

Информатика

Мы

15 апреля 2023 г.

Оглавление

1	Программная инженерия	2
1.1	Изменяемость	2
1.2	Этапы компиляции	3
1.2.1	Лексический анализ (сканирование)	3
1.2.2	Синтаксический анализ (парсинг)	3
1.2.3	Семантический анализ	4
1.2.4	Генерация промежуточного кода	4
1.2.5	Оптимизация кода	4
1.2.6	Генерация кода	4

Глава 1

Программная инженерия

Любую программу можно написать на низкоуровневом языке - языке, наиболее приближённом к устройству компьютера. Однако, тем не менее, разрабатывают всё новые языки программирования. Далее рассматриваются причины, по которым возникает нужда в высокоуровневых языках.

1.1 Изменяемость

Программистам в течение жизненного цикла разработки ПО приходится изменять программу. По причине изменений требований к продукту или для устранения ошибки.

Чтобы что-то изменить, нужно найти всю имплементацию этого чего-то в программе. С этим возникают две трудности: имплементация этого чего-то простирается в большой части кода - слишком много приходится править, чтобы внести нужное изменение - и код трудно читаем - сложно понять, где то, что нам нужно.

Языки программирования создают такими, чтобы они как можно более полно решали данные проблемы.

Часто добавляют "мультипарадигменные" конструкции, которые должны матчаться в нашем мозгу с устоявшимися паттернами. Однако всевозможных паттернов настолько много, что данные вводящиеся конструкции только капля в море.

Вторым способом бороться с данными проблемами, который работает всегда, является продолжающийся рефакторинг согласно сложности Джона.

Трансляция кода

Компилятор - это программа, переводящая текст программы с одного языка на другой.

Интерпретатор - это программа, выполняющая код программы, не переводя её на другой язык.

Компоновщик (линкер) - это программа, выполняющая разрешение внешних адресов памяти, по которым код из одного файла может обращаться к информации из другого файла.

Загрузчик - это программа, которая помещает все выполнимые объектные файлы в память для выполнения.

Компиляция состоит из анализа и синтеза.

В течение компиляции код может переводиться по цепочке в несколько промежуточных представлений.

Таблица символов содержит в себе информацию, которая накапливается на протяжении компиляции.

Проход (pass) - это этапы компиляции, преобразующие один файл в другой (необязательно в файл с целевым кодом).

1.2 Этапы компиляции

1.2.1 Лексический анализ (сканирование)

Лексема - это значащая последовательность символов.

Токен - это значение <имя токена, значение атрибута> относительно лексемы, где значение атрибута указывает на запись в таблице символов.

1.2.2 Синтаксический анализ (парсинг)

Синтаксический анализатор структурирует токены в синтаксическое дерево.

1.2.3 Семантический анализ

Семантический анализатор проверяет синтаксическое дерево на корректность.

1.2.4 Генерация промежуточного кода

Генерация кода для абстрактной вычислительной машины.

1.2.5 Оптимизация кода

Оптимизация промежуточного кода.

1.2.6 Генерация кода

Генерация кода на целевом языке.