Why does my asynchronous I/O complete synchronously?

devblogs.microsoft.com/oldnewthing/20110923-00

September 23, 2011

Raymond Chen

A customer was creating a large file and found that, even though the file was opened with `FILE_FLAG_OVERLAPPED` and the `WriteFile` call was being made with an `OVERLAPPED` structure, the I/O was nevertheless completing synchronously. Knowledge Base article 156932 covers some cases in which asynchronous I/O will be converted to synchronous I/O. And in this case, it was scenario number three in that document. The reason the customer's asynchronous writes were completing synchronously is that all of the writes were to the end of the file. It so happens that in the current implementation of NTFS, writes which extend the length of the file always complete synchronously. (More specifically, writes which extend the *valid data length* are forced synchronous.)

We saw last time that merely calling `SetEndOfFile` to pre-extend the file to the final size doesn't help, because that updates the file size but not the valid data length. To avoid synchronous behavior, you need to make sure your writes do not extend the valid data length. The suggestions provided in yesterday's article apply here as well.

Raymond Chen