Subject: Re: Subject : locks, semaphores, and such
Posted by Paul van Delst on Mon, 28 Jan 2002 15:25:37 GMT
View Forum Message <> Reply to Message

Ken Mankoff wrote:
>
> Hi,
>
> I need to create a unique directory to do some work in. This is a
> web-based IDL script (through ION), and multiple people may access the
> site at the same second. Each user gets their images generated in a
> uniquely named directory, which is then deleted 5 minutes later...
>
> I know a bunch about semaphores and threads and stuff like that, but
> here is the problem. The "threads" (users) do not have inter-thread
> communication, unless its through files on disk. If they did, this
> would be a perfect use for COMMON blocks. Also, the web server has 2
> CPUs so much of the traditional semaphore logic is invalid, as both
> threads can acquire a lock on the same clock cycle.
>
> ---
> Here is the code I currently use to lock a file:
>
> uniq = strtrim(long(systime(1)), 2)
> repeat begin
> uniq = uniq + 1L
> f = findfile(uniq,count=cnt) ; [1] ; find unclaimed file
> endrep until cnt eq 0 ; [2]
> spawn, 'touch ' + file ; [3] ; claim file
> spawn, 'rm ' + file + ' | at now + 5 minutes' ; free file later
>
> Of course, there are multiple clock cycles and disk accesses between
> when [1] loads a "0" into cnt, and when [3] finishes executing the
> 'touch' part of the command.
>
> The system this runs on is 5 people on 5 computers using 5 different
> IDL sessions all sharing 1 cross-mounted disk. So far, the bug has
> never evolved into an error :)
> ---
>
> At worst, can anyone think of a way to detect if multiple users got
> assigned the same directory and at least fail gracefully? At best,
> does anyone have an algorithm pre-built for just this purpose?

I had a similar problem a while back related to running batch jobs via unix shell scripts, but I think my problem was much simpler than yours and my solution is probably also too simple an approach (don't know nuthin' 'bout semaphores and such...) But anyway....

I create a time_date string and prefix that with the username (or some other user specific ID). This prevents users from running into each other. To prevent the user from running into him/herself (if a single user is running multiple jobs) I suffix the user_time_date string with increasing integers (up to some predefined max value.), e.g. if directory "user_time_date" exists, I check for "user_time_date_2". If that exists I increment the suffixed integer and loop. etc. etc. So far it's worked fine - I've initiated 10's of 1000's of jobs (some simultaneously but doing the job submission through multiple windows; other not necessarily simultaneously since it takes some time to process the inputs, but over about a minute or so) and haven't yet had a collision. (Fingers crossed)

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul