

Раздельная компиляция

Чтобы программу, написанную на языке C++, запустить, ее нужно преобразовать в эквивалентную, т. е. делающую то же самое, программу на машинном языке. Этот процесс обычно называется сборкой. Программа на машинном языке — это просто набор чисел, который можно записать в оперативную память компьютера и который затем можно выполнить как программу при помощи процессора этого компьютера. Дело в том, что процессор обычно понимает и может выполнить только программы, представляющие собой набор определенных чисел, в котором некоторые числа являются номерами команд процессора, а некоторые указывают на то, какую информацию процессор должен использовать, чтобы выполнить ту или иную команду.

Сборка программы на C++ состоит из нескольких этапов. Сначала каждый исходный файл обрабатывается препроцессором. Препроцессор — это небольшая программа, читающая файл с исходным текстом, и выдающая результирующий файл, также текстовый, как правило, гораздо длиннее исходного. Процесс преобразования исходного файла препроцессором управляется при помощи так называемых директив препроцессора, включаемых непосредственно в исходный файл программы. Препроцессор позволяет включать в программу другие файлы, определять и использовать макроопределения (некоторые правила, в соответствии с которыми одни фрагменты программы нужно заменять на другие), включать в программу или пропускать определенные ее фрагменты (так называемая условная компиляция). Кроме того, имеется специальная конструкция, выглядящая как директива препроцессора, но на самом деле управляющая не препроцессором, а компилятором.

После обработки каждого исходного файла программы препроцессором, полученный файл должен быть скомпилирован, т. е. содержащиеся в нем фрагменты программы на C++ преобразуются в эквивалентные, но записанные на языке ассемблера. Язык ассемблера — это язык, на котором записаны текстовые представления команд процессора. Этот язык, вообще говоря, зависит от компьютера, на котором мы собираемся запускать нашу программу, и от операционной системы.

После компиляции файл с текстом программы на языке ассемблера подается на вход ассемблеру, который превращает его в файл, содержащий представления команд процессора в виде чисел, т. е. в тот вид, который может быть выполнен непосредственно процессором. Результат работы ассемблера называется объектным модулем. В операционной системе Windows объектные модули имеют расширения `.obj`.

Несмотря на то, что в объектном модуле команды процессора записаны в том виде, в котором они могут быть выполнены процессором, запустить объектный модуль на выполнение нельзя. Дело в том, что программа может состоять из нескольких исходных файлов, из которых получается несколько объектных модулей, каждый из которых содержит определенную часть нашей программы.

Для запуска программы нужно еще собрать вместе все объектные модули, из которых она состоит. Программа, которая это делает, называется компоновщиком. Результатом работы компоновщика является готовый к запуску файл программы (в Windows такие файлы имеют расширение `.exe`).

При написании программы на C++ можно также использовать средства стандартной библиотеки C++, а также средства из других библиотек. Библиотека — это архив из некоторых объектных модулей, и если в программе используются средства из какой-то библиотеки, ее объектные модули также должны участвовать в процессе компоновки.

Такая технология программирования (использование отдельно препроцессора, компилятора, ассемблера и компоновщика) называется раздельной компиляцией. Раздельная компиляция широко применяется в программировании. Практически для упрощения процесса получения работающей программы из исходных текстов в большинстве систем программирования используются дополнительные инструменты, как применяемые из командной строки, так и с графическим интерфейсом.

У раздельной компиляции есть ряд преимуществ по сравнению с теми способами написания программ, в которых весь текст программы помещается в один огромный файл:

1) Такая организация программы поддерживает концепцию модульности, когда каждый из исходных файлов отвечает за один небольшой, законченный по смыслу фрагмент программы, что сильно упрощает и ускоряет разработку программ, а также резко упрощает командную разработку, когда над одной программой трудится (большой) коллектив разработчиков.

2) Если изменяется один исходный файл, перекомпилировать нужно только его. Правда, перекомпоновывать нужно все, но компоновка обычно выполняется быстрее компиляции.

3) Продавцы библиотек могут распространять свои библиотеки в скомпилированном виде, что затрудняет извлечение из них применяемых там алгоритмов и структур данных.

4) Наконец, иногда такой подход позволяет писать разные фрагменты программы на разных языках программирования. Здесь, правда, нужно соблюдать определенную осторожность. В качестве примера назовем два возможных препятствия: а) разные способы вызова функций, например, соглашения о том, в каком порядке параметры функции хранятся в памяти, и б) так называемое декорирование имен, т. е. преобразование имен функций на стадии компиляции, так что компоновщик видит не те имена функций, которые указаны в тексте программы, а результаты их преобразования компилятором.

Для языка C++ существует много разных библиотек, в том числе и с открытым исходным кодом. Наиболее

известной из них является boost (<http://www.boost.org/>). Многие из ее частей в том или ином виде становились впоследствии частью стандартной библиотеки.

Что нужно знать в результате (проверяется на зачете):

- Что такое машинный язык?
- Что такое препроцессор?
- Что происходит при компиляции?
- Что такое ассемблер?
- Что такое компоновщик?
- Что такое объектный модуль, его расширение?
- Что обычно содержится в файле с расширением exe (на Windows)?
- Что такое библиотека?
- Преимущества отдельной компиляции.
- Что такое декорирование имен?
- Какие библиотеки для C++ Вы знаете? (стандартная, boost)