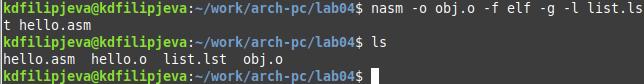
В отличие от многих современных высокоуровневых языков программирования, в ассемблерной программе каждая команда распологается на отдельной строке. Так же синтаксис чувствителем к регистру.

С помощью команды “nasm” скомпилируем текст программы “Hello world”. Текст был вставлен без ошибок, поэтому после, при проверке через “ls”, у нас появился необходимый файл - “hello.o” (рис. ??)



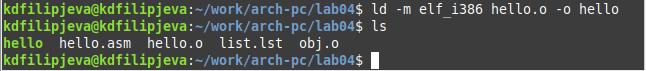
Компиляция программы

После введем полный вариант команды “nasm”, который скомпилирует нашу программу в файл “obj.o”. После выполнения проверим правильность выполнения с помощью команды “ls”. (рис. ??)



Компиляция с помощью полной команды

Теперь передаем исполняемую программу на обработку компановщику, используя команду “ld”. Проверив выполнение с помощью команды “ls” необходимый нам исполняемый файл “hello”. (рис. ??)



Компановка программы

Опять используем компановщика, чтобы задать имя создаваемого исполняемого файла. Проверяем правильность выполнения. (рис. ??)



Задаем имя исполоняемого файла

Выполним созданный нами исполняемый файл с помощью команды “./hello” (рис. ??)

Выполнение исполняемого файла

Выполнение исполняемого файла

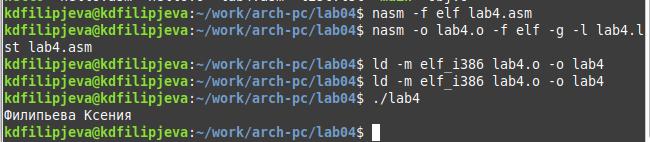
# 5 Выполнение самостоятельной работы

1. Переходим в каталог ~/work/arch-pc/lab04 с помощью “cd”, после с помощью команды “cp” создаем копию файла “hello.asm” с именем “lab4.asm” (рис. ??)



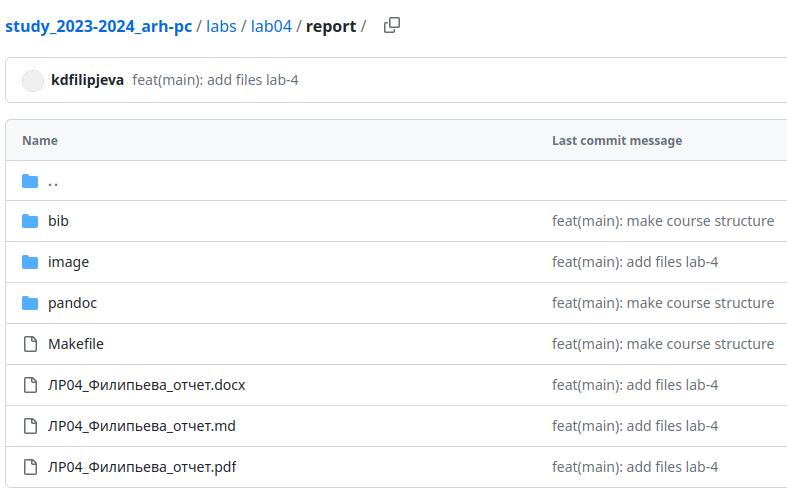
Создание копии файла

1. С помощью текстового редактора gedit внесем изменения в текст программы так, чтобы вместо фразы “hello world” выводились мои имя и фамилия. После конвертируем полученный текст в файле “lab4.asm” в объектный файл, выполним компановку и запустим получившийся исполняемый файл (рис. ??)



Вывод измененного файла

1. Скопируем все файлы в локальный репозиторий и выгрузим на Github (рис. ??)



Файлы на Github

# 6 Выводы

В ходе лабораторной работы мы освоили процедуры компиляции и сборки программ на языке ассемблера.