**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 4**

*дисциплина: Архитектура компьютера*

Студент: Сырцева Анастасия Романовна

Группа: НММбд-03-24

**МОСКВА**

2024 г.

Содержание

[1 Цель работы 2](#__RefHeading___Toc188_3911220052)

[2 Задание 2](#__RefHeading___Toc190_3911220052)

[3 Теоретическое введение 2](#__RefHeading___Toc192_3911220052)

[4 Выполнение лабораторной работы 4](#__RefHeading___Toc194_3911220052)

[5 Самостоятельная работа 5](#__RefHeading___Toc196_3911220052)

[6 Выводы 6](#__RefHeading___Toc198_3911220052)

[Список литературы 7](#__RefHeading___Toc200_3911220052)

# 1 Цель работы

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

# 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 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. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды.Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер. Процесс создания ассемблерной программы можно изобразить в виде следующей схемы (рис. 1).



Рис. 1: Схема создания ассемблерной программы

В процессе создания ассемблерной программы можно выделить четыре шага: - *Набор текста* программы в текстовом редакторе и сохранение её в отдельном 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 компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map. - *Запуск программы*. Конечной целью является работоспособный исполняемый файл. Ошибки на предыдущих этапах могут привести к некорректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При нахождении ошибки необходимо провести коррекцию программы, начиная с первого шага. Из-за специфики программирования, а также по традиции для создания программ на языке ассемблера обычно пользуются утилитами командной строки (хотя поддержка ассемблера есть в некоторых универсальных интегрированных средах).

# 4 Выполнение лабораторной работы

Создаю рабочий каталог и перехожу в него (рис. 2).

Рис. 2: Рабочий каталог

Рис. 2: Рабочий каталог

Создаю текстовый файл с именем hello.asm и открываю его в текстовом редакторе gedit (рис. 3).

Рис. 3: Файл hello.asm

Рис. 3: Файл hello.asm

Ввожу в файл текст указанный в условии лабораторной работы (рис. 4).

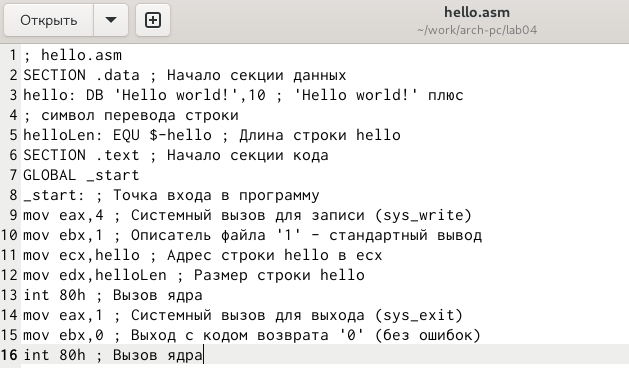


Рис. 4: Редакстирование файла

Компилирую приведённый выше текст и проверяем наличие файла hello.o, в который запишется объектный код (рис. 5).

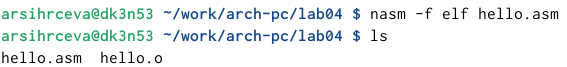


Рис. 5: Преобразование текста программы

Компилирую исходный файл в odj.o и создаю файл листинга (рис. 6).

Рис. 6: Компилирование исходного файла

Рис. 6: Компилирование исходного файла

Проверяю наличие нужных файлов (рис. 7).

Рис. 7: Проверка

Рис. 7: Проверка

Передаю объектный файл на обработку компановщику для получения исполняеймой программы (рис. 8).

Рис. 8: Обработка объектного файла компановщиком

Рис. 8: Обработка объектного файла компановщиком

Выполняю команду, указанную в задании (рис. 9).

Рис. 9: Обработка объектного файла компановщиком

Рис. 9: Обработка объектного файла компановщиком

Исполняемы файл будет называться main, а объектный файл - obj.o Запускаю исполняемый файл hello (рис. 10).

Рис. 10: Запуск программы

Рис. 10: Запуск программы

# 5 Самостоятельная работа

Копирую файл hello.asm в рабочий каталог, меняя название на lab4.asm (рис. 11).

Рис. 11: Копирование файла

Рис. 11: Копирование файла

Открываю текстовый редактор gedit (рис. 12).

Рис. 12: Открытие редактора

Рис. 12: Открытие редактора

В открывшемся редакторе вношу изменения в скопированный файл таким образом, чтобы выводилась строка с моим именем и фамилией (рис. 13).

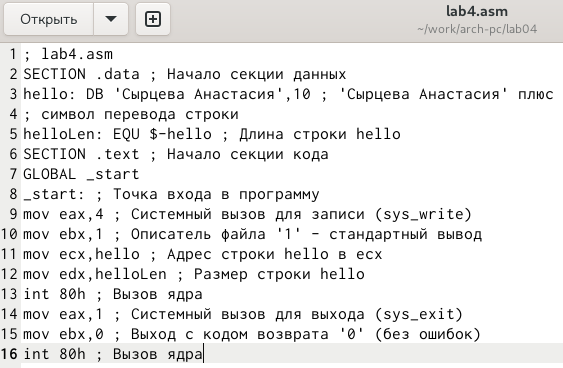


Рис. 13: Изменение файла

Транслирую текст программы в объектный файл и проверяю правильность выполнения (рис. 14).

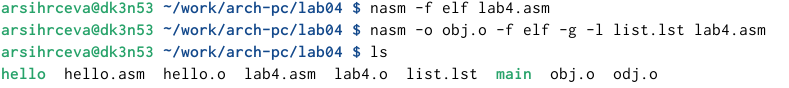


Рис. 14: Компиляция

Компаную объектный файл (рис. 15).

Рис. 15: Компановка

Рис. 15: Компановка

Запускаю исполняемого файла (рис. 16).

Рис. 16: Вывод программы

Рис. 16: Вывод программы

Копирую файлы hello.asm и lab4.asm в свой локальный репозиторий(https://github.com/Anastasia6780/study\_2024-2025\_arh-pc.git) и загружаю написанный отчёт.

# 6 Выводы

Освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM. Написана программа “Hello world!” и программа, выводящая имя и фамилию.