Subject: Re: extracting bits from bytes
Posted by thompson on Tue, 30 Nov 1993 02:54:17 GMT
View Forum Message <> Reply to Message

dean@phobos.cira.colostate.edu writes:

> I have a (graphic) file that is an 8-bit byte. Each of the 8 bits in each
> byte indicates the graphic position is on or off.

> Is there a way with IDL to determine which of the 8 bits is on (1) or
> off (0)? Like it can be done in FORTRAN or C.

The following routine does this.

Bill Thompson

```
 ================================================================
==================
;+
; Project     : SOHO - CDS
;
; Name        : DEC2BIN
;
; Purpose     : Convert integer decimal number to binary representation.
;
; Explanation : The binary representation of a decimal number is calculated
;               and can be displayed or returned or both or neither.
;
; Use         : IDL> dec2bin, decimal [, binary, /quiet]
;
; Inputs      : decimal - the number to convert
;
; Opt. Inputs : None
;
; Outputs     : See below
;
; Opt. Outputs: binary - the binary representation of the input.
;
; Keywords    : quiet - unless given the binary number is printed to the
;                       terminal
;
; Calls       : None
;
; Restrictions: Input must be of byte, int or long type.
;
; Side effects: None
;
; Category    : Utils, Numerical
```

```
;
; Prev. Hist. : None
;
; Written     : C D Pike, RAL, 7-Oct-93
;
; Modified    :
;
; Version     : Version 1, 7-Oct-93
;-

pro dec2bin,inp,out,quiet=quiet

;
;  convert input to LONG so that arithmetic later on will work
;
in=long(inp)

;
;  maximum possible output array
;
out=bytarr(32)

;
;  perform the conversion
;
for i=0,31 do out(31-i)=(in and 2L^i)/2L^i

;
;  trim output depending on nature of input
;
case datatype(inp) of
   'BYT': begin
            if not keyword_set(quiet) then print,'$(8I1,1X)',out(24:31)
            out = out(24:31)
          end
   'INT': begin
            if not keyword_set(quiet) then print,'$(2(8I1,1X))',out(16:31)
            out = out(16:31)
          end
   'LON': begin
            if not keyword_set(quiet) then print,'$(4(8I1,1X))',out
          end
    else: begin print,'Error: only integer types allowed.' & out = 0 & end
endcase

end
```

Subject: Re: extracting bits from bytes
Posted by chase on Tue, 30 Nov 1993 18:16:57 GMT
View Forum Message <> Reply to Message

>>>> > "dean" == dean <dean@phobos.cira.colostate.edu> writes:
In article <Nov29.232946.48256@yuma.ACNS.ColoState.EDU> dean@phobos.cira.colostate.edu writes:

dean>   I have a (graphic) file that is an 8-bit byte. Each of the 8
dean> bits in each byte indicates the graphic position is on or off.

dean>   Is there a way with IDL to determine which of the 8 bits is
dean> on (1) or off (0)? Like it can be done in FORTRAN or C.

dean>   Kelly Dean

You can do it just like in C, using a mask and the AND operator.
The IDL Boolean operators operate bitwise for any size integer
operands (byte, integer, long).

Example:

Suppose A is a byte variable containing data from your file. Then the
following will print 1 if bit 3 (numbering bits from 0 in least
significant order) is set and 0 if not set ('n'XB notation is a
hexidecimal byte constant for IDL):

print, (A and '08'XB) ne 0

or the following will print all the bits in succession:

for i=0,7 do print, ((2^i) and A) ne 0

Try it.

If you are checking a lot of data, you should store the masks in an
array:

```
mask = bytarr(8)
for i=0,7 do mask(i)=2^i

for i=0,7 do print, 'Bit',i,' = ', (mask(i) and A) ne 0
```

I hope this helps,

Chris

P.S.

Anyone using idl.el or idl-shell.el? Any comments, suggestions,
additional bug reports? Send them my way.
--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
(301)953-6000 x8529
chris_chase@jhuapl.edu

Subject: Re: extracting bits from bytes
Posted by ryba on Tue, 30 Nov 1993 22:16:02 GMT
View Forum Message <> Reply to Message

|> mask = bytarr(8)
|> for i=0,7 do mask(i)=2^i

Getting closer....try

tmp = lindgen(32) ; For Bill's routine handling up to 4 bytes
mask = 2L^tmp
out = (in and mask) ne 0

For arrays of inputs, you can use rebin:
ninbytes = n_elements(in)
tmp = rebin(lindgen(32),32,ninbytes)

etc....

--
Dr. Marty Ryba | Generation X:
MIT Lincoln Laboratory | Too young to be cynical,
ryba@ll.mit.edu | too old to be optimistic.
Of course nothing I say here is official policy, and Laboratory affiliation is
for identification purposes only, blah, blah, blah....
