Отчёт по лабораторной работе №4

НКАбд-03-25

Кулаженкова Яна Сергеевна

Содержание

# 1. Цель работы

Освоение процедуры компиляции и сборки программ, написанных на языке ассемблера NASM, включая этапы трансляции, компоновки и запуска исполняемого файла.

# 2. Задание

Лабораторная работа направлена на освоение полного цикла создания программ на языке ассемблера NASM, начиная с изучения теоретических основ архитектуры ЭВМ и системы регистров процессора, практического написания простейшей программы “Hello world!” с использованием системных вызовов Linux, и заканчивая процессами трансляции исходного кода в объектный файл с помощью компилятора NASM, компоновки исполняемого файла линкером LD и его последующего запуска.

# 3. Теоретическое введение

Язык ассемблера — это машинно-ориентированный язык низкого уровня, максимально приближенный к архитектуре ЭВМ и предоставляющий программисту прямой доступ к аппаратным возможностям, таким как регистры процессора. Например, в [табл. **register?**] приведены примеры основных регистров общего назначения. Программы на ассемблере транслируются в машинный код (последовательности нулей и единиц) с помощью специальной программы-транслятора — ассемблера. В рамках данной работы используется ассемблер NASM (Netwide Assembler), который поддерживает Intel-синтаксис и инструкции x86-64. Типичная команда NASM имеет формат: [метка:] мнемокод [операнд] [; комментарий]. Помимо инструкций процессора, программы могут содержать директивы — инструкции для управления работой транслятора, например, для определения данных.

Процесс создания исполняемой программы на ассемблере включает несколько этапов: написание исходного кода и сохранение его в файл с расширением .asm; трансляция исходного текста в объектный код (файл с расширением .o) с помощью ассемблера NASM; компоновка (линковка) объектных файлов в исполняемый файл с помощью компоновщика LD; запуск и, при необходимости, отладка полученной программы.

Примеры регистров. {#register}

| Размер | Регистры |
| --- | --- |
| 64 бита | RAX, RCX, RDX, RBX, RSI, RDI |
| 32 бита | EAX, ECX, EDX, EBX, ESI, EDI |
| 16 бит | AX, CX, DX, BX, SI, DI |
| 8 бит | AH, AL, CH, CL, DH, DL, BH, BL |

# 4. Выполнение лабораторной работы

## 4.1 Программа Hello world!

Создадим директорию для работы с программами на ассемблере и перейдём в неё. Далее создадим файл hello.asm и напишем в нём свою первую программу на ассемблере ([рис. 1](#fig-001)).

|  |
| --- |
| Рисунок 1: Начало работы с файлом hello.asm |

Скомпилируем программу с помощью NASM. Затем создадим исполняемый файл с помощью компоновщика ([рис. 2](#fig-002)).

|  |
| --- |
| Рисунок 2: Подготовка файла hello.asm |

Перед запуском исполняемого файла объектный файл необходимо передать на обработку компоновщику ([рис. 3](#fig-003)).

|  |
| --- |
| Рисунок 3: Передача файла компоновщику |

Запустим свою первую программу на ассемблере ([рис. 4](#fig-004)).

|  |
| --- |
| Рисунок 4: Запуск исполняемого файла |

# 5. Задания для самостоятельной работы

Создадим модифицированную версию файла с кодом. Для этого получим копию файла hello.asm с именем lab4.asm. Затем заменим в файле lab4.asm строку “Hello world!” на свои фамилию и имя ([рис. 5](#fig-005)).

|  |
| --- |
| Рисунок 5: Замена файла hello.asm на файл lab4.asm |

Теперь мы можем скомпилировать и запустить данный файл ([рис. 6](#fig-006)).

|  |
| --- |
| Рисунок 6: Запуск файла lab4 |

Перед завершением работы скопируем все исходные файлы в репозиторий и загрузим на GitHub для сохранения результатов ([рис. 7](#fig-007)).

|  |
| --- |
| Рисунок 7: Загрузка файлов на Github |

# 6. Выводы

Мы освоили работу с компилятором NASM для преобразования исходного кода в объектные файлы и с линкером LD для создания исполняемых программ. На практике мы убедились, как текст программы на ассемблере превращается в работающее приложение. Все этапы - от написания кода до запуска программы - были успешно выполнены. Таким образом, мы достигли цели работы и освоили процесс сборки программ на NASM.