

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++` заключается в изменении операнда и он будет получен, когда программа перейдет к выполнению следующего шага.

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 одного и того же операнда в коммутативном и ассоциативном бинарном операторе (`*`, `+`, `&`, `^`, `|`), может перегруппировываться.

Дизассемблированный код.

Операции выполняются последовательно.

Проанализируем в дизассемблере, как выполняется 12 строка исходного кода. Первая операция – инкремент. В точке останова после выполнения команды `mov` (подчеркнуто) содержимое регистра `eax` перемещается в переменную `n` и становится равным 10:

The screenshot shows a debugger window with the following components:

- Source Code:** Lines 1-31 of `лекциям.cpp`. Line 12 is highlighted: `n = ++n - n--;`. A red box highlights the increment part: `i = ++i - ++i;`.
- Контрольные значения 1:** A table showing the state of registers and variables.

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:** A list of instructions. The instruction at address `01093D24` is highlighted: `mov eax,dword ptr ds:[01098038h]`. A red box highlights the subsequent instructions: `add eax,1`, `mov dword ptr ds:[01098038h],eax`, `mov ecx,dword ptr ds:[1098038h]`, and `sub ecx,dword ptr ds:[1098038h]`.

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

The screenshot shows the same debugger window as before, but with the second instruction highlighted:

- Source Code:** Line 12 is still highlighted: `n = ++n - n--;`. The red box now highlights the decrement part: `g = ++g - g++;`.
- Контрольные значения 1:** The state of registers and variables is updated.

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:** The instruction at address `01093D24` is still highlighted: `mov eax,dword ptr ds:[01098038h]`. The subsequent instruction at address `01093D29` is now highlighted: `add eax,1`. A red box highlights the instruction at address `01093D37`: `sub ecx,dword ptr ds:[1098038h]`.

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

The screenshot displays a C++ IDE with the following components:

- Source Code:** A C++ program with variables `i`, `g`, `n`, `eax`, `ecx`, and `edx`. A red box highlights the line `n = ++n - n--;` in the `_tmain` function.
- Контрольные значения 1 (Control Values 1):** A table showing the current values of registers and variables.
- Дизассемблированный код (Disassembled Code):** A window showing the assembly instructions corresponding to the highlighted source code line.

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

Дизассемблированный код (выделенная строка):

```
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]
n = ++n - n--;
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
```

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

Последовательность операций:

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

Контрольные значения 1

Имя	Значение	Тип
i	0x0000000c	int
g	0x0000000a	int
n	0x0000000a	int
eax	0x0000000b	unsigned int
ecx	0x0000000c	unsigned int
edx	0x00000000	unsigned int

Дизассемблированный код

Адрес: wmain(int, wchar_t **)

Параметры просмотра

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

Адрес	Оpcodes	Comment
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:[01098034h]
01093CDB	83 C1 01	add ecx,1
01093CDE	89 0D 34 80 09 01	mov dword ptr ds:[01098034h],ecx
01093CE4	8B 15 34 80 09 01	mov edx,dword ptr ds:[01098034h]
01093CEA	2B 15 34 80 09 01	sub edx,dword ptr ds:[01098034h]
01093CF0	89 15 34 80 09 01	mov dword ptr ds:[01098034h],edx