

*Калиничев Станислав Юрьевич, преподаватель факультета довузовской подготовки и среднего профессионального образования, ФГБОУ ВО «Мордовский государственный университет им. Н.П. Огарёва»  
e-mail: [sonmordor@gmail.com](mailto:sonmordor@gmail.com)*

*Пивкин Алексей Александрович, студент факультета довузовской подготовки и среднего профессионального образования, ФГБОУ ВО «Мордовский государственный университет им. Н.П. Огарёва»*

## **ЕСТЬ ЛИ БУДУЩЕЕ У ЯЗЫКА ПРОГРАММИРОВАНИЯ АССЕМБЛЕР?**

**Аннотация:** в данной статье мы попытаемся разобраться в актуальности изучения низкоуровневого языка программирования с целью последующего его применения на при написании кода. Также рассмотрим, где и в каких областях используется ассемблер.

**Ключевые слова:** ассемблер, низкоуровневый язык программирования.

**Abstract:** in this article, we will try to understand the relevance of studying a low-level programming language for the purpose of its subsequent application when writing code. Also consider where and in what areas the assembler is used.

**Keywords:** assembler, low-level programming language.

В наше время многие программисты считают, что не стоит тратить лишнее время на изучение ассемблера поскольку он считается языком программирования низкого уровня, следовательно большие сложные программы на нем писать крайне проблематично. Но если с этой точки зрения всё довольно-таки понятно, то почему же есть немалая часть программистов, которая не готова похоронить в прошлом этот язык программирования? Давайте разбираться.

Язык ассемблера (от англ. assembly language) – это низкоуровневый язык

программирования, который используется для представления команд центрального процессора. При этом команды процессора остаются понятными программисту. Также под термином «ассемблер» подразумевают ПО, которое преобразует исходный код в машинный. Обратный процесс (конвертация машинного кода в ассемблер-код) выполняется при помощи дизассемблера [2].

Иначе говоря, ассемблер – это язык, который позволяет программисту обращаться напрямую к процессору машины.

Можно отметить, ассемблер является символьным видом записи машинного кода, отсюда следует, что понимание кода для процессора на нем становится в какой-то мере проще. При этом для одного устройства можно разработать несколько языков ассемблера. Работа программиста с высокоуровневым языком скрывает проблемы реализации алгоритмов, в отличие от языка ассемблера, который позволяет открыто взаимодействовать с компьютерной системой команд.

Язык ассемблера дает доступ к регистрам и методам адресации, использует терминологию команд процессора для описания операций. Кроме того, он может содержать инструментарий более высокого уровня, такой как макрокоманды, возможность задавать автоматический выбор команд и инструменты для описания структур данных.

Ассемблер нередко называют языком процессоров, отсюда следует, что пока существуют различные типы процессоров, будет жить и сам ассемблер. Программисты, которые отлично владеют ассемблером, могут не только понять код, записанный на языке более высокого уровня, но и оптимизировать его, так как они полностью понимают процессорную архитектуру используемого компьютера.

Самое главное достоинство ассемблера в том, что он максимально приближен к процессору, но в то же время его основной недостаток заключается в делении типовых операций, что очень тяжело воспринимается для немалого количества людей.

У ассемблера также имеется ряд других недостатков. Самый большой

недостаток заключается в том, что для решения любой простой задачи нужно обладать немалыми знаниями, а это иногда является непреодолимым препятствием для начинающего программиста. Также к недостаткам этого языка программирования можно отнести очень длинные листинги программ, из-за чего прочтение ассемблерных программ становится тяжким занятием. Ещё к одному недостатку можно отнести отсутствие потребности у многих современных устройств в подробном описании операций на примитивном языке, то есть, некоторые действия проще и быстрее будет описать на языке высокого уровня. Что же касается библиотек, то они ограничены, в сравнении с высокоуровневыми языками.

Перенос ассемблерного кода с одного контроллера на другой, является трудоемким процессом. А именно, если при переносе где-нибудь в коде программы появится ошибка, то на отладку программы уйдет не мало времени. Данный язык сильно привязан к конкретному семейству контроллеров.

Как и все другие области, область программирования также переживает различные эволюции и нововведения. Из-за этого язык ассемблера становится менее популярным. Появляются новые языки программирования (Фортран, Лисп, Кобол, Паскаль, Си и другие), которые более практичны и быстры. Несмотря на это, ассемблер всегда будет востребован, так как есть довольно большой список задач, где именно использование данного низкоуровневого языка даст максимальный успех. Ассемблер как нельзя лучше всего подходит для написания таких программ, где важным критерием является:

- Скорость работы, быстродействие программы. Это может быть какая-нибудь игра или драйвер для компьютера.
- Объём, который доступен для использования памяти. К ним можно отнести вирусные и защитные программные обеспечения, также программы для микроконтроллеров и микропроцессоров или загрузочные сектора.

Ниже приведён небольшой список задач, решаемых с использованием ассемблера:

1. Повышение скорости реагирования участков программ, написанных

на Си++, к примеру, или на других более высокоуровневых языках. Программа считается оптимально если она выполняется быстро, правильно, занимает малый объем памяти и имеет легкое зрительное восприятие. Многие программисты нашего времени считают, что использовать ассемблер лучше лишь для оптимизации кода программ, путём создания ассемблерных вставок в листинг, написанный на высоком языке программирования, а в одиночку его применять не стоит.

2. Отчасти это утверждение верно, но не стоит забывать, что на ассемблере написаны даже некоторые операционные системы. Довольно часто для написания операционных систем используют язык Си, но изначально он и разрабатывался для создания одной из первых версий Unix. Ассемблер активно используют при написании аппаратно-зависимых участков кода, например, загрузчик ОС, HAL и ядро. В ядрах популярных операционных систем, таких как Windows и Linux также присутствуют небольшие участки с ассемблерным кодом. Некоторое ОС полностью пишут на этом языке, пример – MenuetOS (любительская операционная система для ПК), самое интересное заключается в том, что она полностью уместается на дискете, а интерфейс представлен пользователю в виде многооконной графики.

3. Написание микроконтроллерных и микропроцессорных программ. В наше время микроконтроллеры (МК) развиваются очень стремительно. Ассемблер очень активно используется для программирования МК, там частой операцией является перемещение отдельных байтов из одних ячеек памяти в другие. Профессоры и ученые подчеркивают важность программирования в данной сфере, подтверждая это тем фактом, что в современных жилых домах в среднем находится около 50 микроконтроллеров.

4. Разработка драйверов. Применяется для написания участков кода, непосредственно связанных с аппаратным обеспечением, а остальное пишут чаще на высокоуровневых языках, тем самым получают максимальную надежность и практичность, что действительно необходимо для драйверов, работающих в режиме ядра, где любая ошибка может уничтожить систему.

5. Написание антивирусов и другого защитного программного обеспечения. С антивирусными программами сложилась аналогичная ситуация, как и с разработкой драйверов, программы имеют высокий уровень ответственности в целостности данных и функциональности ОС.

6. Разработка трансляторов для языков программирования. Здесь ситуация куда более понятнее, ассемблер – язык процессоров, можно сказать является одним из близких языком для них, и не на каком другом языке программирования нельзя так грамотно описать компилятор для процессора.

С самого начала появления процессоров ассемблер стал фундаментальным языком программирования. Как ученые, стремящиеся исследовать структуру вселенной, ищут ее первоначальные и неделимые элементы, так и программисты используют ассемблер, чтобы понять работу машинных команд, необходимых для написания программ. Даже если программист никогда не планирует писать на этом языке программирования, без знания его основных принципов невозможно стать опытным профессионалом. Аналогично тому, как невозможно стать математиком, если не знать азы арифметики. Если программист не понимает, как процессор реагирует на операции, то вероятнее он станет наваливать в одну кучу известные ему операции, не понимая, каким будет конечный результат [3].

Однако это не гарантирует, что изучив ассемблер вы найдете хорошую работу, но и попросту времени вы не потеряете, так как обязательно найдете ему применение при написании собственных программ.

Ассемблер, как и любой другой язык, имеет ряд достоинств и недостатков, но плюсов низкоуровневый язык программирования имеет гораздо больше, чем минусов.

Учитывая вышеупомянутые факты, можно сделать вывод, что у ассемблера определённо есть будущее.

### **Библиографический список:**

- 1 Есть ли будущее у ассемблера? [Электронный ресурс] Студенческий

научный форум — Режим доступа: <https://scienceforum.ru/2019/article/2018010638>.

2 Что такое ассемблер: где используется и актуален ли этот язык программирования в 2023 году [Электронный ресурс] LoftSchool.com — Режим доступа: <https://loftschool.com/blog/posts/assembler-что-это>.

3 Язык ассемблера: в чем суть и нужно ли изучать сегодня [Электронный ресурс] GeekBrain IT-образование — Режим доступа: <https://gb.ru/blog/yazyk-assemblera>.