

Dispersión Extensible

Desarrollo de un Caso de Estudio

Se considera un archivo con bloques con capacidad para dos registros.

Se incorporarán registros con claves 279, 487, 392, 651, 548, 120, 774, 50.

Marco Conceptual y Procedimental de la Organización

Esta organización de archivos consiste en una **tabla de dispersión**, que se persiste como un archivo secuencial de números relativos de bloque, y un **archivo de bloques de registros**, cuyos números relativos aparecen en la tabla. Es decir, la función de dispersión aplicada a un registro devuelve una posición de la tabla de dispersión que contiene el número relativo de bloque del archivo donde ubicar el registro.

La **función de dispersión** para un registro con clave numérica k , es $d(k)=k \bmod t$, siendo t el tamaño de la tabla de dispersión.

Como el tamaño de la tabla de dispersión es siempre una potencia de 2, el módulo tamaño de la tabla de una clave, $k \bmod t$, implica que el resultado de la función equivale a tomar los $\log_2(k)$ bits menos significativos (finales) de la representación binaria de k . Por ejemplo, si $k=396$ y $t=8$, $396 \bmod 8=4$. Si se considera la representación de la clave en numeración binaria, $k=1\ 1000\ 1100$, y $\log_2(8)=3$: los últimos 3 bits de la clave k son 100, que en decimal equivalen a 4.

Por tanto, los **bits de dispersión de la tabla** son los que se necesitan para representar en numeración binaria a los números enteros desde 0 hasta $t-1$, es decir a las posiciones (direcciones de registros) de la tabla de dispersión: $\log_2(t)$.

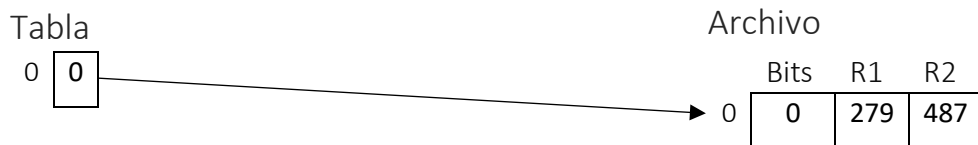
En los bloques del archivo de registros se requiere disponer como información de control de los **bits de dispersión del bloque**, que representan los bits menos significativos *invariantes* de los registros que contiene. Los bits de dispersión de un bloque se actualizan al producirse su desborde, sumando 1 a sus bits de dispersión y asignando también este valor al bloque que se agrega a la tabla para dispersar nuevamente los registros del bloque desbordado y repartirlos entre ambos bloques según la función de dispersión.

Los bits de dispersión de un bloque implican cuántas veces aparece el número de ese bloque en la tabla de dispersión, y a qué distancia circular entre apariciones. Lo primero se calcula por el cociente entre el tamaño de la tabla y la potencia de 2 con exponente igual a los bits de dispersión del bloque, y la distancia es la potencia de 2 con exponente igual a los bits de dispersión del bloque.

Cuando un bloque se desborda, si su cantidad de apariciones es 1, se debe duplicar la tabla de manera que la primera mitad sea igual a la segunda, y en la posición donde se obtuvo el número del bloque desbordado se reemplaza al número de ese bloque por el del que se agrega al archivo. En cambio, si su cantidad de apariciones en la tabla es mayor que 1, se reemplaza a la mitad de sus apariciones en la tabla por el número del nuevo bloque, comenzando por la posición que dio como resultado la función dispersión aplicada al registro que produjo el desborde, y respetando las distancias que correspondan a los bits de dispersión del nuevo bloque.

Desarrollo

Se crea la organización vacía, es decir, una tabla de dispersión con una sola posición-registro, que apunta al primer bloque del archivo, el 0, vacío. Los bits de dispersión de la tabla y del bloque vacío son $\log_2(1)=0$. Esto significa que para dispersar los dos primeros registros se aplicará la función $d(k)=k \bmod 1=0$: todos los registros se ubican en el bloque de la tabla que aparece en la posición 0, lo que equivale a no considerar ningún bit de dispersión de sus claves.

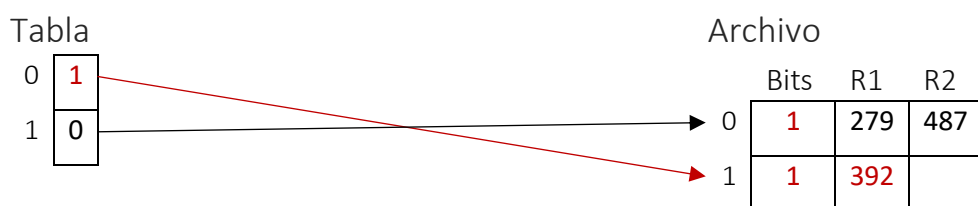


+392

$392 \bmod 1 = 0$ (va al bloque referido en la posición 0 de la tabla, el 0).

Al desbordarse el bloque 0, se aumenta en 1 su cantidad de bits de dispersión y se inicializa un nuevo bloque con este mismo valor para sus bits de dispersión. Como la cantidad de bits de dispersión del bloque que se desborda es igual a la cantidad de bits de dispersión de la tabla, se duplica la tabla, y en la posición que se obtuvo como resultado de la función de dispersión se agrega el número del nuevo bloque.

Luego se dispersan los registros del bloque 0.



$297 \bmod 2 = 1$ (queda en el bloque 0)

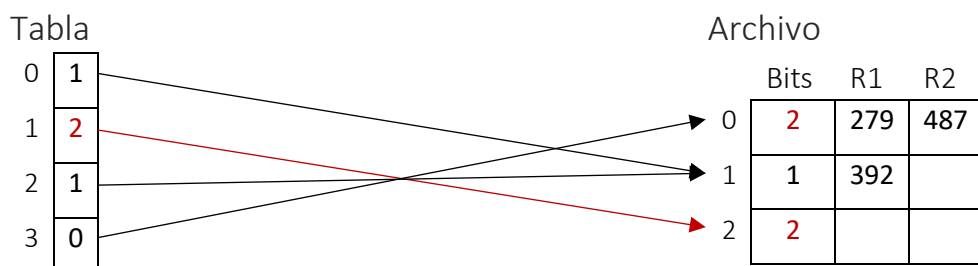
$487 \bmod 2 = 1$ (queda en el bloque 0)

Al reintentar el alta, $392 \bmod 2 = 0$ (va al bloque 1)

El costo en accesos de esta operación fue L0, E0, E1.

+651

$651 \bmod 2 = 1$ (va al bloque 0, que vuelve a desbordarse).

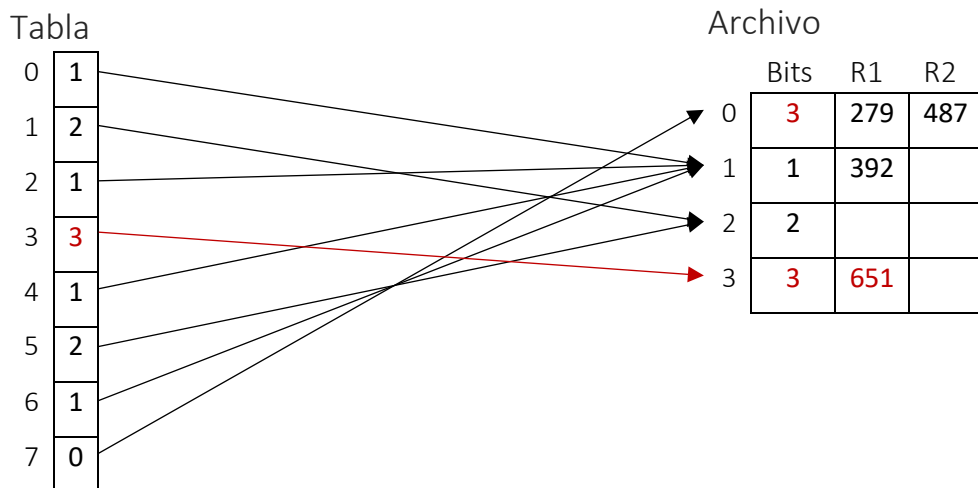


$279 \bmod 4 = 3$ (queda en el bloque 0)

$487 \bmod 4 = 3$ (queda en el bloque 0)

$651 \bmod 4 = 3$ (va al bloque 0, que se desborda nuevamente)

Hasta este punto, el costo de la operación es L0, E2.



$279 \bmod 8 = 7$ (queda en el bloque 0)

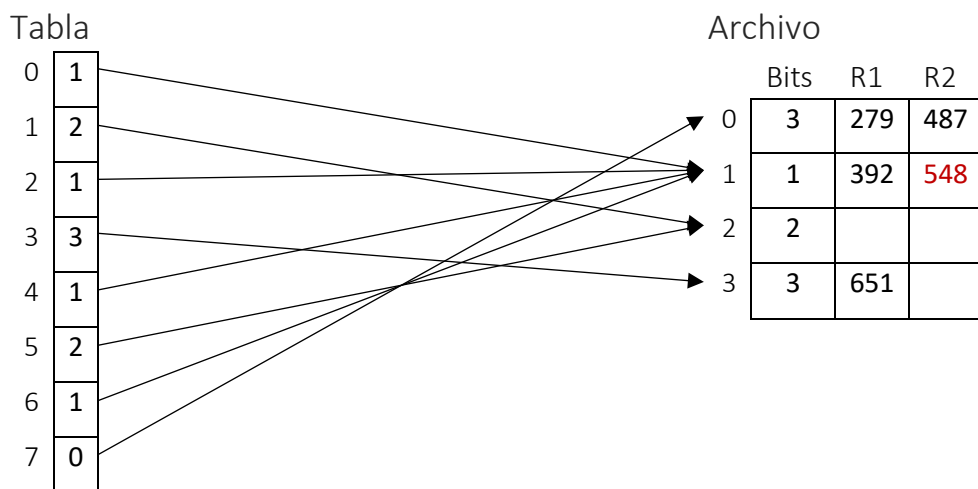
$487 \bmod 8 = 7$ (queda en el bloque 0)

$651 \bmod 8 = 3$ (va al bloque 3)

Al costo previo de agrega E0, E3.

+548

$548 \bmod 8 = 4$ (va al bloque 1)



Costo: L1, E1.

+120

$120 \bmod 8 = 0$ (va al bloque 1, que se desborda)

Como la cantidad de bits de dispersión del bloque 1 es 1, quiere decir que el bloque 1 aparece $8/2^1=4$ veces en la tabla; entonces, desde la posición 0, que dio como resultado la función de dispersión para ubicar al registro con calve 120, se reemplaza $8/2^2=2$ veces el número de bloque 2 por el número de bloque 4, a distancia $2^2=4$.

Tabla

0	4
1	2
2	1
3	3
4	4
5	2
6	1
7	0

Archivo

	Bits	R1	R2
0	3	279	487
1	2		
2	2		
3	3	651	
4	2	392	548

$392 \bmod 8 = 0$ (va al bloque 4)

$548 \bmod 8 = 4$ (va al bloque 4)

No se puede insertar el registro con clave 120 en el bloque 4, referido en la posición 0, porque se desborda: hay que resolver el desborde recursivamente.

Hasta este punto, el costo de la operación es L1, E1.

Tabla

0	5
1	2
2	1
3	3
4	4
5	2
6	1
7	0

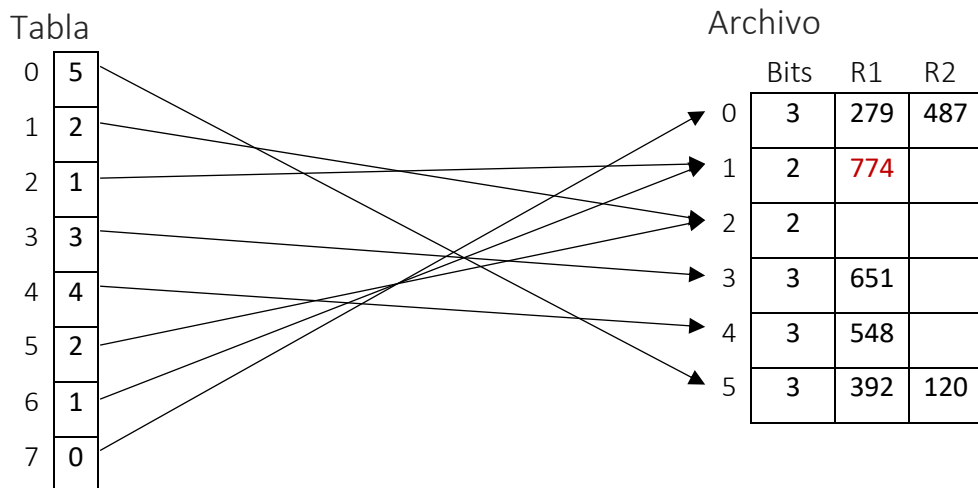
Archivo

	Bits	R1	R2
0	3	279	487
1	2		
2	2		
3	3	651	
4	3	548	
5	3	392	120

Al costo previo se agrega E4, E5.

+774

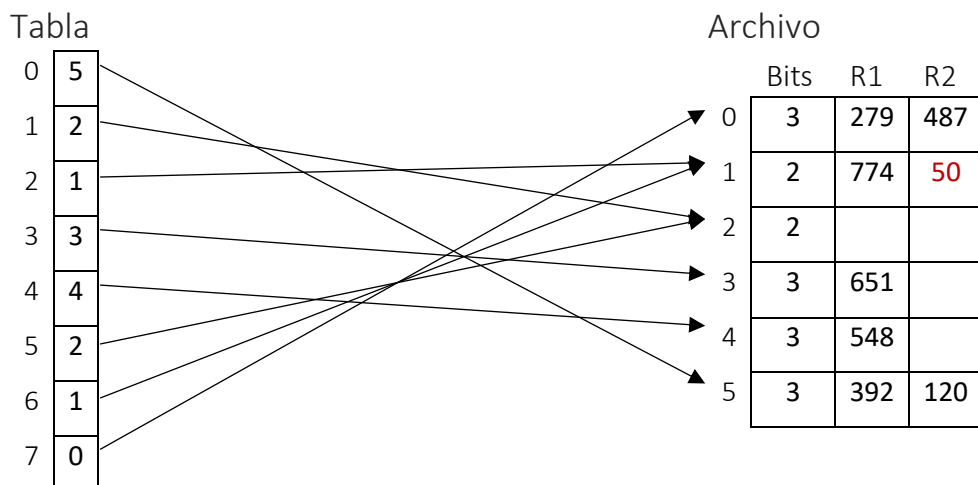
$774 \bmod 8 = 6$ (va al bloque 1)



Costo: L1, E1.

+50

$50 \bmod 8 = 2$ (va a bloque 1)



Costo: L1, E1.

Análisis de Integridad de Estado Final

Se analiza la integridad del estado final, aunque el mismo análisis puede efectuarse al estado posterior a cualquier operación. Para esto se cambia la notación de posiciones de la tabla y de las claves de registros a numeración binaria. Las claves se pasan a numeración binaria a efectos de comprobar que la cantidad de dígitos binarios menos significativos invariantes coincide con la cantidad de bits de dispersión del bloque.

Obsérvese que los bits de dispersión de la tabla son $\log_2(8)=3$, y que los bloques 0, 3, 4 y 5, que tienen 3 bits de dispersión, deben aparecer $8/2^3=1$ vez en la tabla de dispersión, y que los bloques 1 y 2, que tienen 2 bits de dispersión, deben aparecer $8/2^2=2$ veces en la tabla de dispersión, a distancia circular $2^2=4$ una aparición de la siguiente. Se comprueba que el bloque 1 aparece en las posiciones 2 y 6 de la tabla, que están a distancia 4 entre ellas ($6=2+4$ y $2=(6+4) \bmod 8$), y que el bloque 2 aparece en las posiciones 1 y 5, también a 4 posiciones de distancia entre ellas.

Tabla

000	5
001	2
010	1
011	3
100	4
101	2
110	1
111	0

Archivo

Bits	R1	R2
0	3	000100010 111
1	2	001100000 110
2	2	
3	3	001010001 011
4	3	001000100 100
5	3	000110001 000

Bajas

Si bien no se contemplará esta operación en la evaluación del tema, se proporcionan ejemplos a modo informativo.

Marco Procedimental

Cuando tras una baja un bloque queda vacío, puede suceder que éste pueda liberarse y por consiguiente desaparecer toda referencia a él en la tabla de dispersión, o puede que no sea posible liberarlo.

Para mantener la integridad de la organización, si un bloque queda vacío, para que pueda liberarse debe existir a la mitad de la distancia entre sus apariciones en la tabla un mismo número de bloque con igual cantidad de bits de dispersión b del vacío. Si esto es así, se puede sustituir en la tabla el número del bloque vacío por el número del bloque equidistante a 2^{b-1} posiciones en la tabla. Para comprobar la existencia de un bloque con tales características, basta con buscar en la tabla, a partir de la posición donde se obtuvo el número de bloque que quedó vacío tras la baja, 2^{b-1} posiciones antes y después (con desplazamiento circular, si fuera necesario); si en ambas posiciones se encuentra un mismo número de bloque, se sustituyen todas las apariciones del número del bloque vacío por el número de bloque encontrado, y se accede a este último bloque para descontar 1 a sus bits de dispersión (ahora su cantidad de apariciones en la tabla se duplica, y la distancia entre apariciones se reduce a la mitad).

Ejemplos

Tabla

0	5
1	2
2	1
3	3
4	4
5	2
6	1
7	0

Archivo

Bits	R1	R2
0	3	279
1	2	774
2	2	
3	3	651
4	3	548
5	3	392

-651

$651 \bmod 8 = 3$ (bloque 3 queda vacío con 3 bits de dispersión)

A distancia $2^{3-1}=4$ antes y después de la posición 3 de la tabla, se obtiene la misma posición 7, con el mismo número de bloque, 0.

Tabla

0	5
1	2
2	1
3	0
4	4
5	2
6	1
7	0

Archivo

	Bits	R1	R2
0	2	279	487
1	2	774	50
2	2		
3	3	651	
4	3	548	
5	3	392	120

Libres: 3

Costo: L3, L0, E0 (no es necesario grabar el bloque que se libera, puesto que para un *overflow* futuro, se sobrescribirá con otro contenido)

-548

$548 \bmod 8 = 4$ (bloque 4 queda vacío con 3 bits de dispersión)

A distancia $2^{3-1}=4$ antes y después de la posición 4 de la tabla, se obtiene la misma posición 0, con el mismo número de bloque, 5.

Tabla

0	5
1	2
2	1
3	0
4	5
5	2
6	1
7	0

Archivo

	Bits	R1	R2
0	2	279	487
1	2	774	50
2	2		
3	3	651	
4	3	548	
5	2	392	120

Libres: 3, 4

Costo: L4, L5, E5

Y como se observa que la segunda mitad de la tabla coincide con la primera mitad, ésta se puede reducir a la mitad:

Tabla

0	5
1	2
2	1
3	0

Archivo

	Bits	R1	R2
0	2	279	487
1	2	774	50
2	2		
3	3	651	
4	3	548	
5	2	392	120

Libres: 3, 4