УДК 004

Работа на Assembler в Microsoft Visual Studio 2015

Волков Виталий Александрович
Мордовский государственный университет им. Н.П.Огарева
студент

Аннотация
В данной статье описывается работа на языке ассемблера в MS Visual Studio 2015. Показан метод настройки проекта для работы с языком ассемблера. Для демонстрации использовались компоненты: Microsoft Visual Studio 2015.
Ключевые слова: Ассемблер; машинный код; дизассемблер; транслятор; Microsoft Visual Studio; MASM.

Work on Assembler in Microsoft Visual Studio of 2015

Volkov Vitaliy Alexandrovich
Ogarev Mordovian State University
student

Abstract
This article describes the work in Assembly language into MS Visual Studio 2015. Shows how to configure a project for assembler language. To demonstrate the components were used: Microsoft Visual Studio 2015.
Key words: Assembler; machine code; disassembler; translator; Microsoft Visual Studio; MASM.

Язык ассемблера (assembly language) - язык программирования низкого уровня, каждая инструкция которого соответствует одной машинной команде процессора вычислительной системы. Таким образом, язык ассемблера является специфичным для каждой конкретной машинной архитектуры, в отличие от большинства языков программирования высокого уровня, которые, в идеале, не зависят от архитектуры целевой машины, и программы на которых можно свободно переносить между различными архитектурами.

Команды языка ассемблера один к одному соответствуют командам процессора. Фактически, они и представляют собой более удобную для человека символьную форму записи команд и их операндов. При этом одной мнемонике языка ассемблера может соответствовать несколько вариантов команд процессора. Кроме того, язык ассемблера позволяет использовать символические метки вместо адресов ячеек памяти, которые при ассемблировании заменяются на вычисляемые ассемблером или компоновщиком абсолютные или относительные адреса.

Процесс трансляции программы с языка ассемблера в исполняемый машинный код называется ассемблированием и выполняется ассемблером - программой-транслятором, которая и дала языку ассемблера его название.

К достоинствам языка ассемблера следует отнести следующие:

1) Язык ассемблера позволяет писать самый быстрый, эффективный и компактный код, возможный для данного процессора.
2) Обеспечение максимального использования специфических возможностей конкретной платформы, что также позволяет создавать более эффективные программы.
3) При программировании на языке ассемблера возможен непосредственный доступ к аппаратуре, и, в частности:
   - портам ввода-вывода,
   - регистрам процессора и др.
4) Язык ассемблера часто применяется для создания драйверов оборудования и Ядра операционной системы.
5) Язык ассемблера используется для создания компонентов BIOS.
6) Дизассемблер - программный инструмент для преобразования машинного кода в текст программы на языке ассемблера.

К недостаткам языка ассемблера можно отнести:

1) Сложную программу написать на языке ассемблера намного сложнее, чем на языке высокого уровня.
2) В силу машинной ориентации языка ассемблера человеку сложнее читать и понимать программу на нем по сравнению с программой на языке высокого уровня. А подлежащий редактированию ассемблерный код сравнительно большой. Соответственно, усложняются программирование и отладка, растет трудоемкость разработки, сравнительно велика вероятность программных ошибок.
3) Требуется высокая квалификация программиста. Код на ассемблере выполняется быстрее, но написанный неопытным программистом, обычно оказывается хуже сгенерированного компилятором с языка высокого уровня.
4) Как правило, меньшее количество доступных библиотек по сравнению с современными индустриальными языками программирования.
5) Программы на языке ассемблера не переносимы на компьютеры с другой архитектурой и системой команд, как на уровне машинных команд. Так и на уровне исходных кодов, что в случае с языками ассемблера практически одно и то же.

На практике язык ассемблера применяют для написания программы, которой критически важны:

- быстродействие;
- объем используемой памяти;
- прямой доступ к ресурсам аппаратуры.

К таким программам относятся драйверы устройств, загрузочные секторы операционных систем, встраиваемое программное обеспечение, программы для микроконтроллеров и процессоров с ограниченными ресурсами, вирусы, программные защиты.

Далее рассмотрим структуру программы на языке ассемблера. Файлы программ на языке ассемблера не должны подчиняться строгой структуре, как, например, программы на некоторых языках высокого уровня. Однако, большинство программ подчиняются общепринятым соглашениям и состоят из следующих пяти секций:

1. Заголовок. Заголовок включает в себя название программы, которое записывается при помощи директивы ассемблера TITLE, и спецификации режимов трансляции: указание целевого процессора, модели памяти.
2. Секция объявлений. В ней записываются объявления импортируемых и экспортируемых функций, структур данных и констант.
3. Секция определения данных. В ней содержатся объявления используемых переменных.
4. Секция кода. Содержит исполняемый код программы, записанный при помощи мнемоник ассемблера, определения процедур.
5. Подвал. Содержит директивы ассемблера, обозначающие окончание файла с программой, описание точки входа в программу.

Простейшая программа на языке ассемблера представлена на рисунке 1. Эта программа не выполняет никаких полезных действий, она просто завершается, передавая управление операционной системе и возвращая успешный код возврата.

main.asm

```
1  TITLE Простейшая программа на языке Ассемблера
2
3    .686
4
5    .MODEL FLAT, STDCALL
6
7    .STACK 4096
8
9  ; Импортируемые функции Win32 API
10
11 ; завершение процесса
12 ExitProcess PROTO dwExitCode: DWORD
13
14   .DATA
15
16   .CODE
17
18 main PROC
19
20   INVOKE ExitProcess, 0
21
22 main ENDP
23
24   END main
25
```

Рисунок 1 – Листинг простейшей программы на языке ассемблера

Теперь используем Visual Studio 2015 для запуска нашей программы. В поставку Microsoft Visual Studio 2015 входит ассемблер Microsoft Macro Assembler (MASM) версии 14. Рассмотрим, каким образом можно создать проект и подключить файл с простейшей программой на языке ассемблера.

Из пункта меню File выберите пункт New, и затем пункт Project. Укажите тип приложения Visual C++ Win32 Console Application.

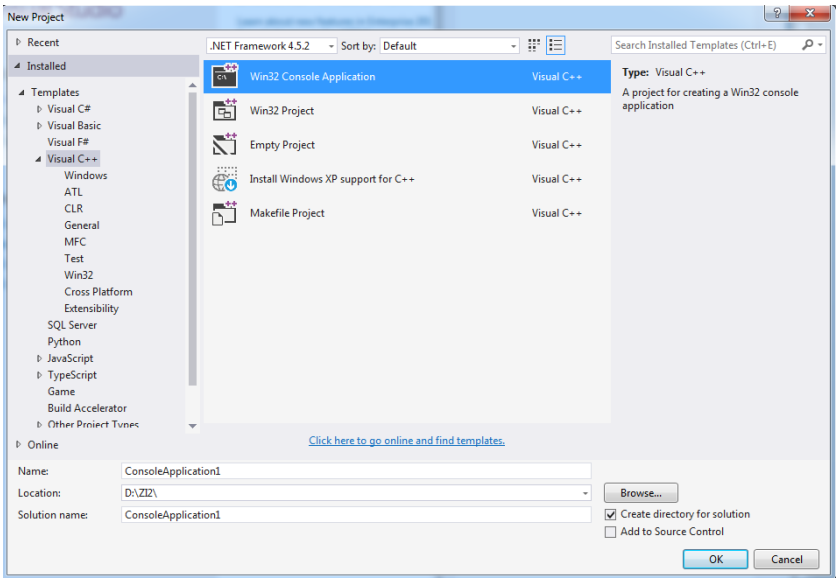

Рисунок 2 – Создание нового проекта

В окне New Project укажите имя проекта, место расположения папки с проектом и имя файла решения Visual Studio. Нажмите OK.

Откроется окно Win32 Application Wizard. В этом окне укажите тип приложения Win32 Console Application для создания консольного приложения и установите флажок Empty Project для создания пустого проекта, не содержащего никаких файлов исходного кода.

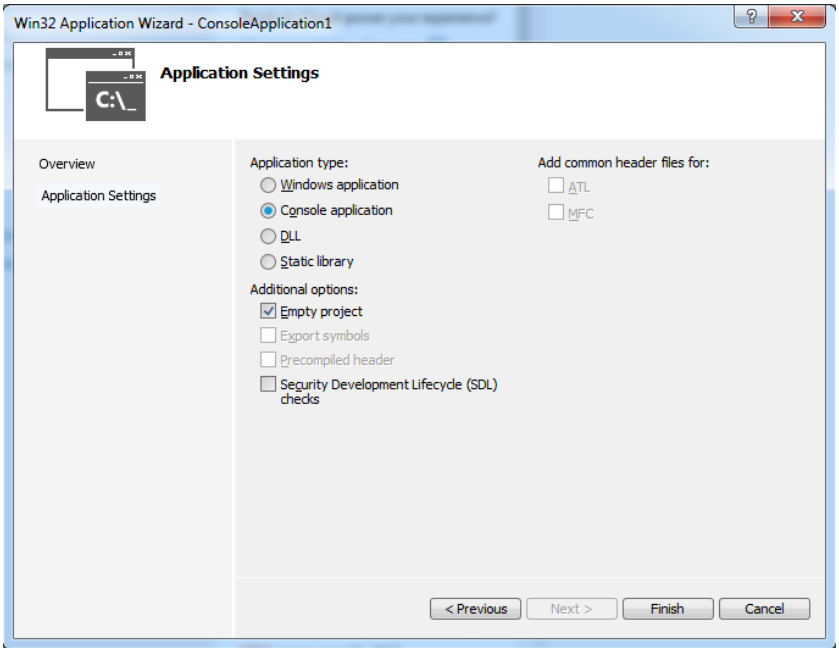

Рисунок 3 – Окно Win32 Application Wizard

В главном окне Visual Studio 2015 В панели Solution Explorer щелкните правой кнопкой мыши на имени проекта и выберите пункт меню Build Customizations.

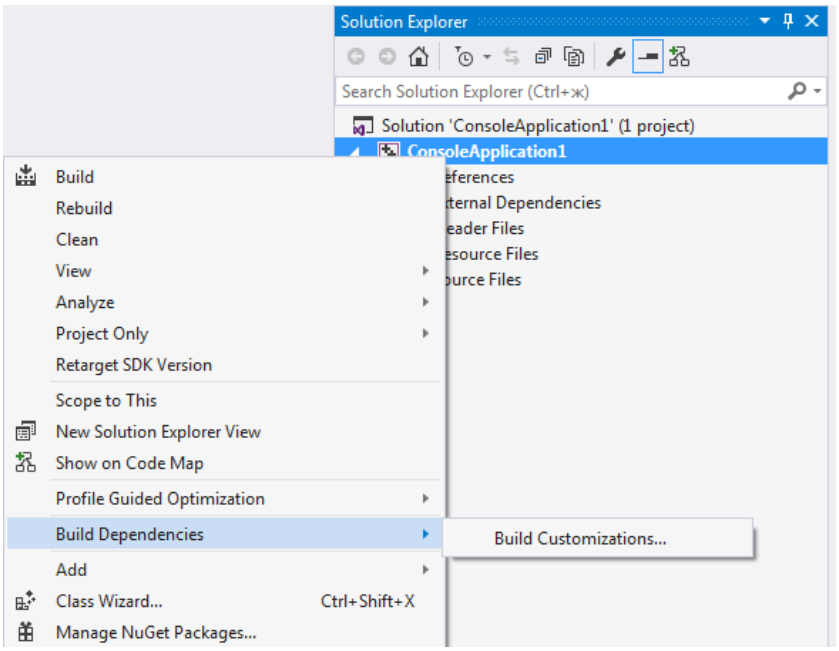

Рисунок 4 – Выбор пункта меню Build Customizations

В открывшемся окне Visual C++ Build Customization Files установите флажок masm(.targets, props). Нажмите ОК. Включение этой настройки позволит подключать к проекту файлы на языке ассемблера и выбирать для них в качестве компилятора Microsoft Macro Assembler.

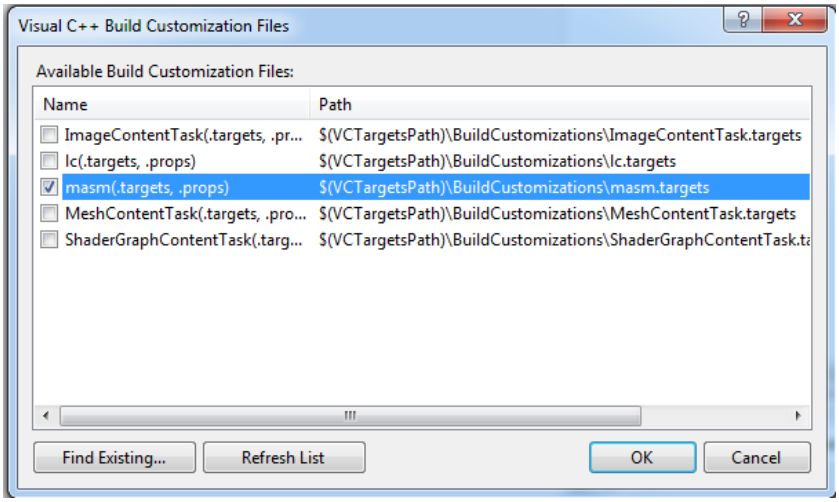

Рисунок 5 – Окно Visual C++ Build Customization Files

Создайте в проекте новый файл исходного кода. Для этого в панели Solution Explorer щелкните правой кнопкой мыши на папке проекте Source Files и выберите пункт меню Add и затем New Item. Откроется окно Add New Item.

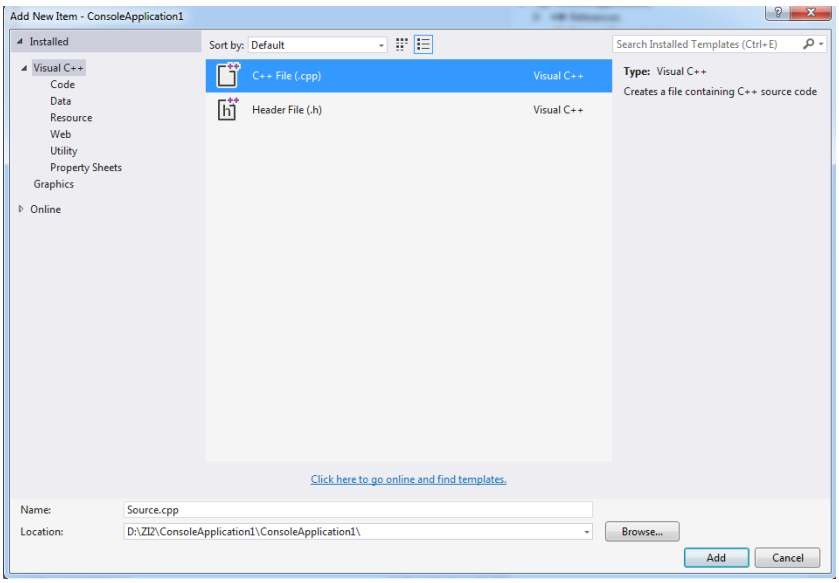

Рисунок 6 – Окно Add New Item

Среда Visual Studio 2015 no умолчанию не содержит шаблона файла для исходного кода на языке ассемблера, поэтому выберем любой подходящий, например, C++ File (.cpp). Введите имя файла main.asm. При вводе имени файла необходимо обязательно указать расширение .asm, это скажет Visual Studio 2015, что используется язык ассемблера. Нажмите Add.

Для созданного файла необходимо указать, что для его трансляции необходимо использовать Microsoft Macro Assembler. В панели Solution Explorer щелкните правой кнопкой мыши на файл main.asm и выберите пункт меню Properties. Откроется окно main.asm Property Pages. В этом окне установите настройку Excluded From Build в No, настройку ItemType в Microsoft Macro Assembler для всех конфигураций.

В файл исходного кода main.asm введите исходный код представленный на рисунке 1.

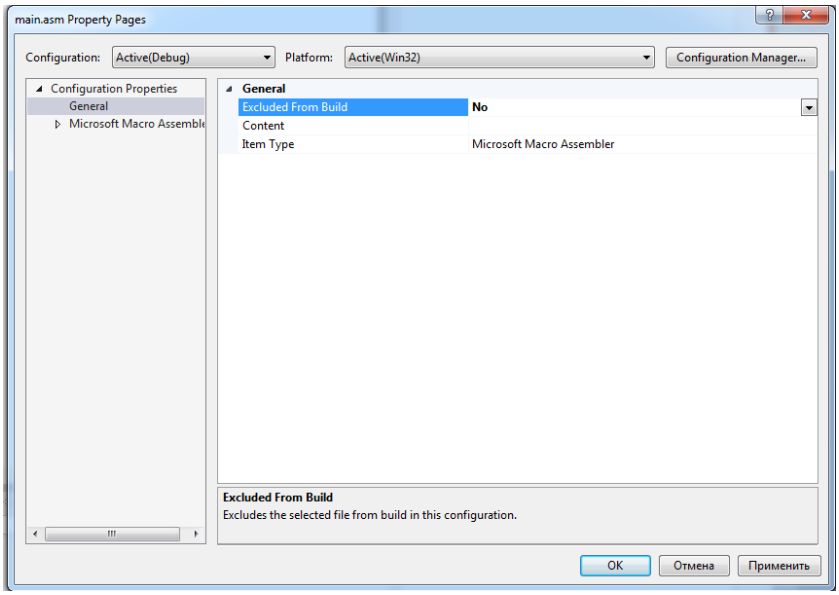

Рисунок 7 – Окно main.asm Property Pages

Запустите созданный проект на выполнение. Для этого в меню Debug выберите пункт Start Debugging. Среда Visual Studio 2010 выполнит компиляцию и сборку проекта, после чего запустит полученный исполняемый модуль на выполнение.

Библиографический список

1. Волков В.А. Запросы в СУБД MySQL // Постулат. 2016. № 11(13). С.33.
2. Волков В.А. Проектирование локальной вычислительной сети учреждения // Постулат. 2016. № 12(14). С.16.
3. Волков В.А. Работа на C++ с MySQL // Постулат. 2016. № 12(14). С.49.
4. Волков В.А. Исторические шифры на примере языка С // Постулат. 2017. № 4.