Отчёт по лабораторной работе №5

Создание и процесс обработки программ на языке ассемблера NASM

Горяйнова Алёна Андреевна

Содержание

# 1 Цель работы

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

# 2 Задание

Рассмотреть пример простой программы на языке ассемблера NASM.

# 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. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора.

Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. До появления языков ассемблера программистам приходилось писать программы, используя только лишь машинные коды, которые были крайне сложны для запоминания, так как представляли собой числа, записанные в двоичной или шестнадцатеричной системе счисления. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер.

Программы, написанные на языке ассемблера, не уступают в качестве и скоро- сти программам, написанным на машинном языке, так как транслятор просто переводит мнемонические обозначения команд в последовательности бит (нулей и единиц).

Используемые мнемоники обычно одинаковы для всех процессоров одной архитектуры или семейства архитектур (среди широко известных — мнемоники процессоров и контроллеров x86, ARM, SPARC, PowerPC,M68k). Таким образом для каждой архитектуры существует свой ассемблер и, соответственно, свой язык ассемблера.

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.

# 4 Выполнение лабораторной работы

Создала каталог для работы с программами на языке ассемблера NASM, перешла в него, создала и открыла текстовый файл hello.asm. Ввела текст из лабораторной работы. Дальше я скомпилировала исходный файл hello.asm в obj.o, передала на обработку компоновщику и проверила всё это с помощью команды ls. И наконец запустила на выполнение созданный файл. (рис. 1)

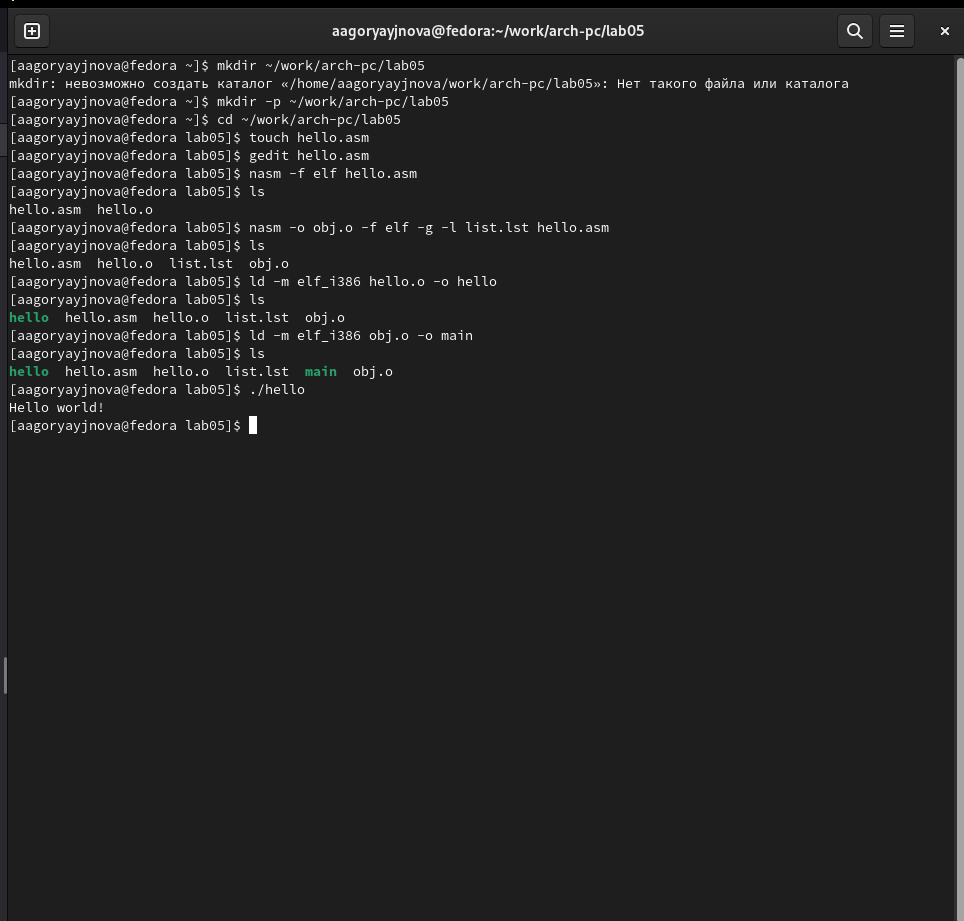


Рис. 1: Создание простой программы, выыводящей Hello world!

Приступила к выполнению заданий ля самостоятельной работы, создала копию файла hello.asm с именем lab5.asm, поменяла текст программы. Оттранслировалв полученный текст программы lab5.asm в объектный файл, выполнила компоновку объектного файла и запустила получившийся исполняемый файл. (рис. 2, 3)



Рис. 2: Текст измененной программы

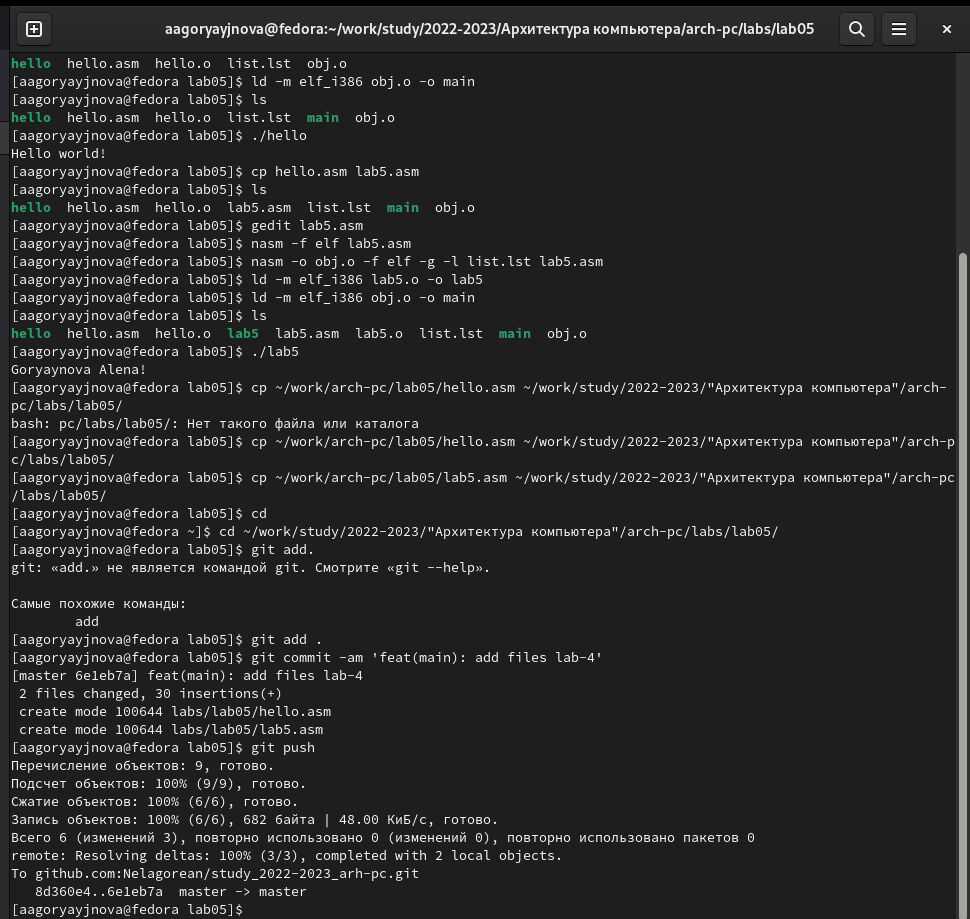


Рис. 3: Запуск программы

# 5 Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы