

El mito de la Variable

¿Qué es una variable?

Una variable no es más que una dirección. Usamos un nombre lógico como una etiqueta para referirnos más cómodamente a ese lugar de memoria donde está almacenado un contenido que nos interesa tratar.

La declaración de la variable normalmente incluye su tipo. Esto permite saber cuántas celdas de memoria ocupa el dato a partir de esa dirección que ha sido asociada a su nombre.

En el ejemplo para el DLX mostrado, la zona de datos se establece que empieza en la dirección 100. Entonces, el dato “n” de tipo **word** y valor 10 es la dirección 100 y desde esa dirección hasta la 103, los 4 bytes contienen un entero de 32 bits que es el tamaño asociado a la “palabra” (una palabra es un entero y es también el tamaño de los registros y muchas operaciones del repertorio operan con enteros). El valor 10 se almacenará, en este caso, en formato *big-endian*. Observar cómo se muestra esto en la primera fila de la siguiente tabla:

Declaración de Datos	[Nombre]: Dirección : contenido	Mostrado en DLXVSim
.data 100	n: 100 a 103 : 10	n: 10
c: .word 10	c: 104 a 107: 6	c: 6
c: .word 6,16,26,36	108 a 111: 16	c+0x4: 16
suma: .space 4	112 a 115: 26	c+0x8: 26
a: .double 10.4,2,3.7,8	116 a 119: 36	c+0xc: 36
cte: .float 3.1416, 1.6	suma: 120 a 123: 0	suma: 0
	a: 124 a 131: 10.4	a: 10.400000
	132 a 139: 2.0	a+0x8: 2.000000
	140 a 147: 3.7	a+0x10: 3.700000
	148 a 155: 8.0	a+0x18: 8.000000
	cte: 156 a 159: 3.14	cte: 3.141600
	160 a 163: 1.6	cte+0x4: 1.600000

A continuación de “n”, hemos dispuesto una serie de declaraciones de distinto tamaño (4 bytes para enteros (.word) y coma flotante de simple precisión (.float), 8 bytes para doble precisión (.double) y “space” solo reserva bytes). Si hay más de un valor separado por comas, se asignan éstos en la memoria. Por ejemplo, en el caso de *c*, cada entero de la lista se referencia como *c+4*, *c+8* ... Es permitido usar *n+4* o *c* puesto que es lo mismo. Así, *n+8*, *c+4* y *108* referencian la misma posición de memoria (la 108) y, por lo tanto pueden aparecer indistintamente en una instrucción del ensamblador.

Por ejemplo:

```
lw r7, n+8(r0)    lw r7, c+4(r0)    y    lw r7, 108(r0)
```

son lo mismo y cargan el número 16 en r7 tras acceder a la posición **108+r0** de memoria. Esto es:

```
r7 <-- M(108)    pues r0 siempre tiene 0 en el DLX
```

Nota: Las palabras reservadas que empiezan con un punto son directivas del ensamblador.

.data y **.text** son directivas del ensamblador para ubicar en esas direcciones a los segmentos de datos (variables) y texto (programa). Ver la AYUDA de la herramienta.