

Re: win32 or native NT windows API

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2006-06/msg00303.html>

- *From:* "Julienne Walker" <happyfrosty@xxxxxxxxxxx>
 - *Date:* 14 Jun 2006 10:47:05 -0700
-

Herbert Kleebauer wrote:

Julienne Walker wrote:

Herbert Kleebauer wrote:

It doesn't work like that. Even the native NT API requires you to link with ntdll.dll. Somewhere along the line you need to access a DLL to use the system interface.

The call of OS functions is done by an INT or SYSCALL and not by a call to a function in a DLL. Any code in the DLL is executed in the context of the running program and there is no difference whether you call this code in the DLL or you include the source of this code in your own program. And this is the reason why the old DOS int21 (or the Linux int80) interface is much more appropriate for learning assembly programming than the DLL calls in Windows.

That's BS.

What is BS?

Bullshit.

There's a huge difference

difference between what?

Difference between calling the stub and copying the code inside the stub.

because not only are the indices
into the descriptor table undocumented,

which descriptor table?

they're possibly variant. That
means that even if you copy the stub code into your program, it's not
guaranteed to work on different versions of Windows whereas the stub
is.

If you change the OS interface, then you can't expect that your old
programs still work. Therefore in DOS the OS interface wasn't modified
but only extended and in Windows there is a abstraction layer (the
system DLL's) which have to be modified with each modification
of the OS interface and which provides an identical interface to
the user program. You have to use the abstraction layer is you want
a user program which runs on different Windows versions, but any
abstraction layer is a bad thing if you want to understand what
really happens.

And once you know what really happens, why bother when you can have a
convenient interface that's more portable? I'm having trouble
understanding what your point is. First you advocate hardcoding crap
that will likely break on any other Windows version, now, after being
called on it, you seem to be changing your perspective.

int 21h is completely different from a system call because it's an
emulation feature. The functionality is handled through the ntdm
subsystem, which employs DLL function calls. Using int 21h to justify a
dangerous and unnecessary practice of trying to call the kernel
directly in Windows is just plain stupid.

Re: win32 or native NT windows API

Don't understand what you want to say. DOS (I mean the real DOS) is not much more than a collection of interrupt routines and a simple file system. If you are speaking of "DOS in Windows", than this is nothing than an emulation of the real DOS, so you only call the emulated DOS OS (emulated by the real OS Windows) and this has nothing to do with a call to the hosting Windows OS (which only indirectly is called to emulate DOS).

That's exactly my point. You were using a DOS emulation feature to justify something completely different. And yes, I'm speaking of "DOS in Windows" because this thread is about Windows, not DOS.

.