БГТУ, ФИТ, ПОИТ, 2 семестр, Языки программирования

Побочные эффекты

В любом выражении, в котором возможны побочные эффекты существует зависимость от порядка использования переменных в вычислениях.

Пример: a[i] = i++;

Неизвестно какое значение индекса і используется для обращения к массиву старое или новое.

Компиляторы могут по-разному интерпретировать этот оператор и выдавать разные результаты в зависимости от своей интерпретации. В стандарте языка эти вопросы оставлены открытыми на усмотрение компилятора.

; - это элемент завершающий оператор, а не разделитель операторов (как, например, в Pascal).

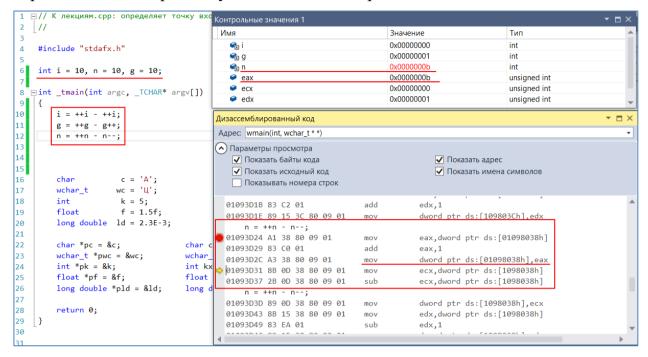
Побочный эффект ++i и i++ состоит в изменении операнда и он будет получен, когда программа перейдет к следующему шагу.

Пример. Вычислить выражение n = ++n - n--;

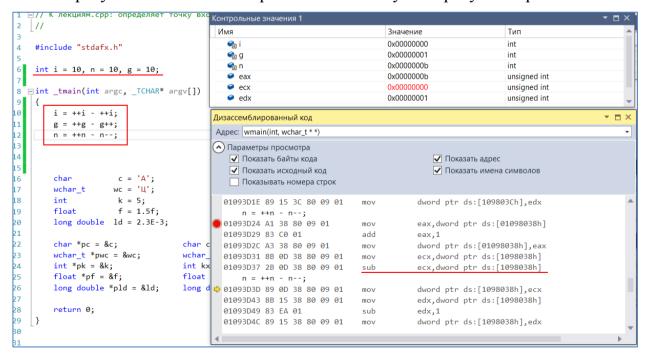
За исключением тех случаев, когда порядок вычисления выражения зафиксирован синтаксическими правилами или указан в **Стандарте** (для операции вызова функции (), операций «&&», «||» «?:» «,»), порядок вычисления подвыражений и порядок возникновения побочных эффектов не уточняется. Выражение, содержащее более чем одно вхождение одной и той же коммутативной и ассоциативной бинарной оператора (*, +, &, ^, |), может перегруппировываться.

Дизассемблированный код.

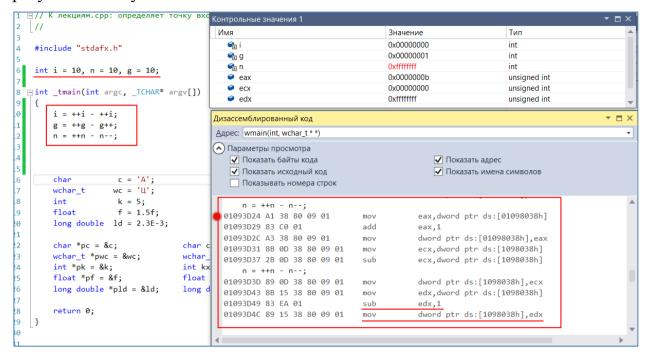
Операции выполняются последовательно. Первая операция — инкремент. После выполнения команды mov одержимое регистра еах перемещается в переменную n и становится равным 10:



Следущая операция (вторая) – вычитание из полученного на этот момент результата значения переменной п. Получаем результат равный 0:



Третья операция – постфиксный инкремент (подчеркнуто). И в результате получаем -1:



Пример. Вычислить выражение i = ++i - ++i;

Последовательность операций:

- 1) префиксный инкремент (результат і = 11),
- 2) префиксный инкремент (результат i = 12),
- 3) вычитание (результат i = 0).

