Re: Memory mapped file pages getting cleared when memory overcommi

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2007−01/msg00367.ht

- *From*: "Alexander Grigoriev" <alegr@xxxxxxxxxxxxx>
- *Date*: Sun, 28 Jan 2007 18:03:03 −0800

File mapping object keeps a reference to the FILE_OBJECT, not a handle. The filesystem gets IRP_MJ_CLEANUP when the last handle to the file is closed. This is the last chance to delete the file, even though the file mapping objects still keep a reference to FILE_OBJECT. When the last reference to FILE_OBJECT is released (file mapping is deleted), the filesystem gets IRP_MJ_CLOSE, but it cannot delete files at this time because of IRQL limitations.

I wonder if MS guys can confirm this behavior.

"Michael Vogt" <MichaelVogt@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:E8E4F916−79C2−432B−B0E9−5902CA88D6B5@xxxxxxxxxxxxxxx

> I don't think the file should even be really closed. Even though the program dose a close of the file handle and the mapping object, it never did
> the UnmapViewOfFile. Should not the system keep the file open because of that.
>
> "Alexander Grigoriev" wrote:
>
> > I guess, that the underlying backing file becomes deleted and inaccessible
> > for page−out. Thus, any pageout discards the page. I don't like this behavior, but it could be by design. If it is, it should better be documented.
> >
> > "Michael Vogt" <MichaelVogt@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
> > news:E14D5FFD−DF41−40D9−A694−05D1CB9010D0@xxxxxxxxxxxxxxx
> >
> > > I have a win32 program running on XP Pro that is using a memory mapped

file
to create shared memory. I find that some or all pages of that
memory
are
being cleared in memory overcommit situations.

The program does not actually need the persistence of a file,
so the
CreateFile() is called with flag
FILE_FLAG_DELETE_ON_CLOSE. After
doing
the
CreateFileMapping()and MapViewOfFile() the program does
a CloseHandle()
on
the file handle and on the handle to the file mapping object
but still
keeps
"open" the pointer returned from MapViewOfFile().

Here is the relevant initialization code for the memory:

```
hFilePtr = CreateFile( TCPDRV32FILE,
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
CREATE_ALWAYS,
FILE_ATTRIBUTE_HIDDEN |
FILE_ATTRIBUTE_TEMPORARY |
FILE_FLAG_DELETE_ON_CLOSE |
FILE_ATTRIBUTE_NORMAL,
NULL );

hFileMap = CreateFileMapping( (HANDLE)hFilePtr,
NULL,
PAGE_READWRITE,
0,
size,
TCPDRV32SHR );

lpMapAddr = MapViewOfFile( hFileMap,
FILE_MAP_ALL_ACCESS,
0, 0,
0 );

CloseHandle( hFileMap );
CloseHandle( hFilePtr );
```

In the relevant scenarios, no other application process opens
the

mapped
memory.

The memory is only getting lost/cleared when I do
something like open
20
IE
sessions.

The memory clearing can be prevented by any of the
following:
− Don't set FILE_FLAG_DELETE_ON_CLOSE
− don't close the file handle
− frequently read from the memory.

I need to know if this behaviour is expected because I need
to rule out
that
the program is doing something else wrong.

The reason I suspect something else is going on is because in
some
application scenarios, the problem does not occur. That is,
there are
certain
application scenarios that experience the problem and others
that do
not.
In
at least some of those cases the memory is not being read any
more
frequently
than a failing case.

See this post in the WinDbg newsgroup for some other
background:
http://msdn.microsoft.com/newsgroups/default.aspx?dg=microsoft.public.windbg&tid=89af28

Thanks for any help or insight.

Michael Vogt

.