

Re: question about thread scheduling

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2007-03/msg01018.html>

- *From:* "Zhiqiang Li" <zhiqiang.li@xxxxxxxxxxx>
 - *Date:* Tue, 27 Mar 2007 16:05:29 +0200
-

Hi Chris,
thanks for your help!

Zhiqiang

"<tacke/>" <tacke[@]opennetcf[dot]com> schrieb im Newsbeitrag
<news:%23DKFEqGcHHA.4888@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

```
CeSetThreadPriority(GetCurrentThread(), priority);
```

--

Chris Tacke – Embedded MVP
OpenNETCF Consulting
Managed Code in the Embedded World
www.opennetcf.com

--

"Zhiqiang Li" <zhiqiang.li@xxxxxxxxxxx> wrote in message
<news:%23fkpciGcHHA.3408@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Dear All,
there is any way to change the default priority of the main thread of an
application process?

thanks in advance

Zhiqiang

"Zhiqiang Li" <zhiqiang.li@xxxxxxxxxxx> schrieb im Newsbeitrag
<news:OBk93KvaHHA.4552@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Hi Remi,

I realize that I have misunderstood what you have told me.
Actually,
what you provided is the template for my implementation.
thanks a lot for your help!

Zhiqiang

"Remi de Gravelaine" <gravelaine at aton dash sys dot fr>

Re: question about thread scheduling

schrieb im
Newsbeitrag
news:OHk45G9ZHHA.4520@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Zhiqiang,

You should try to tell us **exactly** what you want to do.

You should also avoid multiplying the discussion threads if you want people to take care of you from the beginning.

If I understood what you want to do, you have implemented a polling loop to do some job (what kind of job?) every 4ms.

You have put this loop in a high-priority thread started by the XXX_Init function of a stream driver. You are using CE 6.0 on a CEPC.

The loop should thus look like:

```
while (!g_bSuicide)
{
Sleep(3);
<do some job>
}
```

There is no reason that such a loop exits except when g_bSuicide is set to TRUE (something that can be done in XXX_Deinit) or the <do some job> code executes a break or goto instruction.

Now, why Sleep(3) instead of Sleep(4)? Because Sleep(3) will sleep for **at least** 3 ms and because Sleep is synchronized on timer ticks, that fire every ms (producing the SYSINTR_RESCHEd you are focusing on.) So, Sleep(3) returns from sleeping right after the third tick and a new 1ms slice begins. You <do some job> during a portion of this slice and reenter Sleep(3). Sleep(3) puts your thread to sleep for 3 timer ticks and that means the time remaining in the current tick + 3 ms.

Re: question about thread scheduling

HTH

Remi