Subject: Re: Help: Byte-swapping program
Posted by grunes on Mon, 12 Dec 1994 15:12:57 GMT
View Forum Message <> Reply to Message

In article <schaep-1212941334430001@130.60.16.90> schaep@rsl.geogr.unizh.ch (Michael E. Schaepman) writes:
> Since we are moving our data from a VAX and a DecStation to Suns and Macs,
> I am looking for a program, that converts big <> low-endian byte order and
> IEEE <> VAX representation.

In IDL or PV-Wave, the bytes of each element of array A can be swapped
by
  BYTEORDER,A

I think IEEE is more dificult.

I once wrote a FORTRAN routine, but it has not been fully tested,
and is not portable to those fortran compilers that don't let you
equivalence numbers and characters:

```
-----------------------------CUT HERE-------------------------------
c-----------------------------------------------------
      function FromVaxR4(x)
c Function to convert Vax real*4 number to local floating point.
c Cannot handle NANs or numbers which are too small or too large.
c By mitchell r grunes.
      integer*4   x,y,i             ! Really Vax real*4--but
                                    !  must be kept in integers
                                    !  so won't be "normalized".
      character*1 a(4)
      equivalence (y,a)
      parameter (Mask23=2**23-1)
      parameter (ioffset=128+24)

      y=x
      i=            iand(ichar(a(2)),255)
      i=ior(ishft(i,8),iand(ichar(a(1)),255))
      i=ior(ishft(i,8),iand(ichar(a(4)),255))
      i=ior(ishft(i,8),iand(ichar(a(3)),255))

      iexponent=iand(ishft(i,-23),255)-ioffset
      mantissa=iand(i,Mask23)

      if(i.eq.0)then
        FromVaxR4=0
      else
        mantissa=ior(ishft(1,23),mantissa)
        if(i.gt.0)then
```

```
          FromVaxR4= mantissa*2.**iexponent
        else
          FromVaxR4=-mantissa*2.**iexponent
        endif
      endif
      end
c-----------------------------------------------------
      function FromIEEER4(x)
c Function to convert IEEE real*4 number to local floating point.
c Assumes number written on a "most significant byte first" machine like
c  a Sun or SGI workstation.
c Cannot handle NANs or numbers which are too small or too large.
c By mitchell r grunes.
      integer*4   x,y,i             ! Really IEEE real*4--but
                                   !  must be kept in integers
                                   !  so won't be "normalized".
      character*1 a(4)
      equivalence (y,a)
      parameter (Mask23=2**23-1)
      parameter (ioffset=128+22)

      y=x
      i=              iand(ichar(a(1)),255)
      i=ior(ishft(i,8),iand(ichar(a(2)),255))
      i=ior(ishft(i,8),iand(ichar(a(3)),255))
      i=ior(ishft(i,8),iand(ichar(a(4)),255))

      iexponent=iand(ishft(i,-23),255)-ioffset
      mantissa=iand(i,Mask23)

      if(i.eq.0)then
        FromIEEER4=0
      else
        mantissa=ior(ishft(1,23),mantissa)
        if(i.gt.0)then
          FromIEEER4= mantissa*2.**iexponent
        else
          FromIEEER4=-mantissa*2.**iexponent
        endif
      endif
      end
c-----------------------------------------------------
      function FromRIEEER4(x)
c Function to convert IEEE real*4 number to local floating point.
c Assumes number written on a "least significant byte first" machine like
c  a PC.
c Cannot handle NANs or numbers which are too small or too large.
c By mitchell r grunes.
```

```
integer*4   x,y,i              ! Really IEEE real*4--but
                        !  must be kept in integers
                        !  so won't be "normalized".
character*1 a(4)
equivalence (y,a)
parameter (Mask23=2**23-1)
parameter (ioffset=128+22)

y=x
i=              iand(ichar(a(4)),255)
i=ior(ishft(i,8),iand(ichar(a(3)),255))
i=ior(ishft(i,8),iand(ichar(a(2)),255))
i=ior(ishft(i,8),iand(ichar(a(1)),255))

iexponent=iand(ishft(i,-23),255)-ioffset
mantissa=iand(i,Mask23)

if(i.eq.0)then
  FromRIEEER4=0
else
  mantissa=ior(ishft(1,23),mantissa)
  if(i.gt.0)then
    FromRIEEER4= mantissa*2.**iexponent
  else
    FromRIEEER4=-mantissa*2.**iexponent
  endif
endif
end
```