

# Re: Win32 Thread Interleaving

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2007-01/msg00238.htm>

---

- *From:* "Michael K. O'Neill" <[MikeAThon2000@xxxxxxxxxxxxxxxxxxxxxx](mailto:MikeAThon2000@xxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 19 Jan 2007 16:50:08 -0800
- 

"Andrew\_zep" <[Andrew\\_zep@xxxxxxxxxxxxxxxxxxxxxx](mailto:Andrew_zep@xxxxxxxxxxxxxxxxxxxxxx)> wrote in message [news:B9FA9B77-F9E6-4F03-B4FC-0FA435BC1DD6@xxxxxxxxxxxxxxxxxxxxxx](mailto:news:B9FA9B77-F9E6-4F03-B4FC-0FA435BC1DD6@xxxxxxxxxxxxxxxxxxxxxx)

< snip >

Now problem we are facing is, in windows XP hyper threaded processor, threads get the mutexes in just a few milliseconds and never had any

problem

with thread interleaving, threads releases after few milliseconds and

other

thread acquires mutex in very little time. But we are seeing different behaviour in windows 2003 server (SP1) hyper threaded machine. Thread

doesn't

interleave perfectly like XP, sometimes a thread runs for long time

compare

to other thread (I used timers and found out that Thread B gets most of

the

processor time compare to thread A.)

You might be seeing this difference because of the difference in the way that Server editions schedule thread time-slices as opposed to the way that XP does.

## Re: Win32 Thread Interleaving

Server editions schedule long, fixed time-slices that favor background tasks. This makes sense for a server, which presumably is running CPU-hungry code that wants as few context switches as possible.

XP and other home editions schedule short, variable time slices that favor GUI applications.

See articles like "Chapter 6: Processes, Threads, and Jobs" of "Inside Microsoft® Windows® 2000, Third Edition" at <http://www.microsoft.com/mspress/books/sampchap/4354c.aspx> . In particular, look at the section entitled "Controlling the Quantum"

Also, "Win2K Quantums" at <http://www.microsoft.com/technet/sysinternals/information/Windows2000Quantums.msp> which formerly was published at the SysInternals site. Both of these sites were written by Mark Russinovich.

On your Server machine, try adjusting the settings in the "Performance" tab "System Properties", to see if you can see a change in the way your program operates.

And often (most consistently and probably root of all these problems)

thread

A waits for very long time just for acquiring mutex (sometimes more than a

min

some time just a second).

< snip >

As a design note, synchronization objects (like the critical section that you are using) should be held for only the briefest time possible, so that there is not contention between threads.

Your code shows that thread B has a lock on the critical section almost continuously, and only unlocks the critical section for a moment when it falls out of the bottom of the "while(true)" loop. Under these circumstances, it's no surprise that thread A almost never gets an opportunity to run.

Your performance will be even worse on a true multi-processor machine. You should re-design your code.

Re: Win32 Thread Interleaving