

Лекция 11. Введение в трансляцию программ

Коновалов А. В.

31 октября 2022 г.

Способы реализации языка программирования

Компьютер исполняет машинный код — последовательность примитивных операций: сложение двух ячеек памяти, пересылка значения из одной ячейки в другую, обращение к устройству, передача управления на другую инструкцию (безусловная или условная — в зависимости от результата предыдущей операции).

Человеку писать машинный код трудоёмко и мучительно. Решение: программист пишет программу на человекочитаемом языке, компьютер её выполняет.

Способы реализации языка программирования: **интерпретация** и **компиляция**.

Способы реализации языка программирования — интерпретация

При **интерпретации** компьютер читает программу на языке программирования и выполняет инструкции, записанные в этой программе.

Преимущества: удобство разработки (нет промежуточной фазы компиляции), возможно, удобство отладки, переносимость.

Недостатки: низкое быстродействие, зависимость от интерпретатора.

Способы реализации языка программирования — компиляция

При **компиляции** компьютер переводит программу с человекочитаемого языка на машинный язык. Транслятор — синоним компилятора.

Преимущества: высокое быстродействие, автономность готовых программ. Недостаток: фаза компиляции, зависимость готовых программ от платформы.

Гибридный подход: компилятор формирует промежуточный более низкоуровневый код, который затем выполняется интерпретатором.

Стадии компиляции

1. Лексический анализ
2. Синтаксический анализ
3. Семантический анализ
4. Генерация промежуточного кода
5. Оптимизации
6. Генерация машинного кода
7. Компоновка

Стадии 1–3 — стадии анализа (front end), стадии 4–7 — стадии синтеза (back end).

Стадии компиляции

Лексический анализ программы — программа делится на «слова» — **лексемы**, некоторые небольшие структурные элементы: знаки операций, идентификаторы, литеральные константы (числа, строки, символы...). На стадии лексического анализа отбрасываются комментарии и символы пустого пространства (пробелы, табуляции, переводы строк).

Лексема — подстрока исходной программы: `)`, `counter`, `007`.

Токен — «обработанная» лексема, токен состоит из метки типа, позиции в исходном файле и атрибута (значения лексемы):
(`'CLOSE-BRACKET (1 1)`), (`'IDENT (2 1) "counter"`),
(`'NUMBER (2 10) 7`).

Стадии компиляции

Синтаксический анализ — принимает последовательность токенов и строит из них синтаксическое дерево.

Последовательность токенов плоская, выход синтаксического анализатора иерархичен — отражает структуру программы.

Семантический анализ — проверяет допустимость операций, правильность имён переменных, функций...

Стадии компиляции

Генерация промежуточного представления. Оптимизации.

Генерация машинного кода. Компоновка. Программа может состоять из отдельно транслируемых файлов, в том числе библиотек — нужно из них собрать единую готовую программу.

Стадии интерпретации

1. Лексический анализ.
2. Синтаксический анализ.
3. Семантический анализ.
4. Исполнение — (интерпретация) построенной программы.