Отчет по выполнению лабораторной работы №4

Дисциплина: архитектура компьютеров

Намруев Максим Саналович

Содержание

# 1 Цель работы

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

# 2 Задание

1. Программа Hello World!
2. Транслятор NASM
3. Расширенный синтаксис командной строки NASM
4. Компоновщик LD
5. Запуск исполняемого файла

# 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 сязыка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер. Программы, написанные на языке ассемблера, не уступают в качестве и скорости програм- мам, 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 контрол- леров x86, ARM, SPARC, PowerPC,M68k). Таким образом для каждой архитектуры существует свой ассемблер и, соответственно, свой язык ассемблера

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Открываю терманал.Создаю каталог для работы с программами на языке ассемблера NASM. (рис. [??]).

Создание каталога

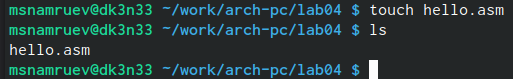
Создание каталога

Перехожу в созданный каталог. (рис. [??])

Переход в каталог

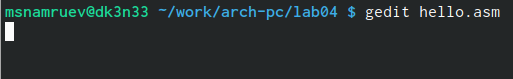
Переход в каталог

Создаю текстовый файл с именем hello.asm. (рис. [??])



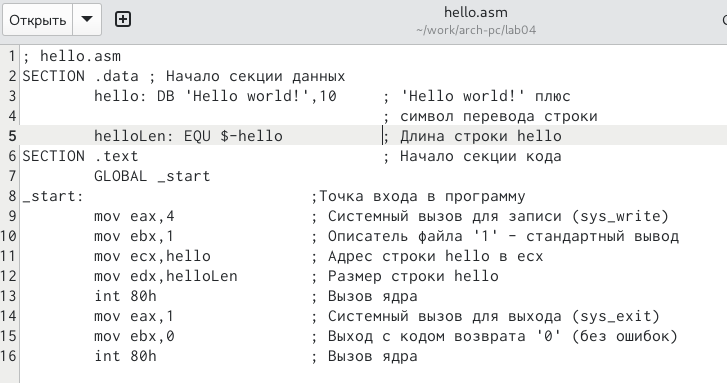
Создание файла

Открываю этот файл с помощью текстового редактора gedit.(рис. [??])



Открываение файла

Ввожу в файл следующий текст.(рис. [??])

{#fig:fig5 width=70%

## 4.2 Транслятор NASM

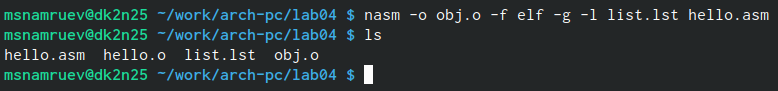
Для выполнения компиляции текста программы “Hello world!” пишу: (рис. [??])

Выполнение компиляции

Выполнение компиляции

## 4.3 Расширенный синтаксис командной строки NASM

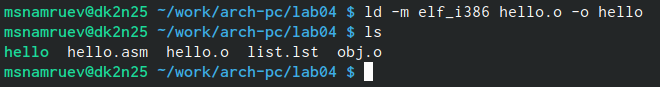
Ввожу команду, которая компилирует исходный файл hello.asm в obj.o при этом формат выходного файла будет elf, и в него будут включены символы для отладки(опция -g), крме того будет создан файл листинга list.lst(опция -l). С помощью команды ls проверяю, что файлы были созданы.(рис. [??])



Компиляция текста

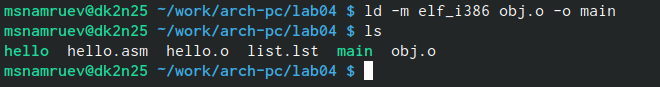
## 4.4 Коспоновщик LD

Передаю объектный файл на обработку компановщику и проверяю, что исполняемый фйл hello был создан. (рис. [??])



Передача объектного файла компановщику

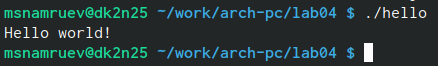
Выполняю следующую команду (рис. [??]) ,после которой файл будет иметь имя main. объектный файл из которого создан исполняемый файл, имеет имя obj.o.



Передача объектного файла компановщику

## 4.5 Запуск исполняемого файла

Запускою созданный исполняемый файл (рис. [??])

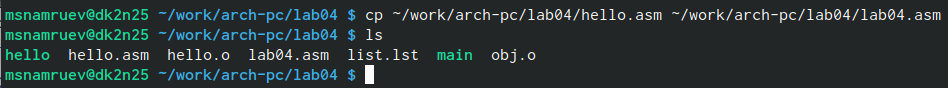


Запуск исполняемого файла

# 5 Задание для самостотельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла hello.asm с именем lab4.asm

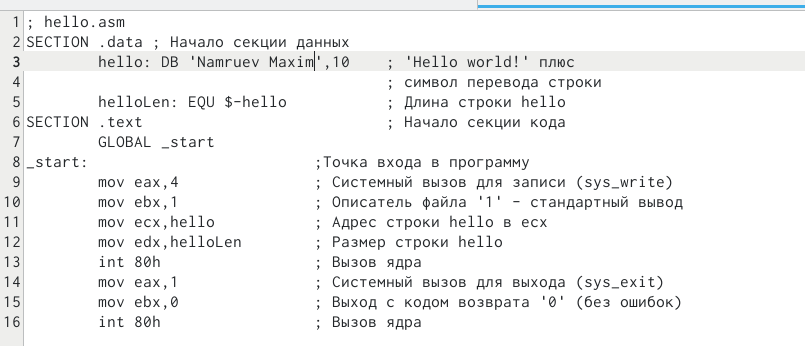
Для этого использую команду cp, указывая сначала тот файл, который хочу скопировать, а затем копию файла с новым названием.Также не забываю проверить правильность выполненных дейтствий с помощью команды ls. (рис. [??])



Копирование файла

1. С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.

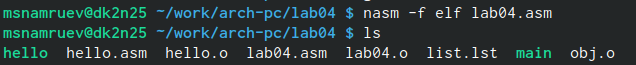
Для этого использую текстовый редактор gedit.Далее вместо “Hello World!” пишу свое имя и фамилию.(рис. [??])



Редактирование файла

1. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл

Далее компилирую текст программы в объектный файл и проверяю выполнились ли действия.(рис. [??])



Компиляция файла

После этого передаю файл на обработку компоновщику ld.(рис. [??])

Отправка файла на компоновку

Отправка файла на компоновку

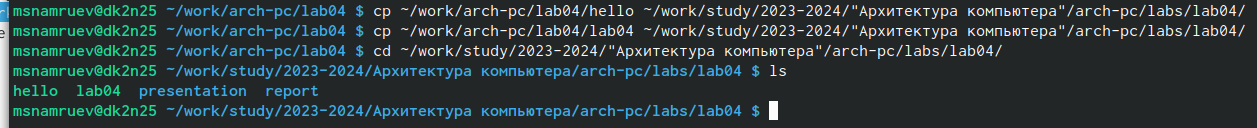
Запускаю файл,чтобы убедиться, что всё выполнено правильно.(рис. [??])

Запуск файла

Запуск файла

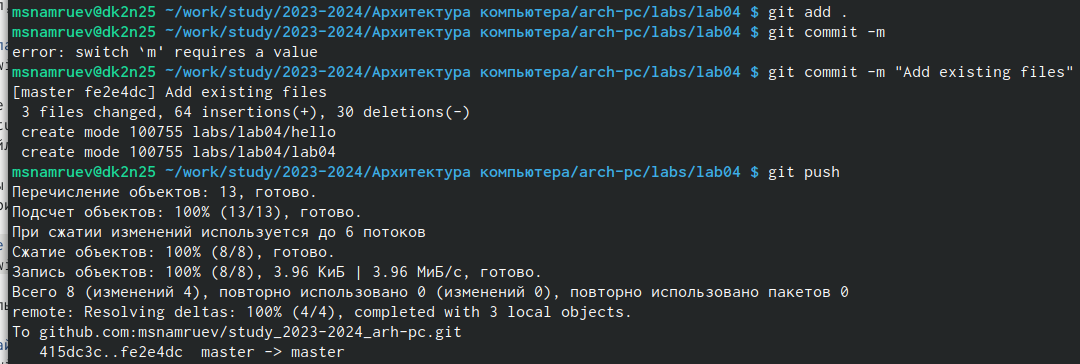
1. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в ката- лог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузите файлы на Github.

Копирую файлы в свой лакольный репозиторий с помощью команды cp, не забывая проверить это с помощью команды ls.(рис. [??])



Копирование файлов

Добавляю файлы на GitHub, используя команды git add., git commit -m и git push.(рис. [??])



Отправка файлов

# 6 Выводы

После выполнения данной работы я освоил написание, сборку и компиляцию программ, написанных на ассемблере NASM.