Subject: Re: Parallel Processing
Posted by lecacheux.alain on Thu, 05 Jul 2012 15:53:26 GMT
View Forum Message <> Reply to Message

On 5 juil, 17:16, Stefan <stefan.meing...@gmail.com> wrote:
> Am Donnerstag, 5. Juli 2012 17:13:02 UTC+2 schrieb fawltyl...@gmail.com:
>
>
>
>
>
>> On Thursday, July 5, 2012 4:52:23 PM UTC+2, Stefan wrote:
>>> Hi
>
>>> Now that I have successfully implemented multi-threading another problem occured:
>
>>> To invoke multiple processes I start in a loop with
>
>>> bridges[i]->EXECUTE, 'program,par1,par2', /NOWAIT
>
>>> where 'bridges' is an object-array which holds the different child processes.
>>> Upon the execution of the last process I do
>
>>> bridges[i]->EXECUTE, 'program,par3,par4'
>
>>> And after that I destroy my bridges in a loop.
>
>>> Now I have a problem if the last process finishes before one of the previous processes since upon its completion it will directly move to the part where all bridges are destroyed and kills my program...
>
>>> Is there an easy way to tell IDL to wait for all my processes to finish and then destroy the bridges?
>
>>> thanks
>>> Stefan
>
>> Use IDL_IDLBridge::Status to determine whether the job is finished. Pseudocode:
>

```
>> ndone=0
>> running=replicate(1, njobs)
>> while ndone lt njobs begin
>>    for j=0,njobs-1 do begin
>>       if running[j] then begin
>>         query status of the j-th job with IDL_IDLBridge::Status
>>         if finished
>>           destroy bridge
>>           running[j]=0
>>                ndone++
>>             endif
>>          endif
>>      endfor
>>      wait, 1
>>  endwhile
>
>>  regards,
>>  Lajos
>
```

> Ah, I see you put WAIT there if its not finished. I thought of using this in a loop but without putting WAIT you spend too many resources.
>
> Thanks! :)
>
>

My way to manage that is putting /NOWAIT, but waiting inside the loop,
as follows:

```
  repeat begin
    wait, 1
    test = 1B
    foreach b,bridges do test AND= (b.Status() ne 1)
  endrep until test
```

alain.