Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Мурзаев Замир Зейнадинович НБИбд-02-22

Содержание

# 1 Цель работы

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

# 2 Задание

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создайте копию файла hello.asm с именем lab5.asm
2. С помощью любого текстового редактора внесите изменения в текст про- граммы в файле lab5.asm так, чтобы вместо Hello world! на экран выво- дилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы lab5.asm в объектный файл. Выполните компоновку объектного файла и запустите получивший- ся исполняемый файл.
4. Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch- pc/labs/lab05/. Загрузите файлы на Github.

# 3 Теоретическое введение

Процесс создания ассемблерной программы можно изобразить в виде следующей схемы **¿fig:020?**



В процессе создания ассемблерной программы можно выделить четыре шага:

• Набор текста программы в текстовом редакторе и сохранение её в от- дельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип asm.

• Трансляция — преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst.

• Компоновка или линковка — этап обработки объектного кода компонов- щиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расшире- ния. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map.

• Запуск программы. Конечной целью является работоспособный испол- няемый файл. Ошибки на предыдущих этапах могут привести к некор- ректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При на- хождении ошибки необходимо провести коррекцию программы, начиная с первого шага.

# 4 Выполнение лабораторной работы

Создаем новый каталог, как на рисунке 1, но в нашем случае каталог уже существует, поэтому терминал и выдаёт ошибку.

Рис. 1: Команда mkdir

Рис. 1: Команда mkdir

Переходим в данный каталог, создаём в нём текстовый файл и открываем его через текстовый редактор gedit, как на рис 2.

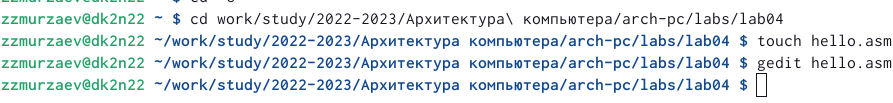


Рис. 2: создание текстового файла

Вводим в файл следующий текст: 3



Рис. 3: написание текстовой программы на языке NASM

Скомпилируем текст написанной программы с помощью команды как на рис 4

Рис. 4: компилияция текста и создание объектного файла

Рис. 4: компилияция текста и создание объектного файла

Проверка успешности совершения команды на рис 5

Рис. 5: команда ls

Рис. 5: команда ls

Скомпилируем тот же самый файл, только теперь зададим сами имя для объектного файла murzaev.o, как на рис 6

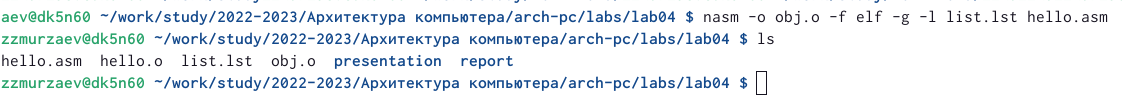


Рис. 6: опция -g - включение симолов для откладки, а опция -l - создание файла листинга

Компановка программы на рис 7 и последующая порверка с помощью ls

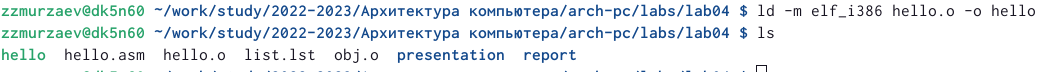


Рис. 7: команда ld

Сделаем то же самое с другими файлами как на рис 8. Исполняемый файл будет иметь имя main, а объектный - obj.o

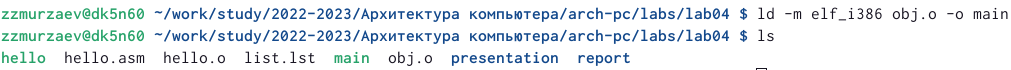


Рис. 8: команда ld

Формат командно строки LD можно увидеть, если ввести ld –help, как на рисунке 9

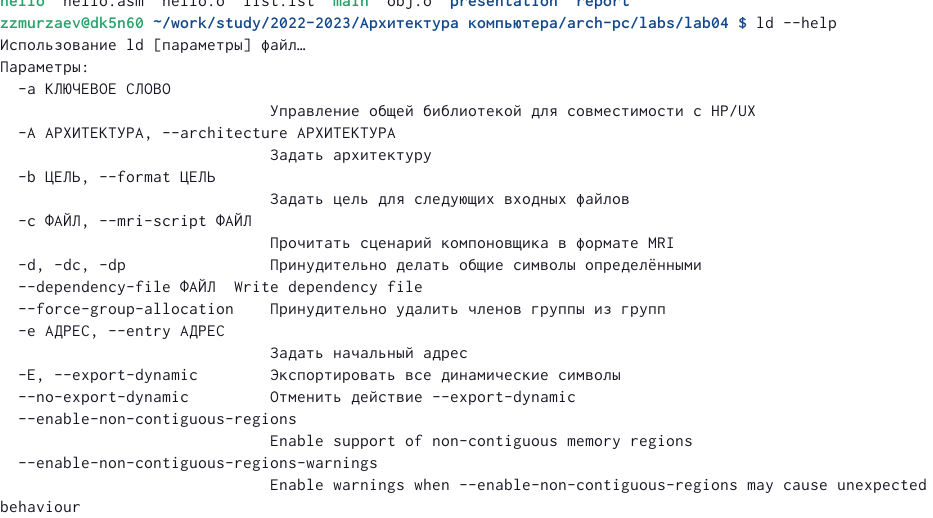


Рис. 9: команда ld –help

Чтобы запустить исполняемый файл, нужно набрать в командной строке ./и имя файла, смотреть на рис 10

Рис. 10: команда запуска файла

Рис. 10: команда запуска файла

# 5 Задания самостоятельной работы

1)C помощью команды cp создайте копию файла hello.asm с именем lab4.asm, смотреть на рис 11

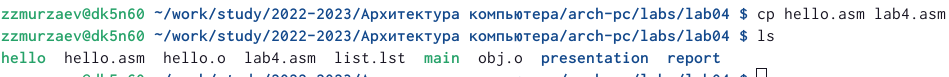


Рис. 11: копирование файла

2)С помощью текстового редактора gedit 12 изменяю текст в файле так, чтобы вместо Hello world! выводились мои имя и фамилия 13



Рис. 12: вызов текстового редактора

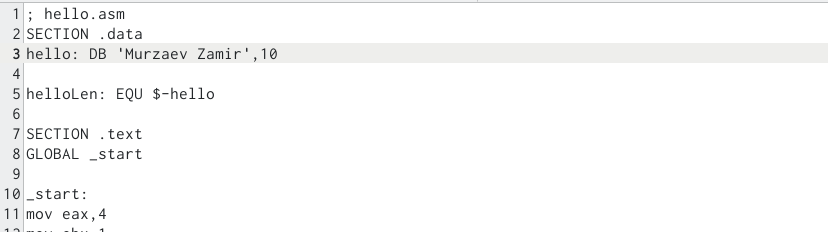


Рис. 13: замена текста в файле

3)Компилируем объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл, 14

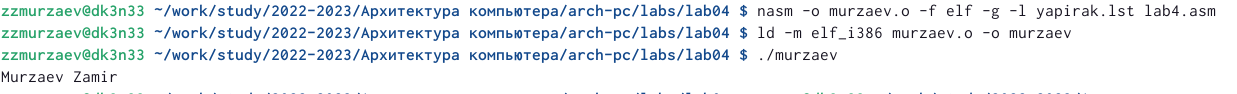


Рис. 14: Процесс преобразования текстового файла в файл исполняемый и его запуск

Нам осталось только скопировать получившиеся файлы в локальный репозиторий и запустить на github. Весь процесс отображен на рис 15

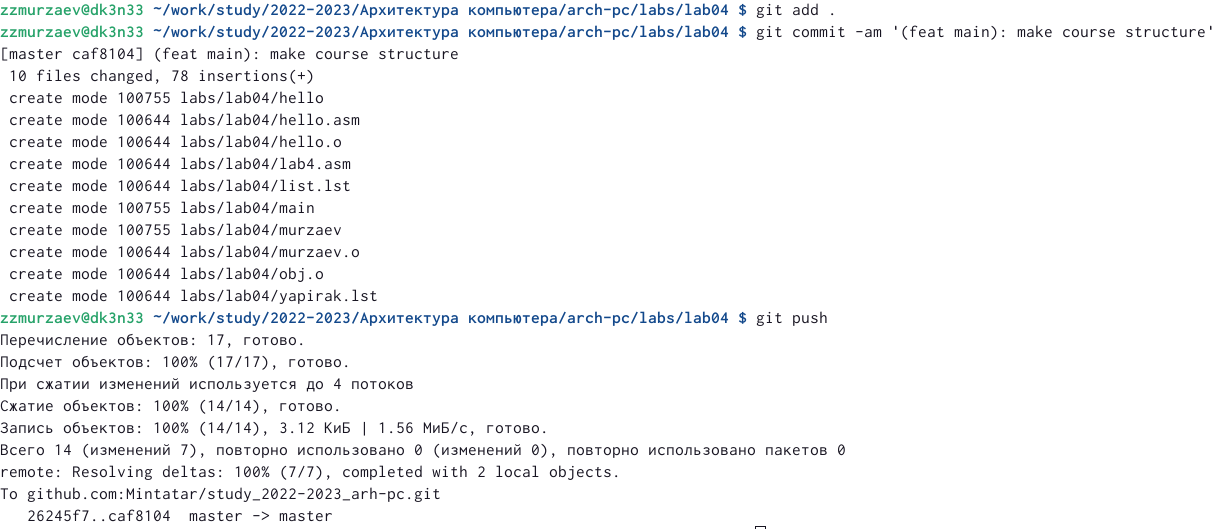


Рис. 15: загрузка всех изменений на github

# 6 Выводы

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

# Список литературы