

Re: Memory leak

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-05/2169.html

From: Karl Heinz Buchegger (*kbuchegg_at_gascad.at*)

Date: 05/17/04

Date: Mon, 17 May 2004 11:19:11 +0200

John wrote:

```
>
> Does it mean the number of "new" is equal to the number of "delete" in
> a code?
> I think there might be exceptions, for example,
>
> void f(X* x0)
> {
> X* x1 = new X;
> X* x2 = new X;
> X* x3 = new X;
> ...
> x1 = x0;
> x2 = x0;
> x3 = x0;
> }
>
> main()
> {
> ....
> f(x0);
> ....
> f1(x0);
>
> }
>
> In the above code, x0 exists outside f(). If I "delete" anyone of x1,
> x2 and x3 before the end of f(), x0 will be released. So f1() will not
> run correctly.
> So there is less "delete" than "new".
> Am I right?
```

When you run this code, the following is happening:

in main there exists x0

```
x0
+-----+
```



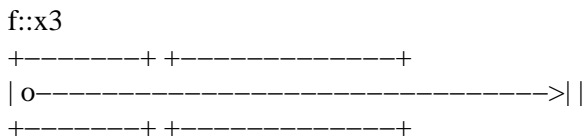
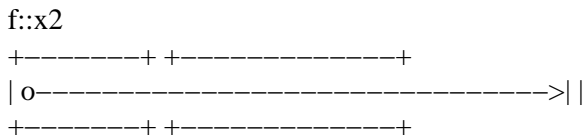
then comes the function call to f.
 In f new variables are created and initialized

```

X* x1 = new X;
X* x2 = new X;
X* x3 = new X;

```

That means:

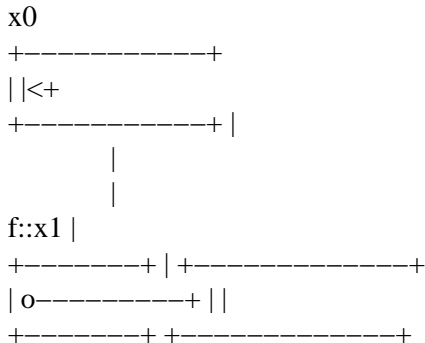


And now you do:

```

x1 = x0; // x0 beeing the address of variable x0 in main

```



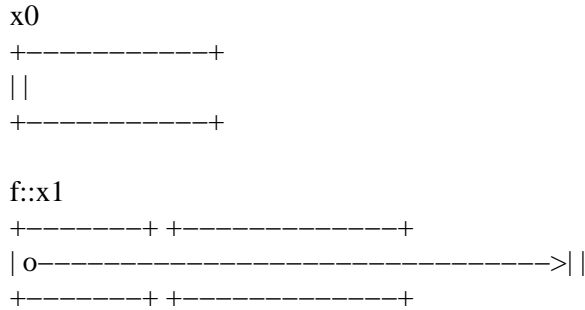
You are certainly right, that you must not do a delete on x1, since this will try to free the memory occupied by variable x0. But look at the lonely rectangle. There is no pointer pointing to it anymore, which means you have no way of freeing it again: you created a memory leak.

The sequence:

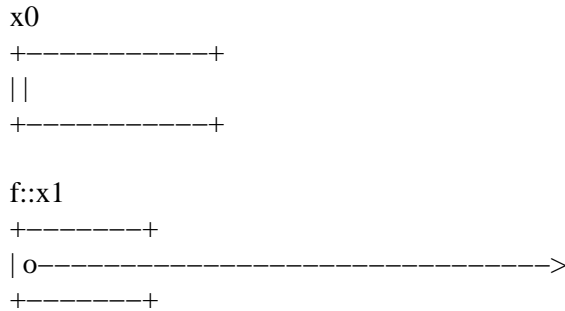
```
delete x1;
x1 = x0;
```

on the other hand will do:

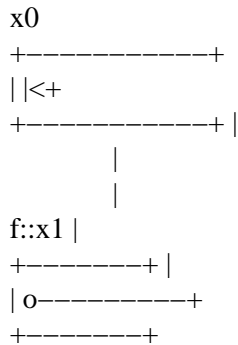
This was the starting situation after the allocations



delete x1



x1 = x0;



and as you can see, there is no longer a rectangle sitting in memory with no pointer to it: no memory leak.

So the correct sequence in your example would be:

```
void f(X* x0)
{
```

```
X* x1 = new X;  
X* x2 = new X;  
X* x3 = new X;  
...  
delete x1;  
x1 = x0;  
  
delete x2;  
x2 = x0;  
  
delete x3;  
x3 = x0;  
}
```

And as you can see, at runtime there is exactly one delete for every new.

```
--  
Karl Heinz Buchegger  
kbuchegg@gascad.at
```