

Re: GetCurrentProcessorNumber() for XP?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2006-05/msg00391.ht>

- *From:* "Alexander Grigoriev" <alegr@xxxxxxxxxxxxxx>
 - *Date:* Sun, 14 May 2006 07:01:46 -0700
-

Well, if you set the affinity, you know on what process you'll be running and then there's no point whatsoever to query the processor number!

On the other hand, I think the problem being discussed (custom CPU meter of an multimedia app) is not worth bothering. That's one of bells and whistles without real return on investment, other than bragging rights of the programmer.

"Skywing" <skywing_NO_SPAM_@xxxxxxxxxxxxxxxxxxxxxx> wrote in message news:O5suzsvdGHA.1204@xxxxxxxxxxxxxxxxxxxxxx

The former can be easily solved by taking the process affinity mask and then in sequence setting a thread to be assigned to only one processor, executing the appropriate cpuid instruction, then changing the affinity to the next processor.

"Steve Dispensa" <dispensa@xxxxxxxxxxxxxxxxxxxxxx> wrote in message news:C08BBADB.AC51%dispensa@xxxxxxxxxxxxxxxxxxxxxx

On 5/12/06 12:34 PM, in article D203BD13-2430-4A4B-9321-D0E8076B1A71@xxxxxxxxxxxxxx, "Ron Kuper" <RonKuper@xxxxxxxxxxxxxxxxxxxxxx> wrote:

Thanks for the help, everyone. I figured out a way to do it.

The CPUID instruction, called with EAX=1, returns the APIC ID in the upper 8 bits of EBX. This isn't the processor number, but it is unique for each logical CPU. I simply take that number and use a tiny map (implemented as 256 byte array) to map APIC ID -> processor number, i.e., if I don't see an

Re: GetCurrentProcessorNumber() for XP?

APIC ID in the map I increment a processor number counter and assign it to the APIC ID. Does the trick, all in user mode, and quite efficient.

I'm sure you understand this already, but just for the record, the results of your CPUID call can be wrong by the very next CPU instruction. There is nothing stopping the OS from context switching away and scheduling you elsewhere next time you come up for cycles. This will probably be right a lot of the time, but it will be wrong some of the time.

Also, this is obviously not terribly portable or easy to maintain (no inline ASM on x64, etc).

Otherwise, interesting solution.

-Steve