

УДК 004

СОВРЕМЕННОЕ ПРИМЕНЕНИЕ ЯЗЫКА АСSEMBЛЕРА



MODERN APPLICATION OF THE ASSEMBLY LANGUAGE

Цыбенко Богдан Сергеевич

студент группы 21-КБ-ПР1,
Институт компьютерных систем
и информационной безопасности,
Кубанский государственный технологический университет
bgdn3107@gmail.com

Питкин Виктор Александрович

старший преподаватель,
кафедры физического воспитания и спорта,
Кубанский государственный технологический университет
irvik25@mail.ru

Аннотация. В статье приведено краткое описание языка ассемблера, рассматриваются его возможности и современные способы применения.

Ключевые слова: компьютер, технологии, низкоуровневое программирование, язык ассемблера.

Tsybenko Bogdan Sergeevich

Student of group 21-КБ-ПР1,
Institute of Computer Systems
and Information Security,
Kuban State Technological University
bgdn3107@gmail.com

Pitkin Victor Alexandrovich

Senior Lecturer of the Department
of Physical Education and Sports,
Kuban State Technological University
irvik25@mail.ru

Annotation. This article gives a brief description of the Assembler language and discusses its capabilities and current application.

Keywords: computer, technology, low-level programming, assembly language.

Изменение цветов на дисплее монитора, использование мыши, нажатие любой клавиши на клавиатуре – любое взаимодействие с компьютером, будь то ввод или вывод, в итоге сводится к машинному коду, инструкций из нулей и единиц, которые определяют каждое действие в исполнимой программе. Массивность машинного кода и громоздкость труда для его написания, связанная с огромным числом комбинаций нулей и единиц, складывающихся в различные команды, привели к использованию языка ассемблера в начале 50-х годов, который заменял инструкции в опкоде на символьные аналоги [1]. Язык ассемблера называют языком программирования низкого уровня. То есть, он считается языком, написание которого близко к написанию машинного кода, хоть инструкции заменены на сокращения осмысленных слов человеческого языка.

Язык ассемблера по-прежнему имел много недостатков, например, сложность разработки программ с существенным количеством кода. Из-за своей зависимости от машинных инструкций, он также имел особенности написания для каждой конкретной архитектуры компьютера. Для их устранения и для удобства работы были созданы высокоуровневые языки, которые были направлены уже не на машину, а на человека.

Высокоуровневые языки отступили от машинного кода, став посредником между программистом и компьютером. В изучении современных языков программирования редко удаётся встретить упоминания о том, как строятся машинные инструкции, или о предназначении регистров процессора. В основе высокоуровневых языков лежат абстракции и структуры данных – понятия более ясные для человека, чем инструкции для аппаратной части компьютеров. Когда подавляющее количество приложений для конечного пользователя не содержат код на низкоуровневых языках, появляется вопрос о необходимости знания языка ассемблера в принципе.

Для выполнения высокоуровневого кода используются библиотеки среды выполнения, поддерживающие функции, предоставляемые системами программирования, и участвующие в компиляции, то есть генерации машинного кода для выполнения программы. Часть генерируемого в этом случае кода может оказаться лишней, особенно если оптимизация выделяемой памяти и скорость выполнения критически важны. В этом случае язык ассемблера обладает большей свободой, позволяя контролировать структуру кода программы и каждый генерируемый байт. Это связано с близостью языка ассемблера по структуре к машинному коду: трансляция в машинный код не требует таких преобразований, как для компиляции кода высокоуровневого языка.

Из этого следует, что на языке ассемблера может быть написано то, что нуждается в максимальной скорости, как можно более оптимальном использовании памяти

или в прямом доступе к возможностям аппаратных частей компьютера. В первую очередь, это драйвера и программы, взаимодействующие напрямую с видеокартами, звуковыми картами, или портами [2]. Сюда же можно отнести создание прошивок для комплектующих ПК и других устройств, а также программирование микроконтроллеров – микросхем для управления электронными устройствами, у которых память довольно ограничена. По той же причине небольшие программы на телефонах, принтерах или автомобильных системах зажигания могут быть написаны языке ассемблера.

В силу своего свободного доступа к любой области памяти и способности свободно управлять адресацией, ассемблер имеет доступ к механизму прерываний. Резидентные программы, остающиеся в оперативной памяти и перехватывающие прерывания, сегодня не используются, но схожий механизм для «внедрения» в другие процессы применяется при создании некоторых видов вредоносного ПО и, соответственно, антивирусного ПО [3].

Ассемблер может использоваться совместно с высокоуровневыми языками, если те не имеют инструментов для выполнения некоторой задачи или выполняют её не наиболее оптимальным образом. Некоторые математические вычисления, задачи копирования памяти, часть алгоритмов обработки мультимедиа или другие критичные фрагменты программы могут быть написаны на языке ассемблера. Впоследствии, эти из этих фрагментов формируют подпрограммы, которые внедряются в высокоуровневый код [1].

Знание языка ассемблера необходимо при дизассемблировании – трансляции отдельных модулей или скомпилированной программы в язык ассемблера для тщательной отладки вплоть до машинных инструкций или для обратного проектирования – исследования программы (если документация неполная), модификации или копирования, в том числе и нелегального.

Если отойти от прикладного применения, то с 70-х годов существует практика создания «демосцен», как вида компьютерного искусства. Они возникли, как своеобразные подписи в виде генерирующегося видеоряда, запускающегося перед или вместе с другой программой, в качестве указания на авторство. Идея демосцены заключалась в создании наиболее впечатляющего видеоряда, занимающего как можно меньшую память, что практикуется сегодня в виде конкурсов, где размер демосцены – не более 256 килобайт [4].

Язык ассемблера по своему существу тесно связан с компьютером. И когда необходимо заглянуть глубже в его устройство, во внутрь любой программы, за высокоуровневые языки, столкновение с ассемблером неизбежно. Современное программирование по большей части полагается на высокоуровневые языки, отчего низкоуровневые уходят на второй план. Но в некоторых случаях компилятор, насколько бы оптимизированным он ни был, не сможет заменить тех скорости и контроля, которые предлагает ассемблер. В руках опытных программистов и энтузиастов язык ассемблера остаётся незаменимо мощным инструментом.

Литература

1. Юров В.И. Assembler. – СПб. : Питер, 2003. – С. 637.
2. Зубков С.В. Assembler для DOS, Windows и UNIX. – М. : ДМК Пресс ; СПб. : Питер, 2004. – С. 608.
3. Хижняк П.Л. Пишем вирус и... антивирус / Под общей ред. И.М. Овсянниковой. – М. : ИНТО, 1991. – С. 90.
4. Демосцена: Хабр. – URL : <https://habr.com/ru/articles/573926>

References

1. Yurov V.I. Assembler. – SPb. : Peter, 2003. – P. 637.
2. Zubkov S.V. Assembler for DOS, Windows and UNIX. – M. : DMK Press; SPb. : Peter, 2004. – P. 608.
3. Khizhnyak P.L. Writing a virus and ... antivirus / Under general editorship of I.M. Ovsyannikova. – M. : INTO, 1991. – P. 90.
4. Demoscene: Habr. – URL : <https://habr.com/ru/articles/573926>