Subject: Re: comparing and concatenating arrays...please help!!
Posted by Pepijn Kenter on Thu, 08 Jan 2004 13:25:13 GMT
View Forum Message <> Reply to Message

Martin Doyle wrote:
> Hello all,
>
> I really hope someone out there can help me with this....I am tearing
> my hair out as my code is so slow!
>
> I have 2 files of data (hourly met data) with one file containing one
> set of parameters, and the other file containing another set of
> parameters. What I am trying to do, is to match the data based on the
> YY, MM, DD and HH values and then write BOTH sets of parameters to a
> seperate file. For example;
>
> file1:
> 1954 12 31 23 90 11 4 366 0.00
>
> file2:
> 1954 12 31 23 2.80 2.10 2.20 95.21
>
> intended result:
> 1954 12 31 23 90 11 4 366 0.00 2.80 2.10 2.20
> 95.21
>
> NOTE: Both files have no order to them, so a simple concatenation
> won't work
>
> I have written some code, but it is wrist slashing-ly slow!;
>
> I read in each variable as a seperate array...
>
> b=0L
> REPEAT BEGIN
> c=0L
> REPEAT BEGIN
> If (year(b) EQ year2(c)) AND (month(b) EQ month2(c)) AND (day(b) EQ
> day2(c)) AND (hour(b) EQ hour2(c)) THEN BEGIN
>
> printf, 3, year(b), month(b), day(b), hour(b), winddir(b), windsp(b),$
> present(b),visib(b), mslpres(b), airt(c), dewt(c), wett(c), relh(c),$
> format = finalformat
> endif
>
> c=c+1
>
> ENDREP UNTIL c EQ lines2-1

>
> b=b+1
>
> ENDREP UNTIL b EQ lines1-1
>
> I'm sure there must be a better way than this.
>
> Please help me!
>
> Many thanks in advance, Martin..

Hi.

You'll need a more efficient algorithm. For each line in file1 you walk through all the data of file2. This costs in the order of lines1*lines2 operations (btw, how big are these files?). This means that if these files double in size, your program will run 4 times as long!

I'm sure that your program can be speeded up with some smart use of the WHERE command, but since the WHERE command also traverses through a complete array, nothing is changed in principle.

To do better than that you first have to sort the data. You can use the SORT procedure of IDL. I don't know what algorithm IDL uses, but in general sorting a dataset with n elements can be done in the order of n*log(n) operations (instead of n^2, what you use now). Furthermore, a lot of effort is put in this routine to make it as efficient as posible; let IDL do the hard work. You could also use an external program to sort your files, like the sort command under linux.

When you have sorted the data, you'll need to write an algorithm that traverses both arrays simultaniously. For example, walk through dataset1 and for each line in set1 search the line in the set2 with the same date starting at the previous found line in set2. Because your files are sorted, you only need to walk trough a small part of file2 for each line in file1. I'm sure you can think of something.

HTH, Pepijn Kenter.

PS. please indent your code, this makes it more readable.