Interrupt Levels :

- Machine Errors
- Clock
- Disk
- Network Devices
- Terminal
- Software Interrupts

↑ Priority

Buffer Cache —:

× Improves : Response Time
Throughtput

× Minimize disk access frequency.

× Read operation :
- First kernel reads from buffer cache
- If data in cache, then no read from disk.
- Or a read from disk in case of miss & cache value for later.

× Delay - Write —: Delay writing of data updated in cache until replacement to secondary memory.

× Write-Through —: Update value in secondary memory on every update to value in cache.

Buffer Components —

Buffer
- Buffer Header (For Identification)
- Memory Array

Maintained as a linked list (pool of buffers)

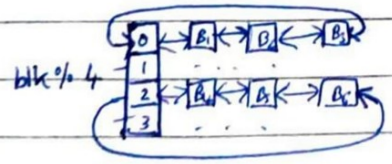
Buffer Header —:

device #
block #
status
ptr to data area
ptr to next hash queue value
ptr to prev. hash queue value
ptr to next free value
ptr to prev free value

Status —: Contains information about —
- × Buffer is currently locked
- × Buffer has valid data
- × Kernel must write data to disk before reassignment
- × Kernel is reading
- × Process is waiting for buffer to be free.

Structure of Buffer Pool —:

× Hashing using block number modulo size of hash table.



Free List —:

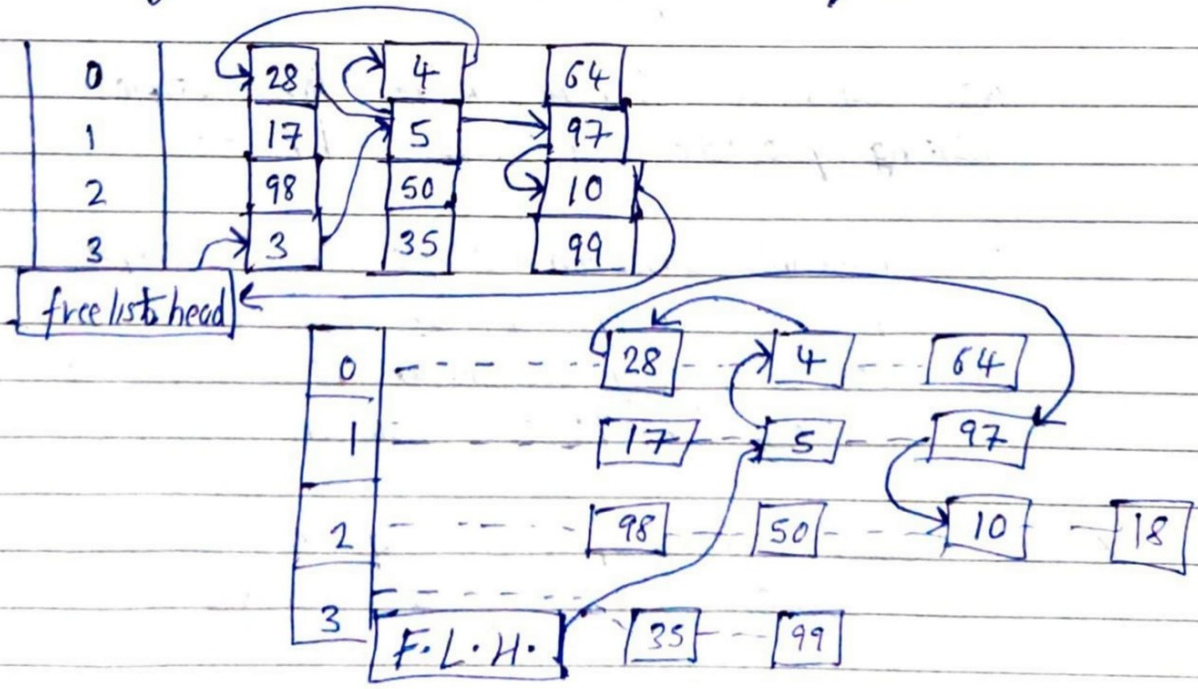
× Data cached as per LRU strategy:
× Maintains free list of buffers to preserve LRU order.
× Queue-style maintainance; insert in one end, delete from other end

Buffer Retrieval Scenarios —:

1) Kernel finds block on hash queue, its buffer is free.
2) Kernel unable to find block on hash queue, allocates buffer from free list
3) Kernel unable to find block on hash queue, allocates buffer marked delay-write from free list.
5) Kernel finds block on hash queue, its buffer is busy
4) Kernel unable to find block on hash queue, and no buffers in free list.

1) Block: free & on hash queue
   + Searches for buffer by device-block number & finds on hash queue & status for buffer is free.
   + Marked as non-free, lock, remove from free list.
   + One process allowed access, others put to sleep.

2) Block: not on hash queue, use from free list
   + Search hash queue
   + Use first buffer from free list if not marked as delayed-write & insert in hash queue.



3) Block: not on hash queue, use from free list, buffer is delay write
+ Search hash queue, fail search
+ Find first free node from free list.
+ If buffer is "delay write", attempt write of data to secondary memory, and allocate next non delay write buffer to process (async write)
+ After async write, add delay write buffers to start of free list (LRU buffers) & remove delay write status.

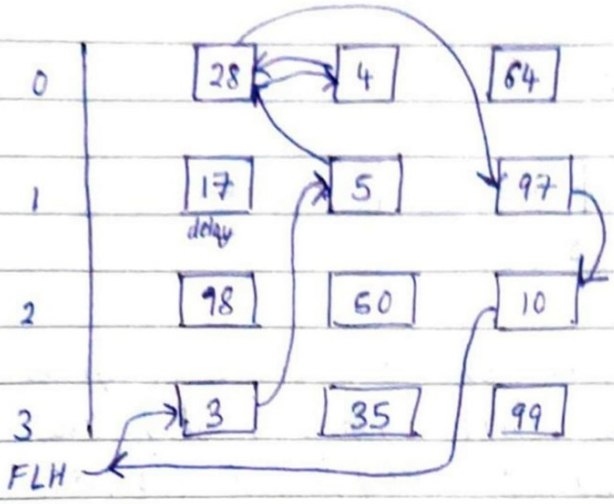
4) Block: not on hash queue, free list empty
+ Search hash queue, fail
+ No free buffer, so put process to sleep until free buffer is available (After brelse)

5) Block: busy, on hash queue
+ Search hash queue, found
+ Wait processes (put to sleep) until the buffer is added to free list (After brelse)

x Race Condition after new buffer for multiple waiting processes in scenario IV —:

1) Immediate allocation not possible for multiple processes competing for buffer.
2) Use a priority or similar resource sharing strategy for buffer allocation.

x Processes waking-up after sleep must search hash queue again to find consistent buffer



1) Search 17 on hash queue, found in queue 2
2) Block found & locked, so scenario 5, process waits till free.

Buffer Cache Advantages -:
1) Increase throughput, decrease response time
2) Allow uniform disk access
3) Reduce amount of disk traffic
4) Ensure file system integrity

Buffer Cache Disadvantages -:
1) More time to write data to disk
2) Loss of data (from delay-write buffers in case of crash)
3) Reduce extra data copy