Subject: Re: printf within time interval (also posted on exelisvis.com)
Posted by justinclouds on Sun, 25 Nov 2012 21:17:55 GMT
View Forum Message <> Reply to Message

On Saturday, November 24, 2012 6:28:50 PM UTC-7, Craig Markwardt wrote:
> On Saturday, November 24, 2012 6:57:48 PM UTC-5, justinclouds wrote:
>
>> I am fairly new to IDL and need some help with coding a specific task. I have a data file with time intervals that I read in from one text file (file A with i elements). For example, the time interval vectors are 'timestart' and 'timeend' and they each have the form FLOAT = Array[1, 78]. The times are in utc seconds. These two vectors form the time interval (i.e. timestart[i] and timeend[i] where timestart[i] < timeend[i]) . I also read in another text file (file B with j elements). File B has a time vector 'time' of the form FLOAT = Array[18295] and also in utc seconds. I need to printf a binary value of 1 to a new text file when 'time' is between 'timestart' and 'timeend' and a binary value of 0 when 'time' is outside the time period identified by 'timestart' and 'timeend' for each value of time[j]. The objective is to identify when 'time' occurs between 'timestart' and 'timeend' by a binary 1 and 0 when it does not.
>
>>
>
>>
>
>>
>
>> Any help with example code related to this task would be greatly appreciated!
>
>
>
> The subject of your message makes it sound like your issue is with printing, but it's not. If I were you, I would separate the reading, processing and writing of your data. Reading and writing I assume you can get under control by yourself.
>
>
>
> For processing, it sounds like you have a set of start and stop intervals. You might consider my "GTI" library which is designed for exactly these kinds of manipulations of "Good Time Intervals". The library can be found here:
>
>   http://cow.physics.wisc.edu/~craigm/idl/arrays.html#GTI
>
> Look at the GTIWHERE function in particular.
>
>
>
> You start with a 2xN array which gives the start & stop of each interval, and a list of times; and then the routine returns an index list (like WHERE) to which elements of your original list are within the time intervals.
>

>
>
> Example:
>
>   gti = transpose([[timestart],[timeend]])  ;; Reform into a 2xN array
>
>
>
>   wh = gtiwhere(time, gti, count=ct) ;; Find which times are within bounds
>
>
>
>   mask = bytarr(n_elements(time)) ;; Make a 1/0 array (initialized to 0)
>
>
>
>   if ct GT 0 then mask(wh) = 1    ;; Fill the good array values with 1
>
>
>
> Best wishes,
>
> Craig

Hi Craig,

Your example and your gtiwhere function worked very well. The only thing I changed was gti = [timestart,timeend]. Thanks a bunch!

Duncan