

Why are my file write operations synchronous, even though I opened the file as `FILE_FLAG_OVERLAPPED`?

devblogs.microsoft.com/oldnewthing/20180725-00

July 25, 2018



Raymond Chen

A customer found that even though they opened a file with the `FILE_FLAG_OVERLAPPED`, their calls to `WriteFile` were nevertheless synchronous. Why are their overlapped writes completing synchronously?

As currently implemented, there are at least three things which will cause a write to be forced synchronous:

1. Writing to an NTFS-compressed file.
2. Writing to an NTFS-encrypted file.
3. Extending a file.

The first two are easy to understand. The third is trickier.

You might think you can avoid extending the file with a write by seeking to the desired final file size, performing a `SetEndOfFile`, seeking back to position 0, and then performing the desired write operations.

It turns out this doesn't work because setting the end-of-file extends the logical file size, but the physical file size is unchanged. There are two obvious ways to change the physical file size:

First, you can use the `SetFileValidData` function. This allocates space for the file on disk but does not zero-initialize those bytes. You just get whatever garbage was lying in those sectors at the time. This is naturally a violation of security since it allows a user to access data that belonged to another user, so this operation requires `SeManageVolumePrivilege` which is normally assigned only to administrators.

Less scary is seeking to the desired end of file position minus one, then writing one byte. This write will be synchronous, and it may take a while since the system needs to zero-initialize all the bytes in between, but it does finish with the desired state, and it doesn't require administrator privileges.

Once you've caused the file to be extended to the desired size, you can seek back to position 0 and perform your overlapped write operations, which will now complete asynchronously, assuming you haven't encountered any other obstacles.

Raymond Chen

Follow

