ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра Математическая кибернетика и информационные технологии

Реферат по дисциплине:

Теоретические аспекты программирования.

на тему:

Компиляция, основные понятия и классификация.

Выполнил:

студент гр. 3МПП1901

Краснов Д.А.

Принял:

Яшина М.В.

Москва 2020 г.

**Содержание**

1. Введение2
2. Основные понятия 3
3. Виды компиляции5
4. Виды компиляторов7
5. Достоинства и недостатки компиляторов8

Список использованных источников9

Введение.

На заре развития компьютеров первые компиляторы называли «программирующими программами (так как в тот момент программой считался только машинный код, а «программирующая программа» была способна из человеческого текста сделать машинный код, то есть запрограммировать ЭВМ).

Собственно, это отражает тот факт, что компилятор по своей сути являются одним из средств автоматизации процесса программирования компьютеров. Дело в том, что любая написанная нами программа на входном языке (например, на С++ или языке ассемблера) по сути есть скрипт-сценарий для компилятора - непосредственно выполняемая - интерпретируемая - компилятором программа, в которой описывается:

* какую программу компилятор должен построить,
* что и как создаваемая нами таким образом программа должна делать (в том числе какие данные и как обрабатывать).

И уже выполняя - интерпретируя - эту нашу программу на входном языке, компилятор и строит эквивалентную ей программу на машинном языке.

Устройство компилятора:

В основе любого компилятора лежат транслятор и компоновщик. Зачастую при компиляции используется внешний компоновщик, а сам компилятор выполняет лишь функцию трансляции. Случается также, что компилятор реализуется как своеобразная программа-менеджер, связанная с транслятором (или трансляторами, если при написании исходного кода использовались разные языки программирования) и компоновщиком и запускающая их исполнение при необходимости.

Основные понятия.

Компилятор, обрабатывая исходную программу, создает эквивалентную программу на машинном языке, которая называется объектной программой, или объектным кодом.

Объектный код, как правило, записывается в файл, но не обязательно представляет собой готовую к исполнению программу.

**Компилятор** это:

* Программа или **техническое** средство, выполняющее *компиляцию*.
* Машинная программа, используемая для компиляции.
* Программа, переводящая текст программы на языке высокого уровня в эквивалентную программу на машинном языке.
* Программа выполняющая (после трансляции) компоновку программы.

**Компиляция** это:

* перевод программы, на язык, близкий к машинному, и последующая её компоновка.
* перевод программы, составленной на исходном языке, в объектный модуль (осуществляется *компилятором*.) и последующая её компоновка в готовый к использованию программный модуль.

**Как работает компилятор?**

Компилятор, обрабатывая исходную программу, создает эквивалентную программу на машинном языке, которая называется объектной программой, или объектным кодом.

Объектный код, как правило, записывается в файл, но не обязательно представляет собой готовую к исполнению программу. (рис.1)

Компилятор обеспечивает получение быстрой программы на машинном языке, время работы которой намного меньше времени, которое будет затрачено на исполнение этой программы. Однако компиляция, являясь отдельным этапом обработки программы, может потребовать заметного времени, снижая оперативность работы.



Рис.1

Пример компиляции можно увидеть на рис.2.

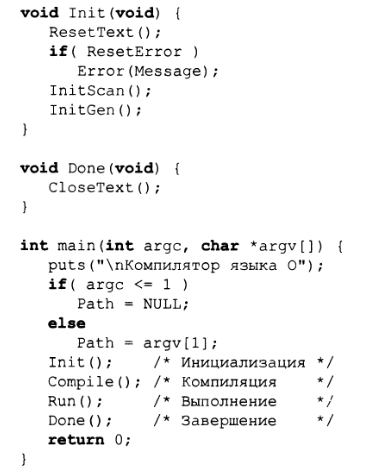
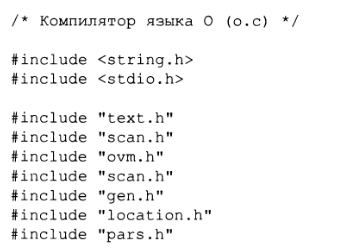


Рис.2Виды компиляторов.

Для удобства программистов при решении различных задач применяются наиболее удобные и приспособленные компиляторы. Если произвести их классификацию, то можно выделить несколько видов компиляторов:

*Векторизующий компилятор* – это утилита, выполняющая трансляцию исходного кода в объектный и адаптированная под вычислительные машины, оснащенные векторными процессорами.

*Гибкий компилятор* запрограммирован на языке высокого уровня по модульному принципу. Его управление осуществляется с помощью таблиц. Возможно также его исполнение с помощью компилятора компиляторов.

*Инкрементальный компилятор* – это языковой процессор, осуществляющий повторное транслирование отдельных фрагментов исходного кода и дополнений к нему. Однако он исключает при этом перекомпиляцию всего приложения.

*Интерпретирующий (пошаговый) компилятор* – это утилита, последовательно выполняющая независимую компиляцию каждого оператора или команды исходного высокоуровневого кода.

*Компилятор компиляторов* – транслятор, способный воспринимать формальное описание для языка программирования и самостоятельно генерировать компилятор для любого языка.

*Отладочный компилятор* позволяет найти и устранить некоторые виды синтаксических ошибок, допущенных при написании исходного кода

*Резидентный компилятор* занимает постоянное место в оперативной памяти и, следовательно, является доступным для повторного использования широким рядом задач.

*Самокомпилируемый компилятор* написан на том же языке, с которого выполняется трансляция.

*Универсальный компилятор* базируется на формальном описании семантических и синтаксических параметров входного языка. Основными компонентами подобной утилиты являются ядро, синтаксический и семантический загрузчики.

Виды компиляции.

*Пакетная* - Компиляция нескольких исходных модулей в одном пункте задания.

*Построчная* - пошаговая независимая компиляция каждого последующего оператора.

*Условная* - Компиляция, при которой транслируемый текст зависит от условий, заданных в исходной программе директивами компилятора. Так, в зависимости от значения некоторой константы, можно включать или выключать трансляцию части текста программы.

Трансляция программы как неотъемлемая составляющая компиляции включает в себя:

[**Лексический анализ**](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7). На этом этапе последовательность символов исходного файла преобразуется в последовательность лексем.

[**Синтаксический (грамматический) анализ**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7)**.** Последовательность лексем преобразуется в древо разбора.

[**Семантический анализ**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7). На этой фазе древо разбора обрабатывается с целью установления его семантики (смысла) — например, привязка идентификаторов к их объявлениям, типам данных, проверка совместимости, определение типов выражений и т. д. Результат обычно называется «промежуточным представлением/кодом», и может быть дополненным древом разбора, новым деревом, абстрактным набором команд или чем-то ещё, удобным для дальнейшей обработки.

[**Оптимизация**](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80%D0%B0)**.** Выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла. Оптимизация может быть на разных уровнях и этапах — например, над промежуточным кодом или над конечным машинным кодом.

[**Генерация кода**](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_%D0%BA%D0%BE%D0%B4%D0%B0). Из промежуточного представления порождается код на целевом машинно-ориентированном языке.

Достоинства и недостатки компиляторов.

Стоит отметить, что скомпилированные приложения обладают большим быстродействием по сравнению с интерпретируемыми, но в то же время полученный в результате компиляции [машинный код](https://yandex.ru/turbo/fb.ru/s/article/351250/mashinnyiy-kod-kak-yazyik-programmirovaniya-yazyik-assemblera?parent-reqid=1604751440301609-253885724493646716100275-production-app-host-sas-web-yp-48&utm_source=turbo_turbo) зависит от аппаратной платформы. Так, программа, написанная и скомпилированная для Windows, не будет работать, например, в Linux. Поэтому в случае интернет-приложений, когда нельзя сказать заранее, в какой среде они будут работать, используют интерпретацию или байт-код (в этом случае исходную программу преобразуют в промежуточный вид, который можно выполнить на различных аппаратных платформах).

**Литература**

* *Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман* Компиляторы: принципы, технологии и инструментарий, 2 издание = Compilers: Principles, Techniques, and Tools, 2nd Edition. — М.: «Вильямс», 2010. — С. 1184. — ISBN 978-5-8459-1349-4
* *Робин Хантер.* Основные концепции компиляторов = The Essence of Compilers. — М.: Вильямс, 2002. — С. 256. — ISBN 0-13-727835-7
* *Хантер Р.* Проектирование и конструирование компиляторов / Пер. с англ. С. М. Круговой. — М.: Финансы и статистика, 1984. — 232 с.
* *В. А. Серебряков, М. П. Галочкин.* Основы конструирования компиляторов - [www.citforum.ru/programming/theory/serebryakov/](http://www.citforum.ru/programming/theory/serebryakov/).
* С.З. Свердлов. - Языки программирования и методы трансляции.2011г.