

## Re: Function call evaluation order

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2006-03/msg01059.html>

---

- *From:* "Tom Widmer [VC++ MVP]" <tom\_usenet@xxxxxxxxxxxx>
  - *Date:* Fri, 24 Mar 2006 18:23:34 +0000
- 

Frederico Pissarra wrote:

And think about the standard strcpy() function... It is implemented in C like this:

```
char *strcpy(char *dest, char *src)
{
    char *sTemp = *dest;
    while (*dest++ = *src++);
    return dest;
}
```

if the post increments are supposed to be evaluated sequentially the routine will never work!

Of course it will! dest and src are different objects. We're talking about modifying the same object multiple times without intervening sequence points.

What do you think should be printed in the following two cases:

```
#include <iostream>
int main()
{
    std::cout << "post\n";
    int i = 0;
    std::cout << i++ << ' ' << i++ << ' ' << i++ << '\n';

    std::cout << "pre\n";
    i = 0;
    std::cout << ++i << ' ' << ++i << ' ' << ++i << '\n';
}
```

I get different results with and without optimization on VC7.1! The point is that precedence only imposes a \*partial\* ordering – beyond that ordering (that the first thing be output before the second), everything else can happen in any order, and because the same object is being modified, it is undefined (as opposed to unspecified) behaviour in any case. Again, read the C faq links I posted.

If you still don't understand after everything you've seen, never mind!

Re: Function call evaluation order

Tom

.