Subject: Re: IDL Spawn command on UNIX system
Posted by John D. Sample on Sun, 22 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <Pine.GHP.4.05.9908212248380.14331-100000@nagik.cs.colorado.edu>,
Kenneth Mankoff <mankoff@nagik.cs.colorado.edu> wrote:

>
> On Sat, 21 Aug 1999, Mike Schienle wrote:
>
>> In article <37bf3790@mindmeld.idcomm.com>, "John D. Sample"
>> <sample@idcomm.com> wrote:
>>
>>> I'm using IDL to SPAWN a c program on many UNIX systems using the 'remsh'
>>> (remote shell) command.
>>>
>>> I would like to SPAWN these jobs in the background and forget them until
>>> later, but, by default my IDL session stops and waits for the SPAWNed job to
>>> be complete.
>> ...
>>
>> I hope this is something you've overlooked and I'm not sounding like an
>> idiot for suggesting it, but how about adding an ampersand (&) to the end
>> of the command string which is sent to the spawn command? That would
>> background the spawned command and return to the IDL command/program
>> without waiting for the process to complete.
>>
>> --
>> Mike Schienle   Interactive Visuals, Inc.
>> mgs@ivsoftware.com   Remote Sensing and Image Processing
>> http://www.ivsoftware.com/   Analysis and Application Development
>>
>
> but it wouldn't solve the problem of your process quota. Talk to your
> sysadmin, and ask them to raise the number of processes you are allowed to
> run at any one given time to solve that problem.
>
> -ken.
>

Thanks for both replies. I am using the ampersand, and initially assumed
this *would* send the process to the background. Apparently this is not
enough to return control to IDL. Even though the process runs in the
background, IDL still stops and waits for the process to complete before
processing the next widget event. This is consistent with the documentation
(which I consulted later).

Using the pipe DOES return control to IDL.

I will check on the number of processes allowed. There must be some kind of work around however, because I *am* able to launch 50 jobs at a time using UNIX scripts...and maybe the answer is to learn this language well enough to decode what is going on.

A problem still remains with the pipe used to relegate the job to the background. If it fills up, it will bring the job to a stop (according to the IDL documentation.) I may need to create some code which reads from the pipe every so often to insure it doesn't fill up, and some how detect when the job is complete so I can FREE_LUN the pipe.

Chip