Re: how to wait for socket communications

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2007−05/msg00145

- *From*: "Alexander Nickolov" <agnickolov@xxxxxxxx>
- *Date*: Tue, 15 May 2007 14:20:54 −0700

I think OP is worried about performance and housekeeping
(not leaving temporary files around). However, with memory−
mapped file you don't need an actual file to back the mapping −
there's a standard approach of using the swap file for backing
the data. Then even file I/O may not occur and all communication
be in memory. Memory−mapped files should really be considered
in this context as memory allocation technique, not file I/O.

−−
==========
Alexander Nickolov
Microsoft MVP [VC], MCSD
email: agnickolov@xxxxxxxx
MVP VC FAQ: http://vcfaq.mvps.org
==========

"Ali" <abdulrazaq@xxxxxxxxx> wrote in message
news:1179210849.716145.295730@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

> On May 14, 10:55 pm, Ananya <Ana...@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote:
>
> > I am thinking of using sockets, because this was suggested in the last
> > post
> > to my question "Connecting Java to C++" in the Java Programming Forum at
> > forum.java.sun.com, the exact link
> > being:http://forum.java.sun.com/thread.jspa?threadID=5133663&tstart=0.
> >
> > I am not sure why Alexander thought that sockets are not the right tool.
> > I
> > think that the advantage using sockets might be that it is faster than
> > writing a file, since you can communicate a whole array.
> >
> > Thanks for all your communications!
> >
> > "Ali" wrote:
> >
> > > On May 14, 9:02 am, Ananya
> > > <Ana...@xxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

I am trying to do socket communications between my C++ and my Java program.

I created a Java program for drawing curves and I connected it to Adobe's C++ program for making a Photoshop plugin.

I am calling ShellExecuteEx for calling my Java program from C++, and it brings up a separate window.

First I would like to communicate the Photoshop picture information into my Java window. Originally I tried to to that with the lpParameters string in ShellExecuteInfo, but it is too slow to build this string.

So now I am trying to communicate this with socket communication. I first create a server in C++ before calling ShellExecuteEx, then I create a client in Java in the main method, but how can I wait in C++ for this Java client to be created before calling accept and sending the Photoshop picture information?

I know I could just wait for a few seconds to make sure that my Java program came up by calling:

```
if ( ei.hProcess != NULL )
{
WaitForSingleObject(ei.hProcess, 10*1000
);}
```

But I would rather not lose any time and just wait exactly until the
Java
client is created.

And how can I wait in Java before receiving the picture information?

Finally, how can I wait in C++ for the Java program to close, at
which point
the information of curves that were drawn in Java is sent to C++ (to
be
received for the workpath in Photoshop)?

Thanks for your time looking at this.

Hmm, i though someone else will jump in to make this point clear. As
pointed by Alexander that using socket is not a good design approach.
His suggestion for mapped file is indeed flexible and commercial grade
solution BUT you can do that with simple file operation as i told in
previous thread. Why simple file I/O cant be your solution? and what
makes you think that sockets are good to use?

ali

Snip:

> I am not sure why Alexander thought that sockets are not the right tool.
> I
> think that the advantage using sockets might be that it is faster than
> writing a file, since you can communicate a whole array.

You must be kidding;−) BTW sockets are also no more than files with over head of clumsy protocol (tcp/udp anyother) on the top . Yeah, sockets can work and if you familiar with that then go head and do it. I bet socket design can't be never faster then file I/O given the facts that you will doing all the stuff on single machine. If you are not familiar with files thats another story but usually writing or reading to file wont take more than 3 to 4 lines. Easy peasy Japanese ;−)

ali

.