

Re: IOCP question: ordering of data

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2007-06/msg00162>

- *From:* "Arkady Frenkel" <arkadyf@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 19 Jun 2007 10:05:04 +0300
-

A good example of such is iocp sample in PSDK which use additional(s) send(s) in advance in the same way as receive
Arkady

"Skywing [MVP]" <skywing_NO_SPAM_@xxxxxxxxxxxxxxxxxxxx> wrote in message
news:uxIUyKhsHHA.4916@xxxxxxxxxxxxxxxxxxxx

It can do partial completion immediately for nonblocking-sockets mode.
For blocking sockets or for overlapped operation, it's all or nothing.

Ken Johnson (Skywing)
Windows SDK MVP
<http://www.nynaeve.net>
"Alexander Nickolov" <agnickolov@xxxxxxxx> wrote in message
news:unFwtaesHHA.4572@xxxxxxxxxxxxxxxxxxxx

Data sequencing synchronization is up to you.

On a side note, WSASend will never return with a partial success – it's all or nothing. This doesn't even make sense for overlapped I/O though, since your send is not completed yet. It's the completion notification that's all or nothing. But, if you do get a partial completion, that means all subsequent sends will complete with failure (the socket is toast).

=====
Alexander Nickolov
Microsoft MVP [VC], MCSO
email: agnickolov@xxxxxxxx
MVP VC FAQ: <http://vcfaq.mvps.org>
=====

"Ulrike Urban" <ULUR@xxxxxxxxxxxx> wrote in message
news:5dn3o7F32o847U1@xxxxxxxxxxxxxxxxxxxx

Hi,

Re: IOCP question: ordering of data

I'm messing around with TCP programming using iocps (XP Pro)

If multiple iocp-worker threads are used, how data-corruption can be avoided.

Example:

– Thread #1 WSASends a block of data.
This send could not completely be done by winsock, so the completion notification comes back with according byte counts, (bytes_sent < bytes_to_be_sent).

Thread #1 tries to WSASend the rest of the data-block

– just before issuing the WSASend, Thread #1 is preempted by Thread #2
Thread #2 WSASends another block of data.
Here data corruption may/will occur.

In such a scenario (it is never guaranteed that winsock is able to send all data of a WSASend-Call, the app may be required to re-send the rest)
how can multiple threads work properly?
My understanding is that if I issue only ONE WSASend on one socket, data integrity can be guaranteed.
This implies that multiple sends issued from multiple threads to one socket does not make sense. Multiple sends from one thread would be ok.
But then there is no advantage using multiple threads...