

Re: C++, kernel mode, NDIS

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2004-02/0373.html>

From: Don Burn (burn_at_stopspam.acm.org)

Date: 02/06/04

Date: Fri, 6 Feb 2004 15:47:16 -0500

Actually, this does not work, since the compiler will call delete without for some things. So you always new with the tag and the tag says you must free with the tag, then the compiler frees without the tag for a blue screen! I believe Microsoft is looking in to the question of object orientation in the kernel, but I'm not sure you will ever see C++ blessed, it is too easy for the novice to think C++, so I will use RTTI and STL, and other things that should never live in the kernel.

--

```
Don Burn (MVP, Windows DDK)
Windows 2k/XP/2k3 Filesystem and Driver Consulting
Remove StopSpam from the email to reply
"Vitaly" <anonymous@discussions.microsoft.com> wrote in message
news:04301907-E391-4CDE-B951-3844231DD8F1@microsoft.com...
> Re: delete() with parameters, as any C++ method it allows overloading;
consider:
>
> void * __cdecl operator new(size_t Size, POOL_TYPE Type, ULONG Tag)
> {
>     return ExAllocatePoolWithTag(Type, Size, Tag);
> }
>
> void __cdecl operator delete(void* p, ULONG Tag)
> {
>     if (p) ExFreePoolWithTag(p, Tag);
> }
>
> IMHO: with enough effort and care C++ can be tuned up to be an efficient
tool for kernel development. The C++ side effects (ones you mentioned and
many others) could be taken care of by a kernel C++ subset/library, endorsed
and thoroughly tested by an entity setting standards in Windows driver
development... Until this happens, C++ in the kernel will remain a realm of
few enthusiasts. IHMO,
>
> - Vitaly
>
>
> ----- Don Burn wrote: -----
>
>     Actually you can use C++ in the kernel, and there have been many wars
on
>     newsgroups on whether you should. For a simple environment for C++
handling
>     see the "C++ kernel runtime environment" from
```

microsoft.public.development.device.drivers: Re: C++, kernel mode, NDIS

[http://www.hollistech.com/.](http://www.hollistech.com/)

```
>
> WARNING MY HUMBLE OPINION STARTING
>
> The problem with C++ in the kernel is not if you should, but how to
> constrain the world if you do. C++ is a powerful language that can
generate
> a lot of code behind your back (I used to run a code generator group
for a
> major compiler firm, so been there and done that). This represents a
> problem for the driver developer since it is easy for the compiler to
do
> things you didn't intend, so of the biggies are:
>
> 1. Surprise copy constructors, where allocations and copies
occur
> without it being obvious
> 2. Large stack usage, remember the kernel has a very limited
stack
> 3. Code generation at unexpected times, this is a variant of the
the
> copy constructor, but can really nail you when you use pageable code
> sections in the kernel.
> 4. Interesting potential interactions between C++ exception
handling
> and structured exception handling.
>
> One other major complaint I have is that I now use
ExFreePoolWithTag
> whenever possible to add the check that I am freeing what I think,
since C++
> doesn't allow parameters on destructors I cannot do this.
>
> I try never to use C++ in the kernel, except as a better C.
Other
> folks including parts of Microsoft do use it (though in Microsofts
case I
> understand they check the generated code quite thoroughly).
>
> MY HUMBLE OPINION OFF
>
> It is your choice, there are some interesting decisions to
make.
>
>
> --
> Don Burn (MVP, Windows DDK)
> Windows 2k/XP/2k3 Filesystem and Driver Consulting
> Remove StopSpam from the email to reply
>
> "Peter" <pskvarka@softinengines.com> wrote in message
> news:%232wPf%23H7DHA.2576@TK2MSFTNGP11.phx.gbl...
> > Question about C++ and NDIS
> >> Hi,
> >> I have some experiences with developing
> > TDI, Protocol and Intermediate drivers.
> >> In all my drivers I have used C language.
> > For me it is known some guidelines relating to writing
> > kernel-mode drivers which I read in DDK doc.
> > For example: Use ANSI C, don't use any standard C runtime library
> > function that maintains state,....etc.
> >> My question is:
```

Re: C++, kernel mode, NDIS

microsoft.public.development.device.drivers: Re: C++, kernel mode, NDIS

```
> > What is wrong with C++ in kernel-mode ?
> > I red some older articles in MSDN where are examples,
> > where are defined new operator, delete operator with help of
> > kernel-mode memory allocation etc.
> >> I heart something about:
> > "Kernel can not process code compiled with C++ compiler normal way"
> > but If it is true I dont understand why.
> >>> What is wrong with C++ in Kernel mode ?
> > Especially what is wrong with C++ in NDIS drivers ?
> >>> Thanks !
> > Peter
> >>>
```