

## Arquitectura de Computadores I

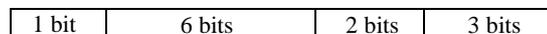
### Sistema de memoria 3: Segmentación + Bancos

En un sistema de memoria, en el que el direccionamiento se realiza a nivel de byte, las direcciones de acceso a memoria tienen la siguiente estructura:

- Memoria virtual segmentada:



- Memoria principal:



**Contesta a las siguientes preguntas:**

- Esquema de traducción de direcciones. ¿Cuántos segmentos puede tener un programa? ¿Cuál es el tamaño máximo de segmento? ¿Cuántas entradas tendrá la tabla de segmentos? ¿Cuál es el tamaño de cada entrada? En el caso de la existencia de un TLB, ¿de qué tamaño es cada una de sus entradas?
- Esquema de la memoria principal. ¿Cuántas palabras tiene? ¿Cuál es el tamaño en bytes de una palabra? ¿Cómo está organizada la memoria? En caso de estar formada por varios módulos, ¿cuál es el tamaño de un módulo?
- En un instante determinado, el procesador lanza la siguiente secuencia de referencias lógicas: 148, 412, 1496 y 428. El sistema dispone de un TLB para la traducción de direcciones, inicialmente vacío, con un tiempo de acceso de 30 ciclos en caso de fallo y 1 ciclo en caso de acierto. ¿Cuánto tiempo es necesario para la traducción de esa secuencia de referencias? Teniendo en cuenta el contenido de la tabla de segmentos indicada a continuación, ¿cuánto tiempo es necesario para acceder a memoria principal para dicha secuencia de referencias? El tiempo de acceso a memoria principal es de 10 ciclos (1 ciclo desde el buffer de entrelazado).

Tabla de segmentos: segmento	10	5	1	2
@base	3000	600	2700	100

- ¿Cuál será el tiempo de traducción para todas las referencias de este programa si está formado por 4 segmentos (instrucciones, vector A, vector B y vector C)?

<pre> for (i=2; i&lt;52; i++)   A[i] = (B[i-2]+B[i]+B[i+2]) / C[0];         </pre>	<pre> movi r1,#16 movi r10,#4320 movi r6,#2 buc: load r2,B[r1-16]       load r3,B[r1]       load r4,B[r1+16]       add r2,r2,r3       add r2,r2,r4       load r5,[r10]       divi r2,r2,r5       store r2,64[r1]       addi r1,r1,#8       addi r6,r6,#1       subi r7,r6,#511       bnz r7,buc         </pre>
--	--