Subject: Inconsistency: "sort" on UNIX <-> Windows
Posted by Daniel Luebbert on Fri, 07 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi,

I've been using the "sort" routine to sort an 2D-array by increasing values of its first column.
The "sort" routine seems to behave differently in the UNIX and the Windows versions of IDL5.3.

Here is a little test routine I wrote:

```
;**************************
pro test_sort
data = [[1,3], $
 [2,5], $
 [1,4], $
 [4,7], $
 [1,2] $
 ]

print,"original: "
print,data
print,"sorted: "
print,data[*,sort(data[0,*])]
end
;**************************
```

The result on UNIX (Linux) is what I would expect:

```
;**************************
original:
      1       3
      2       5
      1       4
      4       7
      1       2
sorted:
      1       3
```

```
      1      4

      1      2

      2      5

      4      7

;*************************
```

BUT: The result on Windows is different:

```
;*************************
original:
      1      3
      2      5
      1      4
      4      7
      1      2
sorted:
      1      2
      1      4
      1      3
      2      5
      4      7
;*************************
```

In other words: The "sort" routine on Windows doesn't preserve the original order of entries which have the same value (see the 3 "1"-values in my test data), but apparently rearranges them in a random way.

What I'd like to know is:
How can it be that such a basic routine behaves differently in differnet implementations of IDL?

And, more importantly: can anybody tell me how to make the Windows-version of "sort" behave correctly,
without writing my own routine for sorting?

THanks for any hint,

Daniel

--
--- ------------
Dr. Daniel Lï¿½bbert
Stanford Synchrotron Radiation Laboratory (SSRL)
Bldg. 137, Mailstop 69, 2575 Sandhill Road, Menlo Park, CA 94025, USA
Phone: (650) 926-4010   FAX: (650) 926-4100
luebbert@slac.stanford.edu

Subject: Re: Inconsistency: "sort" on UNIX <-> Windows
Posted by Martin Schultz on Tue, 18 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Daniel Luebbert (luebbert@stanford.edu) writes:

>> In other words: The "sort" routine on Windows doesn't preserve the
>> original order of entries which have the same value (see the 3
>> "1"-values in my test data), but apparently rearranges them in a random
>> way.
>>

>
> P.S. There is not much that can be done, except sort
> the other dimension by some consistent criteria (e.g.
> the "order" in the original array). Ugly, but perhaps
> necessary depending upon your application.
>
> David

... and for this you could use my multisort.pro which you find at
http://www.mpimet.mpg.de/~schultz.martin/idl/

Martin

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

Subject: Re: Inconsistency: "sort" on UNIX <-> Windows
Posted by R.Bauer on Fri, 04 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

here is another sort method from our library too.

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/n_sort.tar.gz

```
; NAME:
;   n_sort
;
; PURPOSE:
;   sort an array according to n columns
;
; CATEGORY:
;   MATH
;
; CALLING SEQUENCE:
;   result=n_sort(matrix)
;
; INPUTS:
;   matrix : the input matrix, which shoul be sorted
;
; KEYWORD PARAMETERS:
;   INDEX:  a vector of column numbers (default: indgen(number of
columns))
;           it defines the order of the columns, how the matrix will be
sorted
;
;   DECREASING_ORDER: the matrix will be sorted decreasingly
;
; OUTPUTS:
;   j_index: the vector of sorted line numbers
;
; EXAMPLE:
;     matrix = [[ 2, 3, 4 ], $
;               [ 1, 4, 6 ], $
;               [ 1, 4, 5 ]]
;     j_index=n_sort(matrix)
;     j_index=[2,1,0]
;
;     matrix = [[ 1, 4, 5 ], $
;               [ 1, 4, 6 ], $
;               [ 2, 3, 4 ]]
;     j_index=n_sort(matrix,index=[1,0,2])
;     j_index=[2,0,1]
```

Reimar Bauer