

Introducción a los Sistemas Operativos

Cache de Disco



I.S.O.

- ✓ Versión: Mayo 2013
- ✓ Palabras Claves: Unix, Buffer Cache, Cache, Disco, Bloque, Archivos

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.



Disk Cache

- ✓ Buffers en memoria principal para almacenamiento temporario de sectores de disco.
- ✓ Contienen una copia de algunos sectores de disco.
- ✓ Objetivo: MINIMIZAR LA FRECUENCIA DE ACCESO AL DISCO



Algunas observaciones

- ☑ Cuando un proceso quiere acceder a un bloque de la cache hay dos alternativas:
 - Se copia al espacio de direcciones de usuario
 - O se trabaja como memoria compartida (no se copia permitiendo acceso a varios procesos)



Estrategia de reemplazo

- ✓ Cuando se necesita un buffer para cargar un nuevo bloque, se elige el que hace más tiempo que no es referenciado.
- ✓ Es una lista de bloques, donde el último es el más recientemente usado (LRU, Least Recently Used)
- ✓ Cuando un bloque se referencia o entra en la cache queda al final de la lista
- ✓ No se mueven los bloques en la memoria: se asocian punteros.
- ✓ Otra alternativa: Least Frequently Used. Se reemplaza el que tenga menor número de referencias



Introducción a los Sistemas Operativos

Buffer Cache
Unix System V



Facultad de Informática
UNIVERSIDAD NACIONAL DE LA PLATA

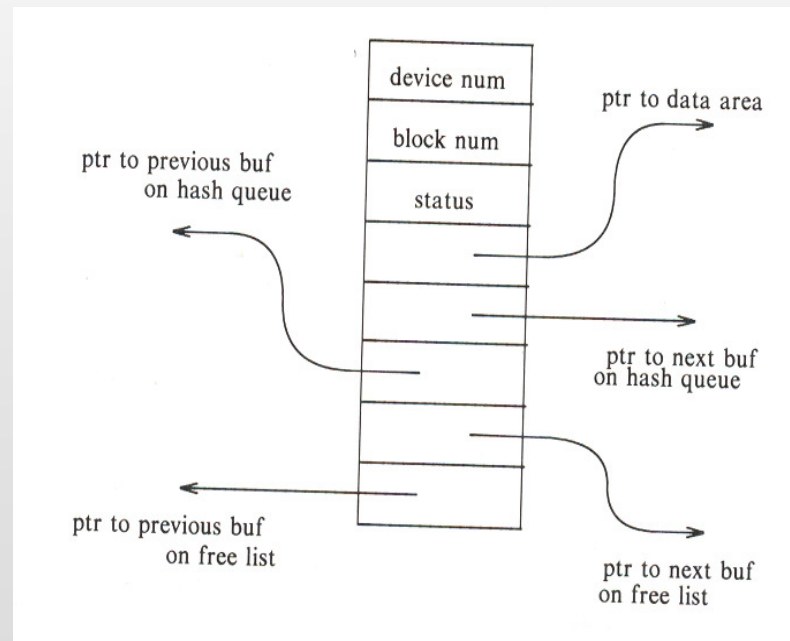
Objetivo y estructura

- ✓ Minimizar la frecuencia de acceso a disco
- ✓ Es una estructura formada por buffers
- ✓ El kernel asigna un espacio en la memoria durante la inicialización para esta estructura.
- ✓ Un buffer tiene dos partes: el header y el lugar donde se almacena el bloque de disco traído a memoria



El header

- ☑ Identifica por nro. de dispositivo y nro. de bloque
- ☑ Tiene punteros
 - ✓ 2 punteros para la hash queue
 - ✓ 2 punteros para la free list
 - ✓ un puntero al bloque en memoria
- ☑ Estado



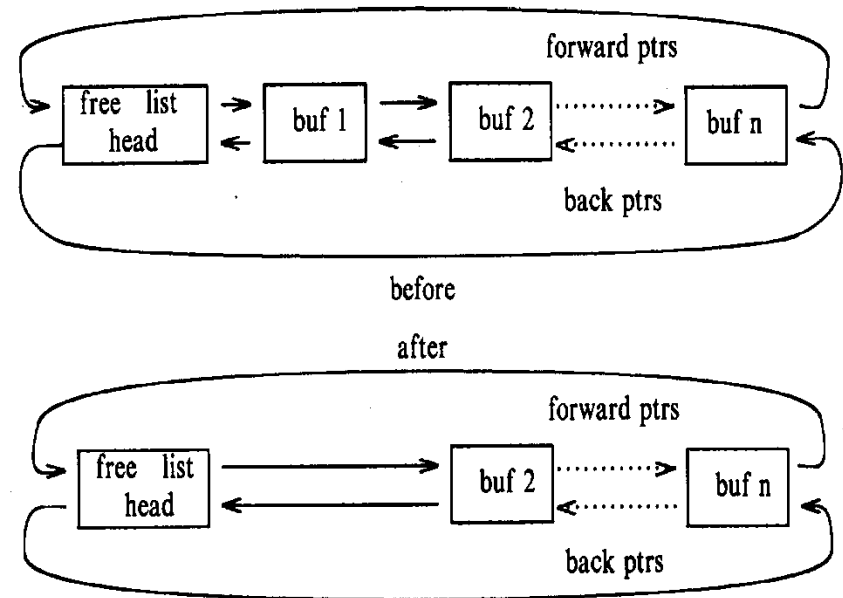
Estados de los buffers

- ✓ Free o disponible
- ✓ Busy o no disponible (en uso por algún proceso)
- ✓ El kernel está escribiendo a disco o leyendo del disco.
- ✓ Delayed write: buffers que hayan sido modificados en memoria, pero el bloque original en disco todavía no fué actualizado



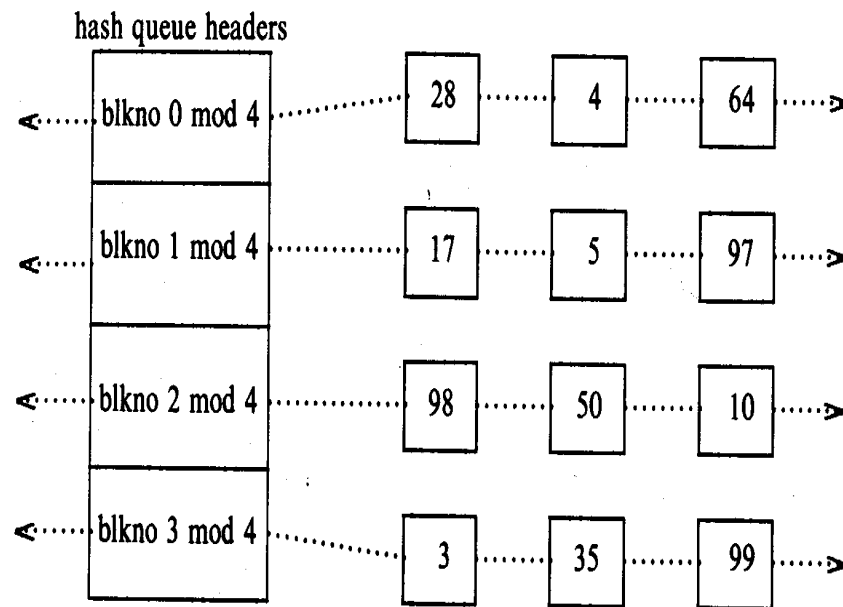
Free List

- ✓ Organiza los buffers disponibles, es decir, los buffers donde se puede cargar un nuevo bloque de disco.
- ✓ No necesariamente los bloques están vacíos
- ✓ Se ordena según LRU (least recent used)



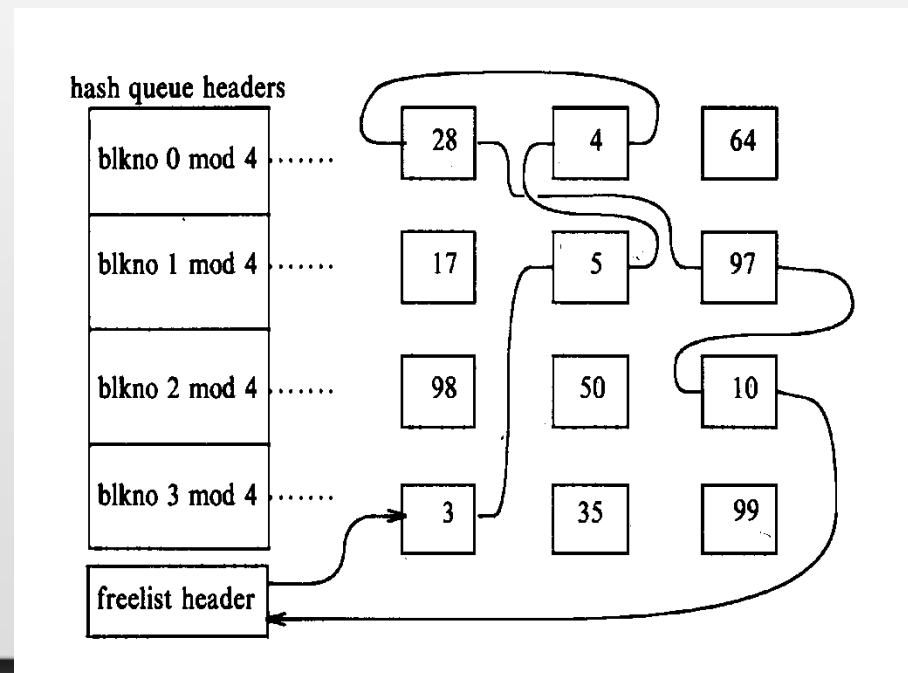
Hash Queues

- ✓ Son colas para optimizar la búsqueda de un buffer en particular
- ✓ Se organizan según una función de hash usando (dispositivo, #bloque)



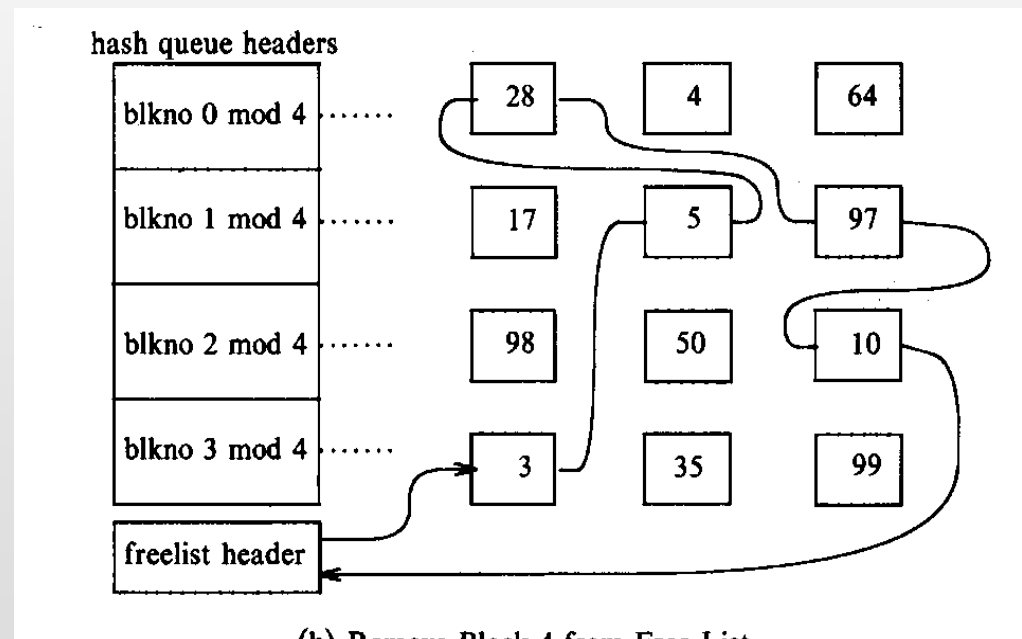
Búsqueda/recuperación de un buffer: 1er escenario

- ✓ El kernel encuentra el bloque en la hash queue .
- ✓ Está disponible (está en la free list).
- ✓ Ejemplo: busco el bloque 4



Búsqueda /recuperación de un buffer: 1er escenario

- ✓ Se remueve ese buffer de la free list
- ✓ Pasa a estado busy
- ✓ El proceso puede usar el bloque

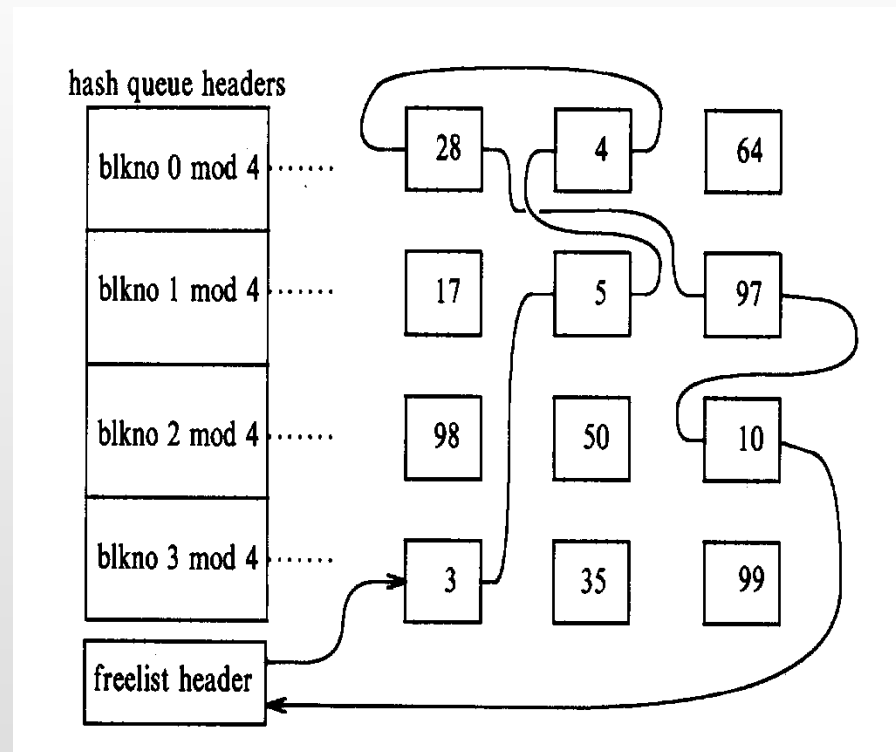


(b) Remove Block 4 from Free List



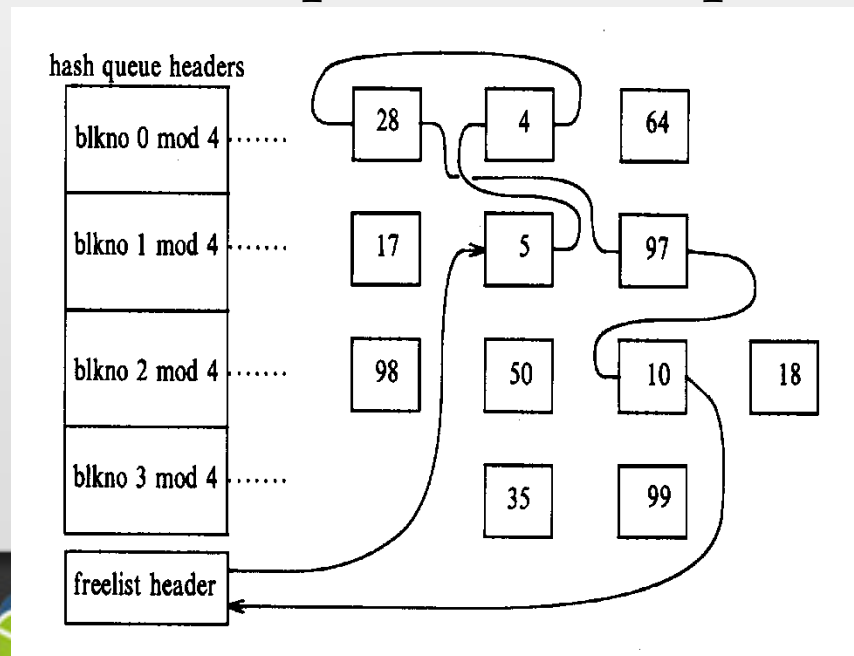
Búsqueda/recuperación de un buffer: 2do escenario

- ✓ El bloque buscado no está en la hash queue
- ✓ Ejemplo: el bloque 18



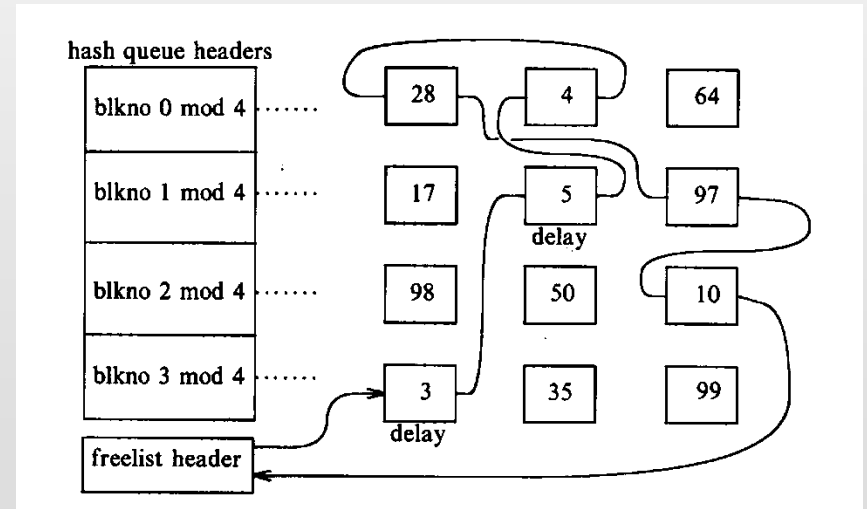
Búsqueda/recuperación de un buffer: 2do escenario

- ✓ Se toma un buffer de la free list (el 3)
- ✓ Siempre se usa el primero
- ✓ Se lee del disco el bloque deseado en el buffer obtenido
- ✓ Se ubica en la hash queue correspondiente



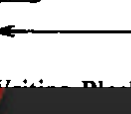
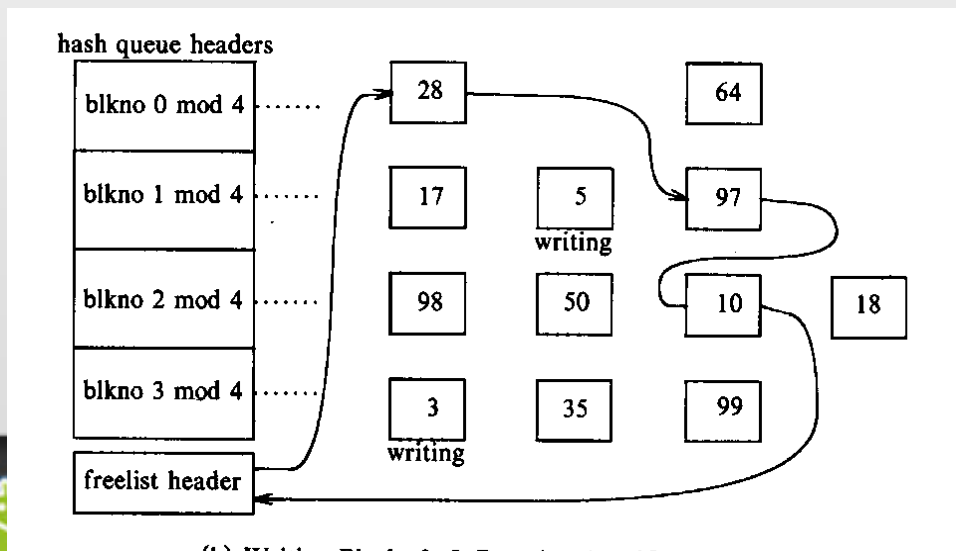
Búsqueda/recuperación de un buffer: 3er escenario

- ✓ El Kernel no encuentra el bloque buscado en la hash queue
- ✓ Debe tomar el 1ro de la free list, pero está marcado DW
- ✓ Ejemplo: busca el 18, toma el 3.



Búsqueda/recuperación de un buffer: 3er escenario

- ✓ El kernel debe mandar ese bloque a disco y tomar otro buffer de la free list
- ✓ Si también está DW, sigue hasta encontrar uno que no lo esté.
- ✓ Mientras manda a escribir a disco los DW, asigna el siguiente free
- ✓ Una vez escritos a disco los bloques DW, estos son ubicados al principio de la FreeList



Búsqueda/recuperación de un buffer:

4to escenario

- ☑ El kernel no encuentra el bloque en la hash queue y la free list está vacía.
- ☑ El proceso espera que se “libere” algun buffer
- ☑ Cuando el proceso despierta se debe verificar nuevamente que el bloque no este en la hash queue (algún proceso pudo haberlo cargado mientras éste dormía)



Búsqueda/recuperación de un buffer:

5to escenario

- ☑ El kernel busca un bloque y el buffer que lo contiene está marcado como busy
- ☑ El proceso queda en espera

