

When hyperthreading is enabled, all the processors are virtual

devblogs.microsoft.com/oldnewthing/20051216-10

December 16, 2005



Raymond Chen

A common problem when answering technical questions is that people sometimes ask a question that can't or shouldn't be answered because it is based upon a misunderstanding. What's particularly frustrating is when they **insist** that you answer their question as posed, even when you try to explain to them that their question is itself flawed.

It's as if somebody asked you the question, "Do I have to use the remote control to lock my kangaroo?" You could answer the question literally ("No"), but the person asking the question would walk away with the wrong conclusion ("Wow, kangaroos are self-locking!"). [Robert Flaming](#) recalls [a similar analogy I made with balsa wood and nails](#).

Here's an example of a question that betrays misunderstanding.

I just enabled hyperthreading on my dual-Xenon machine, and Task Manager now shows four processors instead of two. Which of them are the physical processors and which are the virtual ones?

When you turn on hyperthreading, each individual physical processor acts as if it were two virtual processors. From Task Manager's point of view, the computer has four virtual processors. The two virtual processors associated with each physical processor are completely equivalent. It's not like one is physical and one is virtual. They are both virtual and compete equally for a share of the one physical CPU. When you set processor affinities, you set them to virtual processors.

To find out which virtual processors are associated with the same physical processor, you can call [the `GetLogicalProcessorInformation` function](#).

[Raymond Chen](#)

Follow

