

Re: Starting another asynchronous I/O in IOCP worker thread causes stack corruption

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2007-04/msg00039.htm>

- *From:* "Alexander Grigoriev" <alegr@xxxxxxxxxxxxx>
 - *Date:* Wed, 4 Apr 2007 09:16:42 -0700
-

Remember that OVERLAPPED structure should remain valid (in the current stack frame or allocated from heap) during all time while the I/O is in progress.

"Rainny" <rainny@xxxxxxxx> wrote in message
news:1175700364.089631.182350@xx

Hi everybody,

I wonder if it is legal to start another asynchronous socket I/O in IOCP worker thread. The actual code is too verbose to post here, and what follows is a sketch:

```
DWORD WINAPI worker_proc(LPVOID param)
{
    LPOVERLAPPED ol;
    ...

    BOOL res = ::GetQueuedCompletionStatus(cport, &n, &ph, &ol,
    INFINITE);

    if (res == TRUE && ph) {
        // foo() allocate a new OVERLAPPED structure,
        // using it to start another async i/o operation on the same
        // socket,
        // or simply call ::PostQueuedCompletionStatus(), and then,
        // SOMETIMES,
        // stack below worker_proc() is corrupted.
        foo();
    }

    return 0;
}
```

By 'corrupted' I mean that the first 4 bytes of foo() stack is overwritten with '0x00000010', or in the absence of local variables, the return address (or something) becomes '0x00000010'.

Re: Starting another asynchronous I/O in IOCP worker thread causes stack corruption

Does anybody know what's going wrong here? It confuses us for a very long time, and any hint is highly appreciated.