РУДН. Архитектура компьютеров

Отчёт по лабораторной работе №4

Косинов Никита Андреевич, НПМбв-02-20

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Для взаимодействия компьютера и человека есть множество языков программирования (**группа C, Python, Java, Ruby** и пр.). В основном используются высокоуровневые языки, программы которых выглядят суть предложения на английском. Однако, у них есть общий недостаток: ограничения по доступу к возможностям ЭВМ и время работы из-за обработки зачастую лишних операций.

Ассемблерная программа же работает напрямую с ядром машины и, как следствие, наиболее приближенна к машинному коду, за счёт чего делает ровно то, что от неё попросил программист. Поэтому она быстрее, но в то же время, и намного более громоздка. Но для выполнения простых задач или программирования простейших электронных устройств ассемблер необходим.

Цель данной работы - приобретение теоретических и практических навыков по написанию и дальнейшей компиляции простейшей программы, написанной на ассемблере **NASM**.

2 Ход работы

Лабораторная работа выполнена в терминале **ОС Linux** с использованием ассемблера и транслятора **NASM**. Действия по лабораторной работе представлены в следующем порядке:

1. Написание простейшей программы;
2. Создание исполняемого файла;
3. Самостоятельная работа.

3 Написание простейшей программы

По традиции, первой программой предлагается написать “*Hello world!*”.

1. Переходим в каталог *lab04* нашего локального репозитоиря.

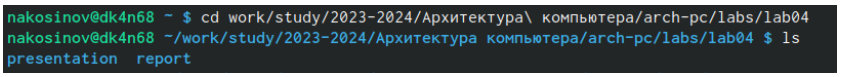

Рис. 3.1: Рабочая папка

2. Создаём файл формата ***.asm** с помощью команды создания файлов **touch**.

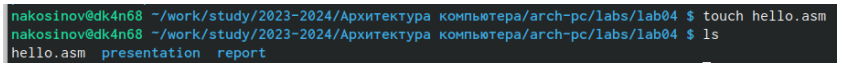

Рис. 3.2: Создание файла программы

3. Открываем файл и записываем туда исполняемый код на языке ассемблер.

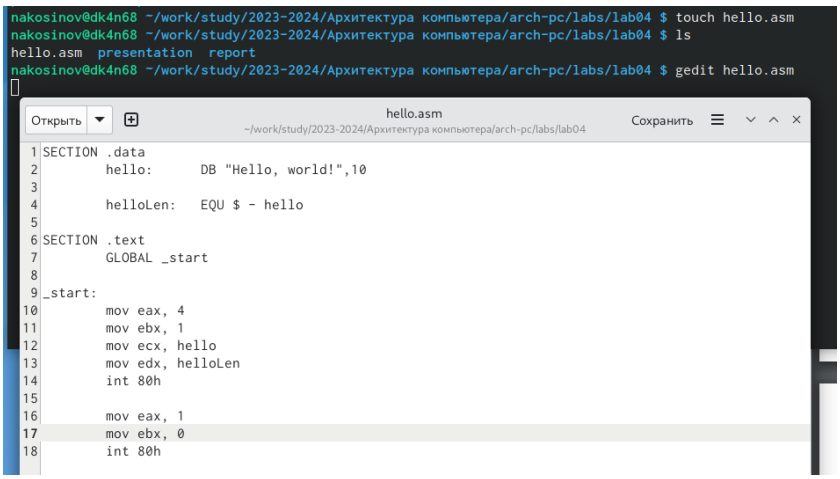

Рис. 3.3: Программа “Hello, world!”

4 Создание исполняемого файла

Чтобы компьютер исполнил нами задуманное, необходимо скомпилировать написанную программу в исполняемый файл. В данном случае будем использовать транслятор **NASM**.

1. Создаём файл формата ***.o** из написанной программы, ипользуя ключ **-f**, чтобы полученный объектный файл был в необходимом формате **elf**.

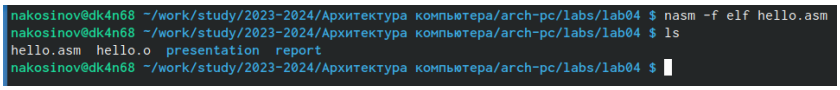

Рис. 4.1: Объектный файл

2. Создаём второй объектный файл более полной командой трансляции, попутно получая файл листинга.

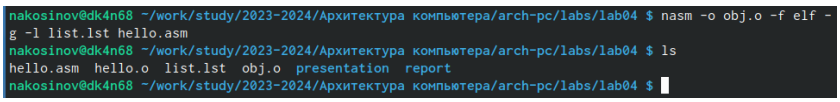

Рис. 4.2: Расширенный синтаксис трансляции

3. Компонуем созданные объектные файлы в исполняемые посредством команды **ld**.

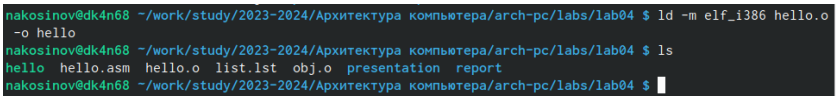

Рис. 4.3: Файл hello создан

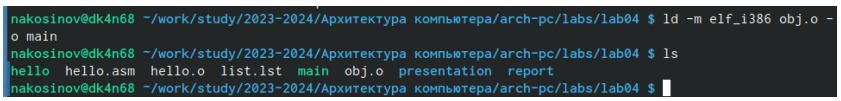

Рис. 4.4: Файл main создан

4. Запускаем оба исполняемых файлов и убеждаемся, что оба работают исправно!

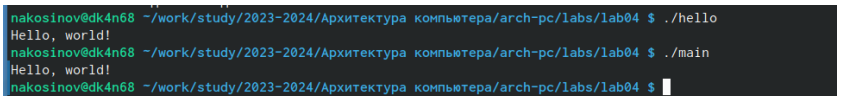

Рис. 4.5: Запуск программ

5 Самостоятельная работа

Для закрепления и большей проработки навыков использования языка ассемблер, напишем программу, выводящую на экран фамилию, имя и номер группы автора.

1. Копируем написанную ранее программу **hello.asm**. Даём копмии новое имя и открываем для редактирования.

Рис. 5.1: Учётная запись с ключом

2. Изменяем код таким образом, чтобы выводились фамилия и имя на первой строке и номер группы на второй.

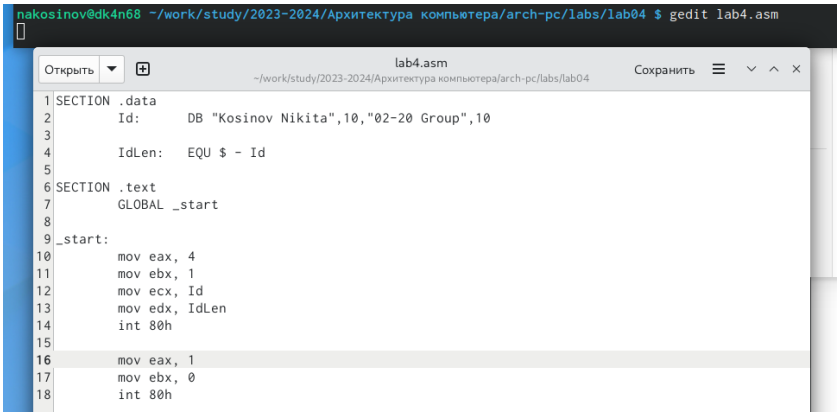

Рис. 5.2: Создание папки

3. Создаём объектный файл **labObj.o** и далее исполняемый **Last_First_Name**. Запускаем последний и видим, что он работает исправно.

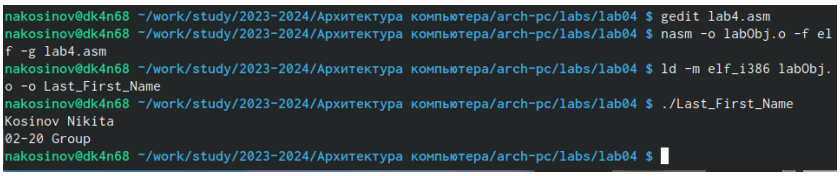

Рис. 5.3: Репозиторий yamadharma

4. Заливаем все сделанные в течение лабораторной работы изменения в глобальный репозиторий на **github**.

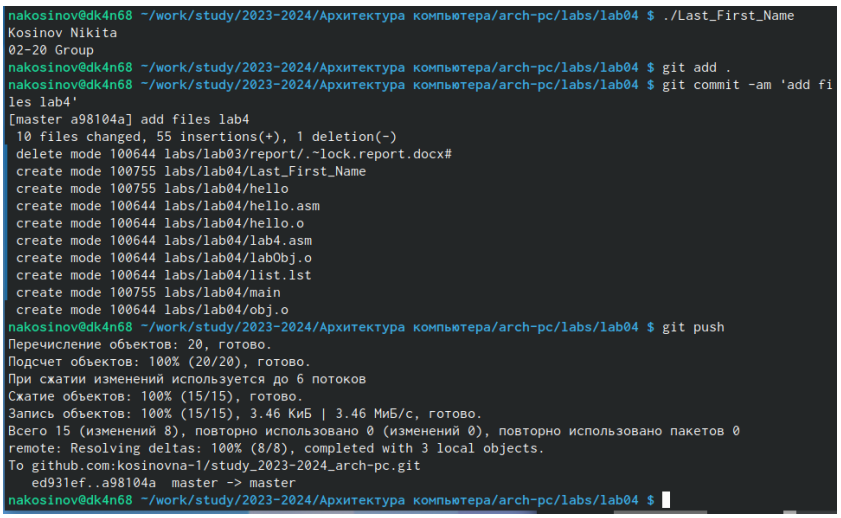

Рис. 5.4: Создание репозитория

6 Выводы

Ассемблерная программа - язык программирования, обращающийся к самому низкому доступному человеку уровню работы ЭВМ. За счёт этого он даёт возможность использовать потенциал машины максимально возможно.

При написании программы на ассемблере важно помнить, что каждая команда дорлжна располагаться на отдельной строке, также важно следить за регистром. Как и любую другую программу, перед запуском её необходимо скомпилировать.