

### 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  $i$  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 `eax` перемещается в переменную `n` и становится равным 10:

Контрольные значения 1

Имя	Значение	Тип
i	0x00000000	int
g	0x00000001	int
n	0x0000000b	int
eax	0x0000000b	unsigned int
ecx	0x00000000	unsigned int
edx	0x00000001	unsigned int

Дизассемблированный код

Адрес: `wmain(int, wchar_t**)`

Параметры просмотра

- ☒ Показать байты кода
- ☒ Показать исходный код
- ☐ Показывать номера строк
- ☒ Показать адрес
- ☒ Показать имена символов

Адрес	Оpcodes	Мnemonic	Comment
01093D1B	83 C2 01	add	edx,1
01093D1E	89 15 3C 80 09 01	mov	dword ptr ds:[109803Ch],edx
01093D24	A1 38 80 09 01	mov	eax,dword ptr ds:[01098038h]
01093D29	83 C0 01	add	eax,1
01093D2C	A3 38 80 09 01	mov	dword ptr ds:[01098038h],eax
01093D31	8B 0D 38 80 09 01	mov	ecx,dword ptr ds:[1098038h]
01093D37	2B 0D 38 80 09 01	sub	ecx,dword ptr ds:[1098038h]
01093D3D	89 0D 38 80 09 01	mov	dword ptr ds:[1098038h],ecx
01093D43	8B 15 38 80 09 01	mov	edx,dword ptr ds:[1098038h]
01093D49	83 EA 01	sub	edx,1

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

Контрольные значения 1

Имя	Значение	Тип
i	0x00000000	int
g	0x00000001	int
n	0x0000000b	int
eax	0x0000000b	unsigned int
ecx	0x00000000	unsigned int
edx	0x00000001	unsigned int

Дизассемблированный код

Адрес: `wmain(int, wchar_t**)`

Параметры просмотра

- ☒ Показать байты кода
- ☒ Показать исходный код
- ☐ Показывать номера строк
- ☒ Показать адрес
- ☒ Показать имена символов

Адрес	Оpcodes	Мnemonic	Comment
01093D1E	89 15 3C 80 09 01	mov	dword ptr ds:[109803Ch],edx
01093D24	A1 38 80 09 01	mov	eax,dword ptr ds:[01098038h]
01093D29	83 C0 01	add	eax,1
01093D2C	A3 38 80 09 01	mov	dword ptr ds:[01098038h],eax
01093D31	8B 0D 38 80 09 01	mov	ecx,dword ptr ds:[1098038h]
01093D37	2B 0D 38 80 09 01	sub	ecx,dword ptr ds:[1098038h]
01093D3D	89 0D 38 80 09 01	mov	dword ptr ds:[1098038h],ecx
01093D43	8B 15 38 80 09 01	mov	edx,dword ptr ds:[1098038h]
01093D49	83 EA 01	sub	edx,1
01093D4C	89 15 38 80 09 01	mov	dword ptr ds:[1098038h],edx

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

The screenshot shows a C++ IDE with a source file named `K лекциям.cpp: определяет точку входа`. The code defines variables `i`, `g`, and `n` as `int` with initial values of 10. It also defines `eax`, `ecx`, and `edx` as `unsigned int` with initial values of 0. The `_tmain` function contains three lines of code: `i = ++i - ++i;`, `g = ++g - g++;`, and `n = ++n - n--;`. The third line is highlighted with a red box. Below the code, there is a table of variables and their values: `i` is 0x00000000, `g` is 0x00000001, `n` is 0xffffffff, `eax` is 0x0000000b, `ecx` is 0x00000000, and `edx` is 0xffffffff. The assembly window shows the disassembled code for the `wmain` function. The assembly code for the third line is highlighted with a red box: `01093D24 A1 38 80 09 01 mov eax,dword ptr ds:[01098038h]`, `01093D29 83 C0 01 add eax,1`, `01093D2C A3 38 80 09 01 mov dword ptr ds:[01098038h],eax`, `01093D31 8B 0D 38 80 09 01 mov ecx,dword ptr ds:[1098038h]`, `01093D37 2B 0D 38 80 09 01 sub ecx,dword ptr ds:[1098038h]`, `01093D3D 89 0D 38 80 09 01 mov dword ptr ds:[1098038h],ecx`, `01093D43 8B 15 38 80 09 01 mov edx,dword ptr ds:[1098038h]`, `01093D49 83 EA 01 sub edx,1`, and `01093D4C 89 15 38 80 09 01 mov dword ptr ds:[1098038h],edx`.

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

Последовательность операций:

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

The screenshot shows a C++ IDE with a source file named `K лекциям.cpp: определяет точку входа`. The code defines variables `i`, `g`, and `n` as `int` with initial values of 10. It also defines `eax`, `ecx`, and `edx` as `unsigned int` with initial values of 0. The `_tmain` function contains three lines of code: `i = ++i - ++i;`, `g = ++g - g++;`, and `n = ++n - n--;`. The first line is highlighted with a red box. Below the code, there is a table of variables and their values: `i` is 0x0000000c, `g` is 0x0000000a, `n` is 0x0000000a, `eax` is 0x0000000b, `ecx` is 0x0000000c, and `edx` is 0x00000000. The assembly window shows the disassembled code for the `wmain` function. The assembly code for the first line is highlighted with a red box: `01093CBC F3 AB rep stos dword ptr es:[edi]`, `01093CBE A1 00 80 09 01 mov eax,dword ptr ds:[01098000h]`, `01093CC3 33 C5 xor eax,ebp`, `01093CC5 89 45 FC mov dword ptr [ebp-4],eax`, `01093CC8 A1 34 80 09 01 mov eax,dword ptr ds:[01098034h]`, `01093CCD 83 C0 01 add eax,1`, `01093CD0 A3 34 80 09 01 mov dword ptr ds:[01098034h],eax`, `01093CD5 8B 0D 34 80 09 01 mov ecx,dword ptr ds:[1098034h]`, `01093CDB 83 C1 01 add ecx,1`, `01093CDE 89 0D 34 80 09 01 mov dword ptr ds:[1098034h],ecx`, `01093CE4 8B 15 34 80 09 01 mov edx,dword ptr ds:[1098034h]`, `01093CEA 2B 15 34 80 09 01 sub edx,dword ptr ds:[1098034h]`, and `01093CF0 89 15 34 80 09 01 mov dword ptr ds:[1098034h],edx`.