Convirtiendo números del sistema decimal al sistema binario.

Pablo Santamaría

v0.1b (Septiembre 2009)

Veamos como puede convertirse la representación decimal de un número al sistema binario. El procedimiento trata a las partes entera y fraccionaria por separado, así que supongamos, en primer lugar, que el número p dado es un entero positivo. Entonces, en la base $\beta = 2$, el número puede ser escrito en la forma

$$p = a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 = \sum_{j=0}^{n} a_j 2^j,$$

donde a_j son los dígitos que queremos determinar. La igualdad anterior puede ser escrita en la forma

$$p = a_0 + 2\left(\sum_{j=1}^{n} a_j 2^{j-1}\right),$$

de donde se sigue que el dígito a_0 se identifica con el resto de la *división entera* de p por la base 2:

$$a_0 = \mathrm{mod}(p, 2).$$

Considerando ahora el cociente entero de tal división

$$c_1 = \sum_{j=1}^{n} a_j 2^{j-1} = a_1 + 2\left(\sum_{j=2}^{n} a_j 2^{j-2}\right),$$

resulta que el dígito a_1 es el resto de la división entera de c_1 por 2:

$$a_1 = \mathrm{mod}(c_1, 2).$$

Procediendo sucesivamente, si c_i denota el cociente de la división entera por 2 del paso anterior (poniendo $c_0 = p$), tenemos que

$$c_i = a_i + 2\left(\sum_{j=i+1}^{n} a_j 2^{j-i}\right), \qquad i = 0, 1, \ldots, n,$$

con lo cual

$$a_i = \mathrm{mod}(c_i, 2).$$

El proceso se detiene en n, puesto que el dividendo en el paso $(n+1)$ es cero. En efecto, siendo $c_n = a_n = a_n + 0 \cdot 2$, es $c_{n+1} = 0$. El procedimiento puede ser resumido esquemáticamente como sigue.

Para convertir un número entero positivo de base 10 a base $\beta = 2$.

Paso 1. Realice la división entera por 2 del cociente obtenido en el paso anterior, comenzando con el número dado.

Paso 2. Guarde el resto de tal división.

Paso 3. Continúe con el paso 1 hasta que el dividendo sea cero.

Paso 4. Lea los restos obtenidos, desde el último al primero, para formar la representación buscada.

Ejemplo: Convertir 29 a binario.

29		
$29/2 = 14$	resto 1	
$14/2 = 7$	resto 0	
$7/2 = 3$	resto 1	
$3/2 = 1$	resto 1	
$1/2 = 0$	resto 1	$\uparrow$

Entonces $(29)_{10} = (11101)_2$.

Consideremos ahora la parte fraccional del número en su representación decimal. Podemos entonces asumir que el número p dado es un número real positivo entre 0 y 1. Entonces, en la base $\beta = 2$, puede ser escrito en la forma

$$p = a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots = \sum_{j=1}^{n} a_{-j} 2^{-j},$$

donde a_{-j} son los dígitos que queremos determinar. Multiplicando la igualdad anterior por 2 tenemos que

$$q_1 = 2\,p = a_{-1} + \sum_{j=2}^{n} a_{-j} 2^{-j+1},$$

de donde se sigue que el dígito a_{-1} se identifica con la *parte entera* de la multiplicación de p por 2:

$$a_{-1} = [q_1]\,.$$

Consideremos ahora la *parte fraccionaria* de tal producto, si ésta no es nula podemos volver a multiplicar por 2 para obtener

$$q_2 = 2\,(q_1 - [q_1]) = a_{-2} + \sum_{j=3}^{n} a_{-j} 2^{-j+2},$$

de donde,

$$a_{-2} = [q_2]\,.$$

Procediendo sucesivamente, si q_i denota el producto de 2 por la parte fraccionaria del paso anterior (poniendo $q_1 = 2\,p$), tenemos que

$$q_i = a_{-i} + \sum_{j=i+1}^{n} a_{-j} 2^{-j+i}, \qquad i = 1, 2, dots,$$

de donde,

$$a_{-i} = [q_i]\,.$$

Notemos que el procedimiento continúa *indefinidamente*, a menos que para algún q_k su parte fraccionaria sea nula. En tal caso $q_{k+1} = 2\,(q_k - [q_k]) = 0$ y, en consecuencia, $a_{-(k+1)} = a_{-(k+2)} = \cdots = 0$, obteniéndose así una cantidad finita de dígitos fraccionarios en la base 2. Resumimos el procedimiento como sigue.

Para convertir un número decimal entre 0 y 1 a la base $\beta = 2$.

Paso 1. Realice la multiplicación por 2 de la parte fraccionaria obtenida en el paso anterior, comenzando con el número dado.

Paso 2. Guarde la parte entera de tal producto.

Paso 3. Continúe con el paso 1 hasta que la parte fraccionaria sea cero, o hasta obtener el suficiente número de dígitos requeridos para su representación.

Paso 4. Lea las partes enteras obtenidas, del principio hacia el final, para formar la representación buscada.

Ejemplo: Convertir 0.625 a binario.

0.625		
2*0.625 = 1.25	[1.25] = 1	↓
2*0.25 = 0.5	[0.5] = 0	
2*0.5 = 1	[1] = 1	

Nos detenemos puesto que $1 - [1] = 0$. Así, $(0.625)_{10} = (0.101)_2$.

Ejemplo: Convertir 0.1 a binario.

0.1		
2*0.1 = 0.2	[0.2] = 0	↓
2*0.2 = 0.4	[0.4] = 0	
2*0.4 = 0.8	[0.8] = 0	
2*0.8 = 1.6	[1.6] = 1	
2*0.6 = 1.2	[1.2] = 1	
2*0.2 = 0.4	[0.4] = 0	
2*0.4 = 0.8	[0.8] = 0	
2*0.8 = 1.6	[1.6] = 1	
2*0.6 = 1.2	[1.2] = 1	
2*0.2 = 0.4	[0.4] = 0	...

En este caso la representación es infinita, $(0.1)_2 = (0.0001100110\ldots)_2$.

El siguiente código en Fortran77 implementa el procedimiento descrito para obtener la representación binaria de cada número que se ingrese por teclado.

```
      program dec2bin
*     ----------------------------------------------------------------
*     Convierte números del sistema decimal al sistema binario
*     ----------------------------------------------------------------
*     Declaración de variables
*     ----------------------------------------------------------------
      implicit none
      integer max_digits          ! número maximo de digitos binarios
      parameter (max_digits = 32) ! de la parte entera y "decimal"
      integer bp(max_digits)      ! digitos binarios de la parte entera
      integer bf(max_digits)      ! digitos binarios de la parte decimal
*     --------------------------------
```

```
      character(*) ifield         ! variables para construir un
      parameter (ifield='(i2)')   ! formato dinámico de impresión
      character(2) sp,sf
      character(14) formato
*     --------------------------------
      integer in,out              ! unidades de entrada/salida estandar
      parameter (in=5,out=6)
*     --------------------------------
      integer p,d
      integer np,nf
      integer i
      real num,f
*     --------------------------------
      integer io
      logical go
*     ------------------------------------------------------------------
*     Proceder sobre la lista de numeros ingresados por la
*     entrada estandar
*     ------------------------------------------------------------------
      io = 0
      do while(io.eq.0)
*        ---------------------------------------------------------------
*        Leer el numero desde la entrada estandar
*        ---------------------------------------------------------------
         read(unit=in,fmt=*,iostat=io) num
         if (io.eq.0) then
*           ------------------------------------------------------------
*           Separar la parte entera de la parte decimal
*           ------------------------------------------------------------
            p = int(num)
            f = num-p
*           ------------------------------------------------------------
*           Convertir la parte entera
*           ------------------------------------------------------------
            go = .true.
            np = 0
            do while(go)
               np     = np + 1
               d      = mod(p,2)
               bp(np) = d
               p      = int(p/2)
               if(p.eq.0.or.np.ge.max_digits) go = .false.
            end do
*           ------------------------------------------------------------
*           Convertir la parte decimal
*           ------------------------------------------------------------
            go = .true.
            f  = 2.0*f
            nf = 0
            do while(go)
               nf     = nf + 1
               d      = int(f)
               bf(nf) = d
               f      = 2.0*(f-d)
               if(f.eq.0.0.or.nf.ge.max_digits) go = .false.
            end do
*           ------------------------------------------------------------
*           Construir el formato dinamicamente: (npi1,A1,nfi1)
```

```
*           ------------------------------------------------------------
            write(sp,fmt=ifield) np
            write(sf,fmt=ifield) nf
            formato = '(' // sp // 'i1,A1,' // sf // 'i1)'
*           ------------------------------------------------------------
*           Imprimir el numero binario en la salida estandar
*           ------------------------------------------------------------
            write(unit=out,fmt=formato) (bp(i),i=np,1,-1), '.',
     *                                  (bf(i),i=1,nf)
*           ------------------------------------------------------------
         endif
      enddo
      stop
      end
```