

Tiempo de acceso a los datos del disco

Tiempo de búsqueda (tb)

Para colocar la cabeza sobre la pista (mecánico)

Depende de la distancia a recorrer

Es el principal retardo: 10-14 mseg en PCs y 8-9 mseg en servidores

Retardo rotacional (rr) (o latencia)

Esperar a que el principio del bloque pase bajo la cabeza

Caso mejor: acceso inmediato

Caso peor: una vuelta entera

Principal retardo en discos de cabeza fija

Tiempo de transferencia de bloque (ttb)

Depende del tamaño del bloque, de las pistas y rpm

Suele costar 1-2 ms/bloque

Registro, bloque y búfer

Registro: entidad (E/R), tupla (relacional), fila (SQL), ...

Bloque o RF o sector del disco:

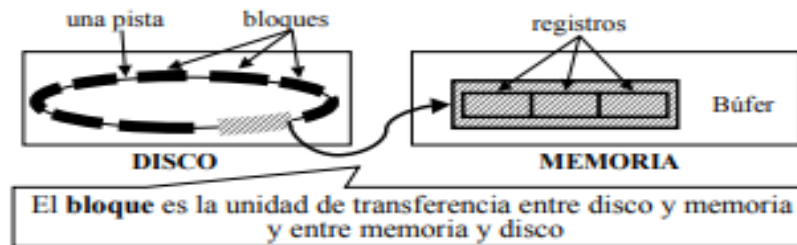
Unidad de transferencia entre disco y memoria

Un Bloque puede contener uno o varios registros lógicos

Acceso directo (o aleatorio) a un bloque o sector o RF:

La dirección de un bloque la forman: el nº de superficie, nº de pista y nº de bloque

Búfer: espacio en memoria principal en el que cabe un bloque



Tiempo para transferir un bloque entre disco y memoria (o viceversa): entre 12 y 60 mseg.

Tiempo bastante alto en comparación con lo que tarda la CPU en procesar sus registros lógicos (cuando están en el búfer)

La localización de datos en disco es un cuello de botella importante en las aplicaciones de BD

Factor de bloqueo

Bloque: unidad de transferencia de datos entre disco y memoria

Programas:

trabajan con registros lógicos

Un bloque puede contener varios registros lógicos

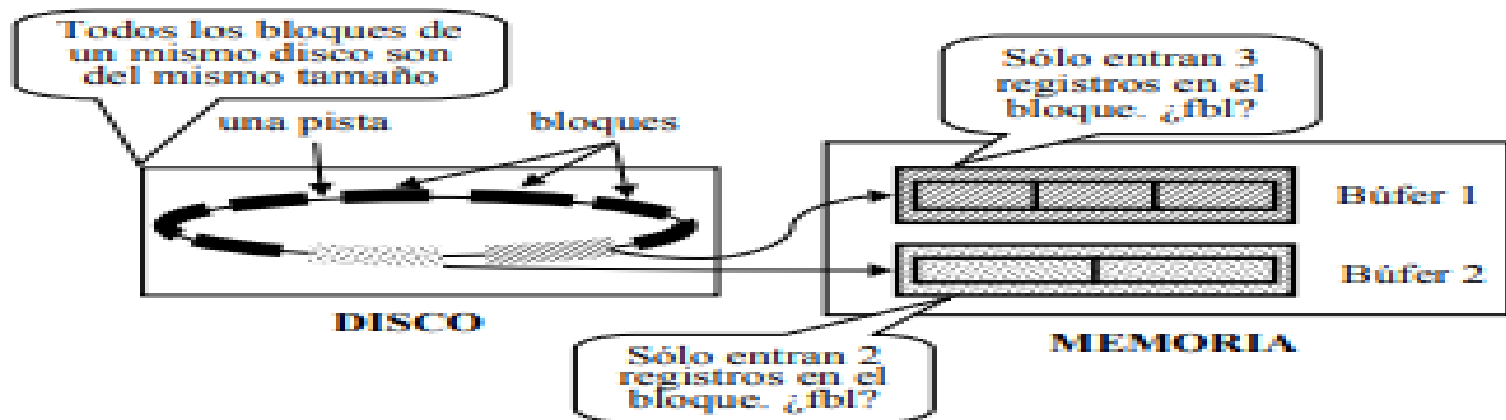
Factor de bloqueo: cantidad de registros lógicos que entran en un bloque

$FB = (\text{Tamaño en bytes del B} / \text{Tamaño en bytes del RL registros lógicos por bloque})$

$B = n^{\circ}$ bytes por bloque

$RL = n^{\circ}$ bytes por registro (tamaño fijo)

Puede quedar espacio desocupado en los bloques (fragmentación interna), $B - (FB * RL)$ bytes



Técnicas de asignación de los bloques de un fichero en disco

Contigua: bloques consecutivos de disco. Facilita el doble búfer y dificulta el expandir el fichero

Enlazada: cada bloque tiene un apuntador al siguiente bloque. Facilita la expansión y dificulta la lectura

Grupos de bloques: combinación de las anteriores, un grupo de bloques consecutivos enlazado con el siguiente grupo consecutivo

Indexada: bloques que contienen sólo índices. Los índices son apuntadores a los bloques que realmente contienen los datos

Técnicas de direccionamiento calculado (hashing)

Acceso muy rápido para ciertas búsquedas

Los ficheros de direccionamiento calculado se llaman también ficheros dispersos o directos

Condición de búsqueda: igualdad sobre el campo de direccionamiento calculado (c.dir.cal.)

El c.dir.cal. en la mayor parte de los casos es clave (valores únicos)

Función de direccionamiento calculado o de aleatorización (f. hash)

Se aplica a valores del c.dir.cal. del registro

Resultado: dirección de un bloque de disco

En el bloque está el registro lógico con ese valor de c.dir.cal.

El registro lógico se busca en el búfer

Suele necesitarse un único acceso a bloques de disco

Ejercicio 1

Tenemos Un Archivo ordenado

Cantidad de RL = 30.000 registros lógicos

Registros de longitud fija de R L= 100 Bytes

Tamaño de Bloque B = 1024 bytes

Factor de bloqueo FB = $B/R = 1024/100 = 10$ RL por bloque

Calcular

El número de bloques requerido para el archivo es $b = ?$ bloques.

Cantidad de accesos a bloques en una búsqueda binaria en el archivo de datos.

Ejercicio 1 Solución

Tenemos Un Archivo ordenado

Cantidad de RL = 30.000 registros

Registros de longitud fija de RL = 100 Bytes

Tamaño de Bloque B = 1024 bytes

Factor de bloqueo FB = $B/RL = 1024/100 = 10$ RL por bloque

Calcular

El número de bloques requerido para el archivo es $b = 3000$ bloques.

Cantidad de accesos a bloques en una búsqueda binaria en el archivo de datos.

$\log_2 b = (\log_2 3.000) = 12$ accesos a bloques.

Fragmentación Interna del Bloque = $1024 \text{ Bytes}/B - (10 \text{ RL} /B * 100 \text{ Bytes}/RL) = 24 \text{ Bytes}$

Ejercicio 1 Índice primario

Ahora tenemos todo lo anterior más

Un campo clave de ordenación del archivo que tiene $V = 9$ bytes de largo.

Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice primario para el archivo.

Calcular

El tamaño de cada entrada del archivo índice es $R = ?$ Bytes.

El factor de bloqueo del archivo índice $fbI = ?$ RL por bloque.

El número total de entradas del archivo índice $r = ?$

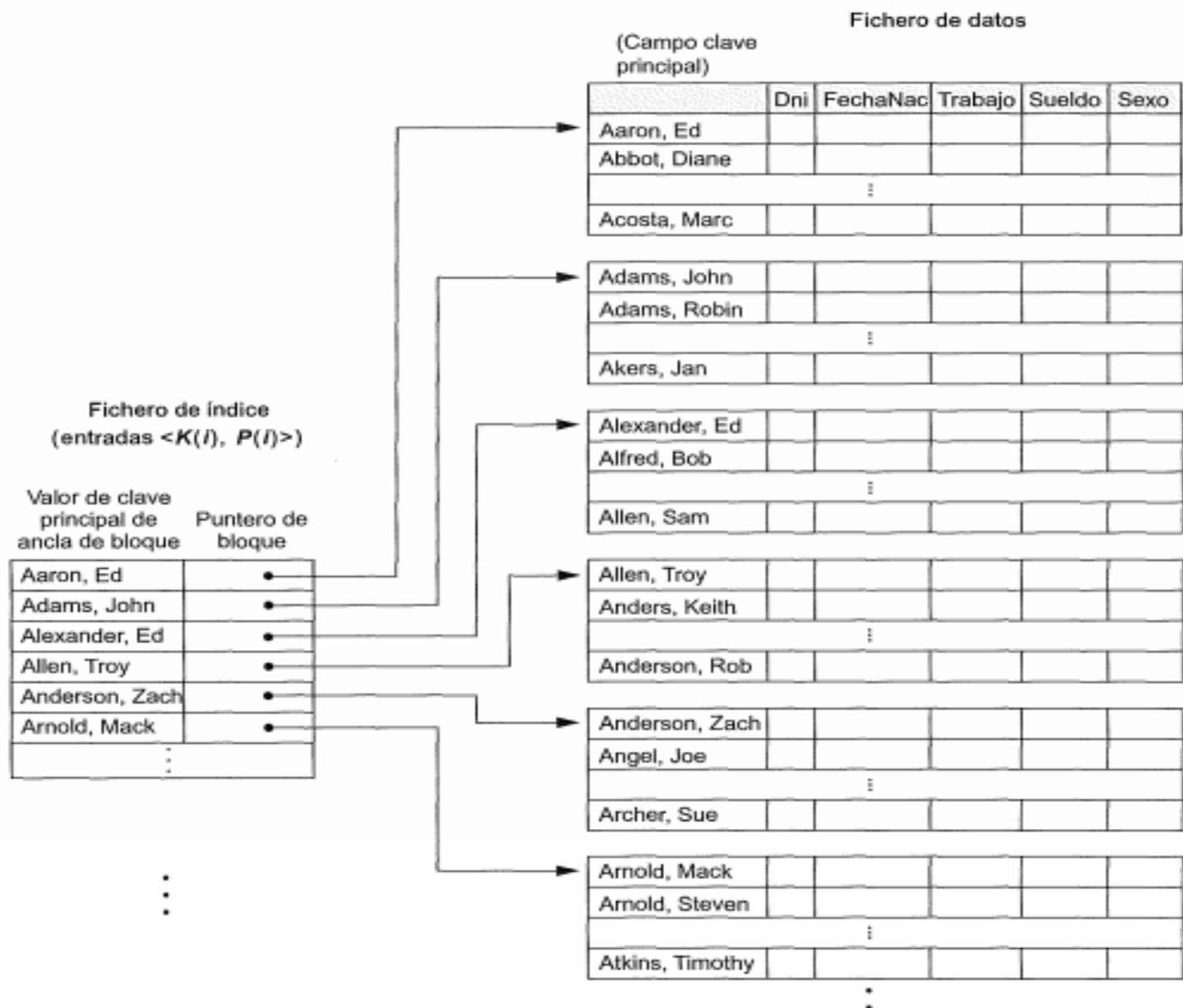
El número de bloques requerido para el archivo índice $b = ?$ bloques.

Cantidad de accesos en una búsqueda binaria en el archivo índice ?

Cantidad de accesos para encontrar un registro empleando el archivo índice ? accesos a bloques

¿Cuál es la mejor forma de accesos?

Figura 14.1. El índice principal en el campo clave de ordenación del fichero de la Figura 13.7.



Ejercicio 1 Índice primario Solución

Ahora tenemos todo lo anterior más

Un campo clave de ordenación del archivo que tiene $V = 9$ bytes de largo.

Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice primario para el archivo.

Calcular

El tamaño de cada entrada del archivo índice es $R = 15$ Bytes.

El factor de bloqueo del archivo índice $FB = 68$ RL por Bloque.

El número total de entradas del archivo índice $r = 3000$

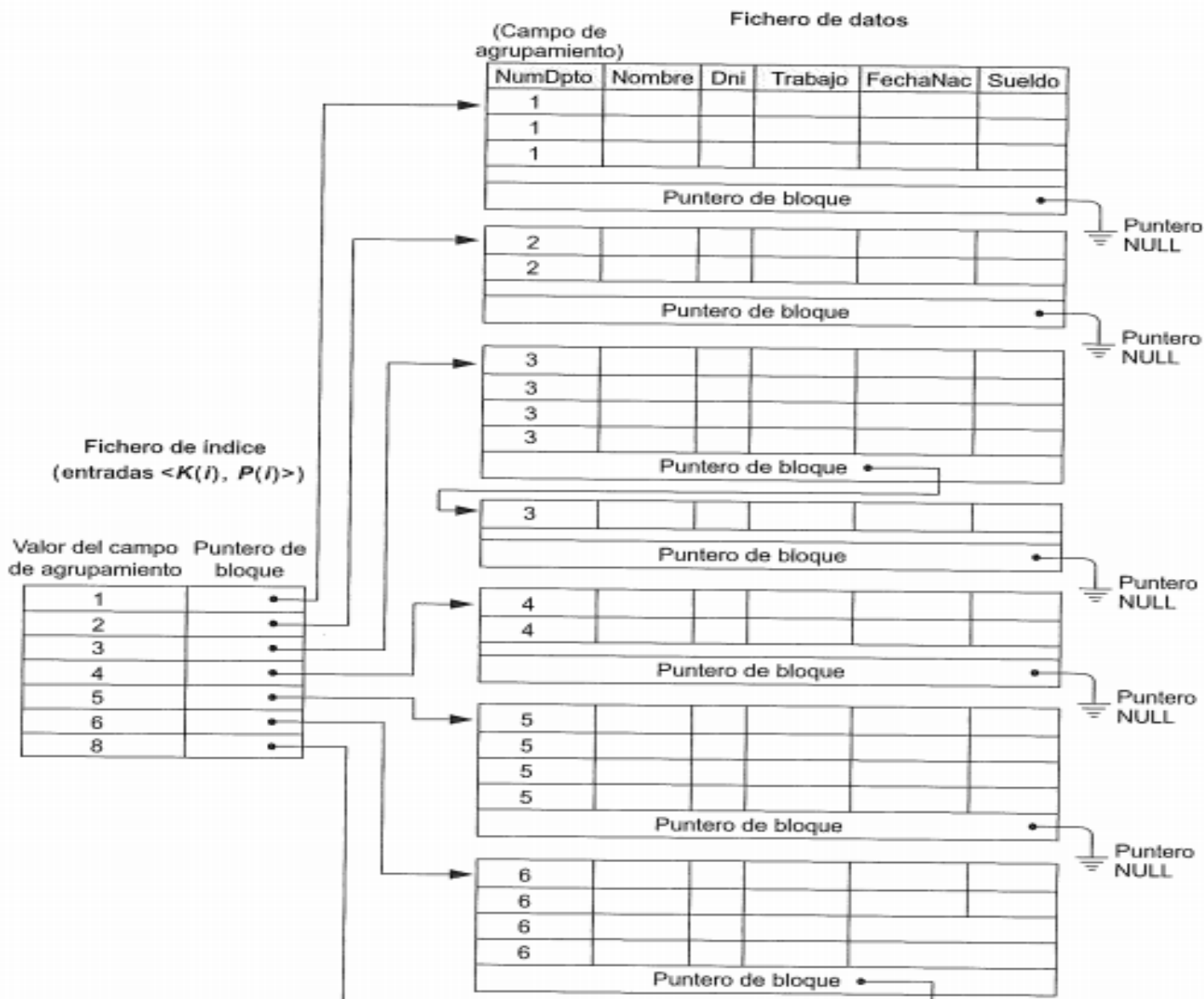
El número de bloques requerido para el archivo índice $b = 45$ bloques.

Cantidad de accesos en una búsqueda binaria en el archivo índice es $(\log_2 b_i) = (\log_2 45) = 6$ accesos a bloques.

Cantidad de accesos para encontrar un registro empleando el archivo índice $6 + 1 = 7$ accesos a bloques (una mejora sobre la búsqueda binaria en el fichero de datos, que requería 12 accesos a bloques).

¿Cuál es la mejor forma de accesos?

Figura 14.3. Índice agrupado con un grupo de bloques separados por cada grupo de registros que comparten el mismo valor del campo agrupado.



Solución Ejercicio 1 Índice primario

Supongamos que tenemos un archivo ordenado de (**$r = 30000$ registros**) almacenados en un disco con un (**tamaño de Bloque $B = 1024$ bytes**). Los registros del archivo son de (**longitud fija de $R = 100$ Bytes**). (**El factor de bloqueo $fbl = B/R = 1024/100 = 10$ registros por bloque**). El número de bloques requerido para el archivo es $b = r/fbl = 30000/10 = 3000$ bloques. Una búsqueda binaria en el archivo de datos requeriría aproximadamente $(\log_2 b) = \log_2 3000 = 12$ accesos a bloques.

Supongamos que el campo clave de ordenación del archivo tiene $V = 9$ bytes de largo, que un apuntador a bloque tiene $P = 6$ bytes de largo y que hemos construido un índice primario para el archivo.

El tamaño de cada entrada del índice es $R = 9+6=15$ bytes, de modo que el factor de bloqueo del índice es $fbl = B/R = 1024/15 = 68$ entradas por bloque. El número total de entradas del índice, r , es igual al número de bloques del archivo de datos, que es 3000. Así el número de bloques requerido para el índice es $b = r/fbl = 3000/68 = 45$ bloques. Efectuar una búsqueda binaria en archivo de índice requeriría $\log_2 b = \log_2 45 = 6$ accesos a bloques. Para encontrar un registro empleando el índice necesitaríamos un acceso adicional a un bloque del archivo de datos, para un total de $6 + 1 = 7$ accesos a bloques, mucho mejor que la búsqueda binaria en el archivo de datos que requiere 12 accesos a bloques.

Ejercicio 2 Índice secundario denso

Tenemos Un Archivo ordenado con Un campo clave de ordenación del archivo que tiene $V = 9$ bytes de largo. Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice secundario por un campo no clave secundario.

$r = 30000$ registros

Tamaño de Bloque $B = 1024$ bytes

Registros de longitud fija de $R = 100$ Bytes

El número de bloques requerido para el archivo es $b = 3000$ bloques.

Factor de bloqueo $fbl = B/R = 1024/100 = 10$ registros por bloque.

Cantidad total de entradas del índice ?.

El número de bloques necesarios para el índice es de ? bloques.

Para efectuar una búsqueda lineal en el fichero, necesitaríamos acceder a ? bloques por término medio.

Una búsqueda binaria en este índice secundario necesita acceder a ? bloques.

Ejercicio 2 Índice secundario denso Solución

Tenemos Un Archivo ordenado con Un campo clave de ordenación del archivo que tiene $V = 9$ bytes de largo. Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice secundario por un campo no clave secundario.

$r = 30000$ registros

Tamaño de Bloque $B = 1024$ bytes

Registros de longitud fija de $R = 100$ Bytes

El número de bloques requerido para el archivo es $b = 3000$ bloques.

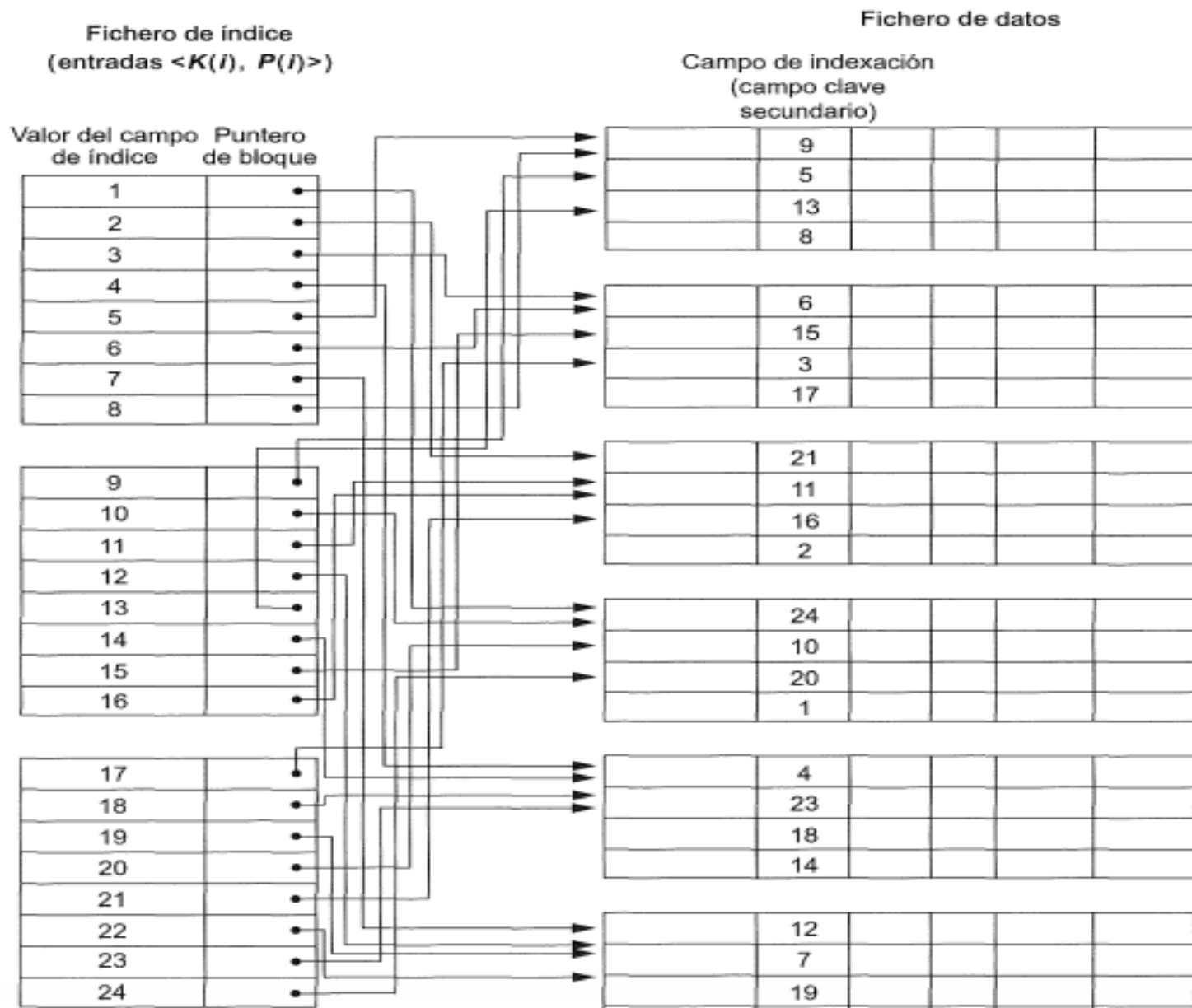
Factor de bloqueo $fbl = B/R = 1024/100 = 10$ registros por bloque.

En un índice secundario denso como éste, la cantidad total de entradas del índice es igual al número de registros del fichero de datos, es decir, 30000. El número de bloques necesarios para el índice es, por tanto, $b = (r / fbl) = (30000/10) = 3000$ bloques.

Para efectuar una búsqueda lineal en el fichero, necesitaríamos acceder a $b/2 = 3000/2 = 1500$ bloques por término medio.

Una búsqueda binaria en este índice secundario necesita acceder a $r (\log_2 b) = 30000 (\log_2 3000) = 9$ bloques.

Figura 14.4. Índice secundario denso (con punteros de bloque) en un campo clave desordenado de un fichero.



Ejercicio 2 Índice secundario denso

Ahora tenemos todo lo anterior más

Un índice secundario sobre un campo clave no de ordenación del archivo que tiene $V = 9$ bytes de largo. Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice primario para el archivo.

Calcular

El tamaño de cada entrada del archivo índice es $R = ?$ Bytes.

El factor de bloqueo del archivo índice $fbl = ?$ entradas por bloque.

En un índice secundario denso como éste, el número total de entradas del archivo índice, r , es de ?.

¿Cuál es el número total de bloques para el archivo índice?

Compárelo con el total de bloques del archivo índice primario no denso del ejercicio 1

¿Cuántos accesos a bloque necesita una búsqueda binaria en este archivo de índice secundario?

¿Cuántos accesos a bloque necesita una búsqueda de un registro empleando el archivo índice secundario denso?

Ejercicio 2 Índice secundario denso Solución

Ahora tenemos todo lo anterior más

Un índice secundario sobre un campo clave no de ordenación del archivo que tiene $V = 9$ bytes de largo. Un apuntador a bloque que tiene $P = 6$ bytes de largo y que hemos construido un índice primario para el archivo.

Calcular

El tamaño de cada entrada del archivo índice es $R = (9+6) = 15$ Bytes.

El factor de bloqueo del archivo índice $fbl = (1024 / 15) = 68$ entradas por bloque.

En un índice secundario denso como éste, el número total de entradas del archivo índice es igual al número de registros del archivo de datos que es 30000.

¿Cuál es el número total de bloques para el archivo índice? $30000/68 = 442$ bloques

Compárelo con el total de bloques del archivo índice primario no denso del ejercicio 1

¿Cuántos accesos a bloque necesita una búsqueda binaria en este archivo de índice secundario? $\log_2 442 = 9$ accesos.

¿Cuántos accesos a bloque necesita una búsqueda de un registro empleando el archivo índice secundario denso? $9 + 1 = 10$ accesos.

Solución Ejercicio 2 Índice secundario denso

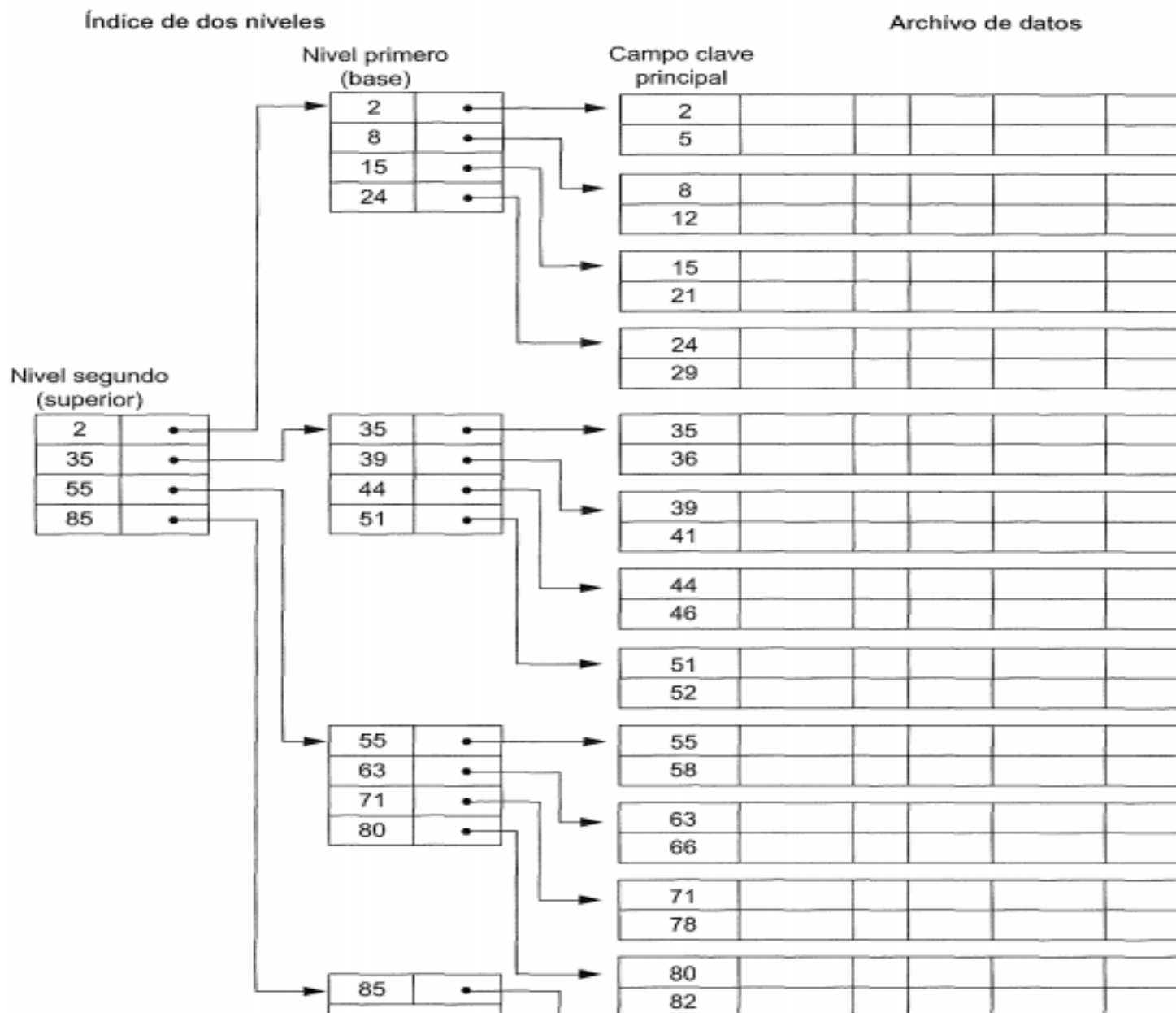
EJEMPLO 2: Consideremos el archivo del ejemplo 1 con $r = 30\ 000$ registros de longitud fija de tamaño $R = 100$ bytes, almacenado en un disco cuyos bloques son de $B = 1024$ bytes. El archivo tiene $b = 3000$ bloques, según se calculó en el ejemplo 1. Para efectuar una búsqueda lineal en este archivo tendríamos que examinar $b/2 = 3000/2 = 1500$ accesos a bloques en promedio. Supongamos que construimos un índice secundario basado en un campo clave (no de ordenamiento) del archivo que tiene $V = 9$ bytes de longitud. Al igual que en el ejemplo 1, un apuntador a bloque tiene 6 bytes de largo, así que cada entrada de índice tiene $R_i = (9 + 6) = 15$ bytes, y el factor de bloques del índice tiene $fbli = \lfloor (B/R_i) \rfloor = \lfloor (1024/15) \rfloor = 68$ entradas por bloque. En un índice secundario denso como éste, el número total de entradas de índice, r_i , es igual al número de registros del archivo de datos, que es 30 000. El número de bloques requeridos para el índice será entonces $b_i = \lceil (r_i/fbli) \rceil = \lceil (30\ 000/68) \rceil = 442$ bloques. Compárese esto con los 45 bloques que necesita el índice primario no denso del ejemplo 1.

Una búsqueda binaria en este índice secundario requiere $\lceil (\log_2 b_i) \rceil = \lceil (\log_2 442) \rceil = 9$ accesos a bloques. Para encontrar el registro empleando el índice, requeriremos un acceso adicional a un bloque del archivo de datos para un total de $9 + 1 = 10$ accesos a bloques; mucho mejor que los 1500 accesos a bloques que requiere en promedio una búsqueda lineal en este archivo, pero no tan bueno como los 7 accesos a bloques que se necesitan con el índice primario. ■

Ejercicio 3 Índice de múltiples niveles

EJEMPLO 3: Suponga que el índice secundario denso del ejemplo 2 se convierte en un índice de múltiples niveles. Calculamos un factor de bloques $fbl_i = 68$ entradas de índice por bloque, cifra que también es el abanico fo para el índice de múltiples niveles. También se calculó el número de bloques del primer nivel $b_1 = 442$ bloques. El número de bloques del segundo nivel será $b_2 = \lceil (b_1/fo) \rceil = \lceil (442/68) \rceil = 7$ bloques, y el número de bloques del tercer nivel será $b_3 = \lceil (b_2/fo) \rceil = \lceil (7/68) \rceil = 1$ bloque. Por tanto, el tercer nivel es el nivel superior del índice, y $t = 3$. Para tener acceso a un registro buscando en el índice de múltiples niveles, deberemos tener acceso a un bloque en cada nivel y a uno más en el archivo de datos, así que necesitamos $t + 1 = 3 + 1 = 4$ accesos a bloques. Compárese esto con el ejemplo 2, donde se requirieron 10 accesos a bloques empleando un índice de un solo nivel y una búsqueda binaria. ■

Figura 14.6. Índice principal de dos niveles que se parece a la organización ISAM.



**Terminamos
por hoy esta parte**