

Re: Kernel mode to user mode

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2008-02/msg00281.html>

- *From:* "Don Burn" <burn@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 13 Feb 2008 10:13:58 -0500
-

Walter,

I hate to disagree, but after having bullet proofed a couple of event base solutions to handle the various termination situations and event overrun problems, I would never use an event as a signal to an applicaiton unless I never cared if the application got the event, and even then recognize since all handles are not cleaned up the applicaiton may be a zombie for a long time.

Worse yet I have not found there to be any speed advantage to an event unless there is no data passed at which point one wonders what the effort is for. Finally, the OP said he is new to driver development, inverted calls is easy, supporting all the mess of detecting terminated processes, can be a lot of complexity.

If this is a simple PNP driver, the OP should use KMDF and let it handle the PnP/Power stuff. If it is a legacy driver, the PnP considerations do not apply. Finally, if this is a learning effort, the OP should learn how to communicate with the standard model which is IRP's, and not go down the deadend approach of events, especially since they will likely be slower.

--

Don Burn (MVP, Windows DDK)
Windows 2k/XP/2k3 Filesystem and Driver Consulting
Website: <http://www.windrvr.com>
Blog: <http://msmvps.com/blogs/WinDrvr>
Remove StopSpam to reply

"Walter Oney" <waltoney@xxxxxxxxxxxxxx> wrote in message
news:47B30668.5F8CE454@xxxxxxxxxxxxxxxxxx

Don Burn wrote:

Don't waste your time with an event. Use an inverted call

Re: Kernel mode to user mode

<http://www.osronline.com/article.cfm?id=94>. Basically, you have you application pass one or more IOCTL's to the driver which pends them till it wants to notify the app, then it competes them. Using an event has a ton of problems.

I would respectfully disagree. You have to provide a way to cancel an IOCTL when various things happen (including PNP events), and synchronization is a nightmare. Managing an event object is child's play in comparison, provided that you do the right things with the KEVENT instead of just using the user-mode handle willy-nilly.

The OP's problem may be that he's doing a non-alertable kernel-mode wait for the user-mode application to respond. It would be far better to have the application send an IOCTL (which doesn't get pended). Next best would be to do an alertable user-mode wait.

--

Walter Oney, Consulting and Training
<http://www.oneysoft.com>