

Re: When KeEnterCriticalRegion ?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2004-09/0214.html>

From: dave (dave_at_discussions.microsoft.com)

Date: 09/04/04

Date: Fri, 3 Sep 2004 17:15:01 -0700

I need to implement this flow:

Appl_1 < r/w > Drv_1 < r/w > Drv_2 < r/w > Appl_2

Where:

Drv_1 & Drv_2 communicate through chared memory.

And I'm looking for the right way to synchronize access to the shared data. Right now I'm using spin lock but I'm not sure if it's the only way.

Probably I will read ch.4/5 one more time.

Thank you

"Walter Oney" wrote:

> *dave wrote:*

> >

> > *"Highest-level drivers can call this routine*

> > *while running in the context of the tread*

> > *that requested the current I/O operation..."*

>

> *This is what happens when tech writers who have too little understanding*

> *interview developers who have too little time. AFAIK,*

> *KeEnterCriticalRegion does exactly one thing: it prevents delivery of*

> *certain APC's to a thread while leaving the thread at PASSIVE_LEVEL. The*

> *reason for using it is so that you can enter a critical section (by*

> *claiming a mutex or a SynchronizationEvent, for example) and know that*

> *your code won't be reentered due to an APC getting queued for the same*

> *thread. File system drivers care about this, but most other drivers*

> *don't. That's because most other drivers either can't be reentered on an*

> *APC path or synchronize at DISPATCH_LEVEL using a spin lock.*

microsoft.public.development.device.drivers: Re: When KeEnterCriticalRegion ?

- >
- > *So what has led you down this particular rathole?*
- >
- > --
- > *Walter Oney, Consulting and Training*
- > *Basic and Advanced Driver Programming Seminars*
- > *Check out our schedule at <http://www.oneysoft.com>*
- >