tam wrote:
>
> JD Smith wrote:
> ... In any case,
>> it is oh so trivial to convert date to systime, the inverse:
>>
>> st=(julday(month,day,year,hr,min,sec)-2440587.5D)*86400.0D
>>
>> That's it! For instance:
>>
>> IDL> st0=systime(0) & st1=systime(1)
>> IDL> d=bin_date(st0)
>> IDL> st=(julday(d[1],d[2],d[0],d[3],d[4],d[5])-2440587.5D)*86400. 0D
>> IDL> print,FORMAT='(A,":",2D30.5)',st0,st1,st
>>
> ...
>>
>> JD
>
> How are leap seconds accounted for here? They usually cause
> niggling problems if one actually needs to worry about
> precision at the second level over periods of more than a
> month or two. It looks like your code is ignoring them.
> The biggest pain is that they are not predictable so the
> code needs to access an external file, or update an
> internal table periodically.

As you can see, all of the details of time handling are left to internal
IDL routines. Inasmuch as IDL may or may not deal with leap seconds, so
too would this algorithm. That said, I'm quite certain they don't.

This is of course the reason to use Julian Date, which explicitly don't
include leap seconds, and thus is quite continuous, independent of the
whimsy of datekeepers. Currently there are about 1 minute 5 seconds
between UTC (the physical time clocks keep, including leap seconds), and
the Julian date. There have been rumours of cancelling the leap second
program for a variety of reasons.

Good luck,

JD