Графика и ассемблер – актуально ли в 2020 году?

Калистратов Иван Александрович, Абрамова Оксана Фёдоровна

Волжский политехнический институт (филиал) ВолгГТУ

"Волгоградский государственный технический университет"

г. Волжский, Россия.

Аннотация

Статья посвящена исследованию актуальности создания графики посредством использования языка программирования ассемблера и сравнения с другими, более высокоуровневыми языками. В статье рассматриваются взаимосвязь и закономерности развития общества с IT индустрией, а также разбираются преимущества и недостатки низкоуровневых языков на примере ассемблера.

Ключевые слова: ассемблер, графика, графика с использованием машинноориентированных языков.

Graphics and assembler – is it relevant in 2020?

Kalistratov Ivan Aleksandrovich, Abramova Oksana Fedorovna

Volzhsky Polytechnic Institute (branch) Volgograd state technical University

Volzhsky, Russia.

Annotation

The article is devoted to the study of the relevance of creating graphics by using the Assembly programming language and comparing it with other, higher-level languages. The article examines the relationship and patterns of development of society with the IT industry, as well as examines the advantages and disadvantages of low-level languages on the example of Assembly language.

Keywords: assembler, graphics, graphics using machine-oriented languages.

В 2020 году уровень технологий по всему миру продолжает стремительно расти вверх. Степень развития IT-отрасли за последние несколько лет подскочила до небывалых высот. Такой прогресс открыл для общества невиданные ранее технологии, которыми уже пользуются в странах третьего мира — например, сотовая связь. Теперь каждый, кто имеет смартфон с доступом в интернет, одновременно имеет доступ к большому количеству информации и к различным средствам работы с нею.

“К чему это все?” – спросите вы. Сейчас я постараюсь ответить на этот вопрос. Начнём с предисловия.

Ассемблер – низкоуровневый язык, который позволяет контролировать каждый мельчайший процесс, на который порой трудно воздействовать в более высокоуровневых языках, например, в C++.

В недалеком прошлом - в начале появления встраиваемых систем - код писался на ассемблере. Это был единственный вариант, потому что тогда шла борьба за каждый байт памяти. Она была в дефиците, и поэтому требовался тщательный контроль за её использованием, с чем прекрасно справлялся ассемблер. Но была и другая причина его повсеместного использования – не было инструментов для языков более высокого уровня.

С течением времени технологии росли, память дешевела, появлялись все новые и новые инструменты. Тогда и настало время, когда благоприятные условия для появления более серьезного кода были сформированы.

Люди всегда стремились упростить свою жизнь - сфера программирования не стала исключением. Программистам стало некомфортно писать огромные коды для реализации трудных задач, и тогда они стали стремиться к разработке более удобных языков под различные задачи, в том числе и под работу с графикой.

Язык ассемблера – довольно сложный для непосвящённого в программирование пользователя и является сложным для вхождения в сферу программирования. Сейчас же существует огромное множество языков, снижающих порог вхождения в IT сферу. Это одно из преимуществ высокоуровневых языков. Больше людей начинают пробовать себя в этой сфере, что только подталкивает вперед развитие IT индустрии.

Но в чём же проблема с графикой? Зачем пришлось создавать адаптированные под неё среды программирования? Дело в том, что с ростом сложности запросов пользователей росли и технологии. Графические возможности низкоуровневых языков уже не удовлетворяли запросов нового времени. Для создания сложных графических анимаций требовалось много трудозатратной работы. На помощь программистам пришли новые языки,

которые со временем становились только лучше и лучше. Создавать графику сейчас стало намного проще и менее трудозатратно, чем не так давно.

В 2020 уровень развития технологий, а именно графики, достиг небывалых высот, которых еще лет 20 назад и не представляли. Графика теперь повсюду. Любое приложение теперь делается по стандартам качества нового времени. Но ведь кто захочет пользоваться интерфейсом нулевых годов?

“Что же насчет создания графики на ассемблере в 2020?” Так я отвечу. Во время прогресса технологий на ассемблере уже очень сложно, я бы сказал почти невозможно создавать все то, что с легкостью делается на других высокоуровневых языках. Огромная трудозатратная и кропотливая работа с кодом уже не стоит того. Даже на создание легкого графического примитива на ассемблере понадобится больше времени и строк кода, чем на том же C++. Это и есть главное преимущество высокоуровневых языков.

Пример:

```
double a = st->value[--st->Count];
```

C++

```
mov ecx, DWORD PTR _st$[ebp]
mov edx, DWORD PTR [ecx]
sub edx, 1
mov DWORD PTR tv71[ebp], edx
mov eax, DWORD PTR _st$[ebp]
mov ecx, DWORD PTR tv71[ebp]
mov DWORD PTR [eax], ecx
mov edx, DWORD PTR _st$[ebp]
mov eax, DWORD PTR [edx+4]
mov ecx, DWORD PTR tv71[ebp]
movsd   xmm0, QWORD PTR [eax+ecx*8
movsd   QWORD PTR _a$[ebp], xmm0
```

MASM

Примитивное действие, объём которого на C++ занимает всего 1 строку, на ассемблере занимает целых 12 строк. С графикой всё довольно сложнее. Естественно, это неудобно и ни к чему, если с этой операцией легко справляется высокоуровневый язык программирования.

Получается, что графика на ассемблере в 2020 году уже не так актуальна, как ранее. Сам язык уже не соответствует тем стандартам, которые требуются для создания качественных графических изображений. Сейчас его используют разве что для вычислений и обработки данных, в скорости чего он определенно фаворит, но совсем проигрывает нынешним языкам в создании графики. То есть графические изображения гораздо практичнее создавать на

специализированных языках, а оптимизацию и обработку данных поручать ассемблеру. Данная комбинация позволит лучше раскрыть потенциал обоих языков и выдать наиболее оптимизированный и качественный код.

Библиографический список:

1. Создание графических примитивов на ассемблере [Электронный ресурс] // URL: https://karpsy.ru/socialnye-seti/sozdanie-graficheskih-primitivov-na-assemblere-programmirovanie-na.html

2. Графический ассемблер [Электронный ресурс] // URL: http://www.chipinfo.ru/literature/chipnews/200007/48.html

3. Ассемблер против C: Зачем изучать ассемблер? [Электронный ресурс] // URL: https://radioprog.ru/post/712

4. Руссу Ю.Д., Абрамова О.Ф. Графика и ассемблер – насколько это совместимо? [Электронный ресурс] // URL: https://scienceforum.ru/2020/article/2018019144

5. Трифанов А.И., Абрамова О.Ф. РЕАЛИЗАЦИЯ СОБСТВЕННОГО МЕТОДА ВИЗУАЛИЗАЦИИ ВОДНОЙ ПОВЕРХНОСТИ «СКОЛЬЗЯЩАЯ ТЕКСТУРА» // Современные наукоемкие технологии. – 2013. – ? 8-1. – С. 96-97