FILE SYSTEMS
BLOCK MANAGEMENT

Module Number 4. Section 4
COP4600 – Operating Systems
Richard Newman

FILE SYSTEMS TOPICS

- Introduction
- Directories
- File Allocation
- Block Management
- File System Reliability
- File System Optimization

BLOCKS

- Effect of block size on performance
- Effect of block size on maximum file, partition size
- Keeping track of free blocks
  - Reliability/consistency
  - Allocation by zones
- Block cache
- Chunked reads

BLOCK SIZE

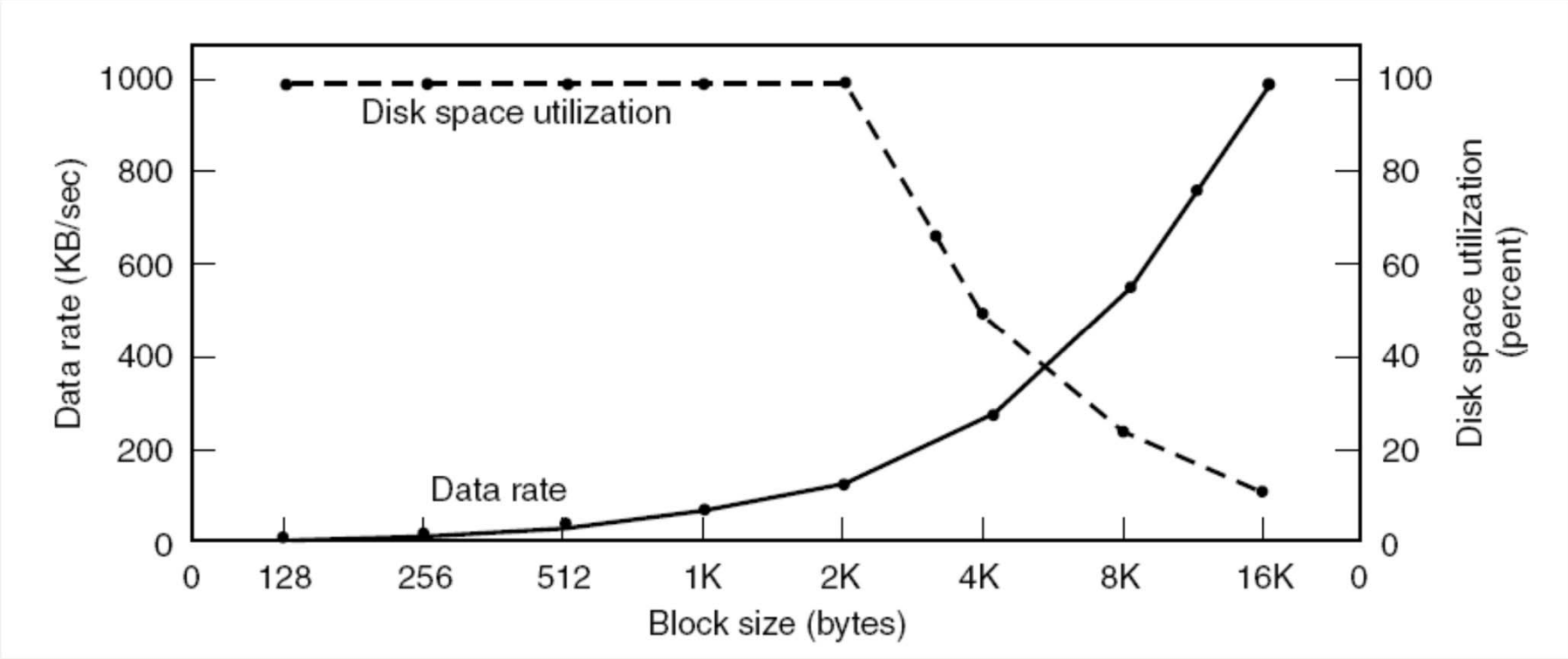


Figure 5-17. The solid curve (left-hand scale) gives the data rate of a disk. The dashed curve (right-hand scale) gives the disk space efficiency. All files are 2 KB.

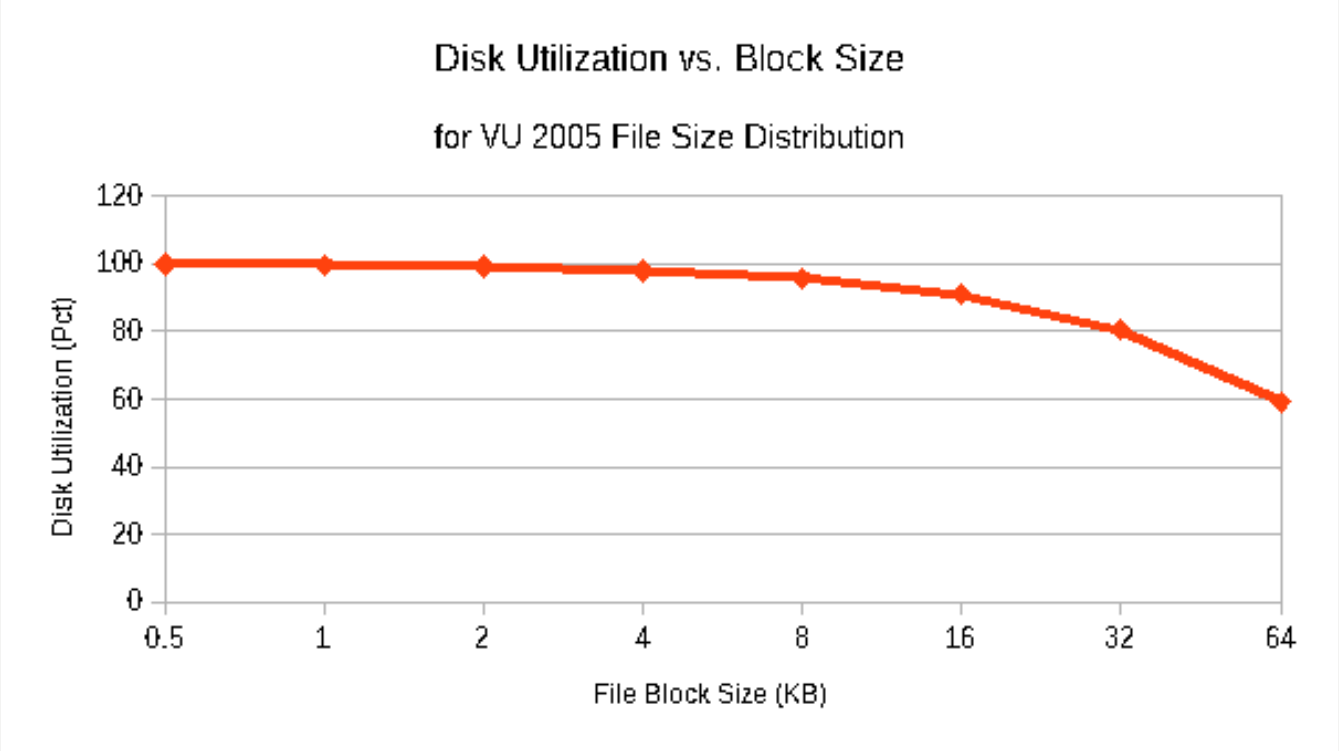
BLOCK SIZE (2)

Length	VU 1984	VU 2005	Web
1	1.79	1.38	6.67
2	1.88	1.53	7.67
4	2.01	1.65	8.33
8	2.31	1.80	11.30
16	3.32	2.15	11.46
32	5.13	3.15	12.33
64	8.71	4.98	26.10
128	14.73	8.03	28.49
256	23.09	13.29	32.10
512	34.44	20.62	39.94
1 KB	48.05	30.91	47.82
2 KB	60.87	46.09	59.44
4 KB	75.31	59.13	70.64
8 KB	84.97	69.96	79.69

Length	VU 1984	VU 2005	Web
16 KB	92.53	78.92	86.79
32 KB	97.21	85.87	91.65
64 KB	99.18	90.84	94.80
128 KB	99.84	93.73	96.93
256 KB	99.96	96.12	98.48
512 KB	100.00	97.73	98.99
1 MB	100.00	98.87	99.62
2 MB	100.00	99.44	99.80
4 MB	100.00	99.71	99.87
8 MB	100.00	99.86	99.94
16 MB	100.00	99.94	99.97
32 MB	100.00	99.97	99.99
64 MB	100.00	99.99	99.99
128 MB	100.00	99.99	100.00

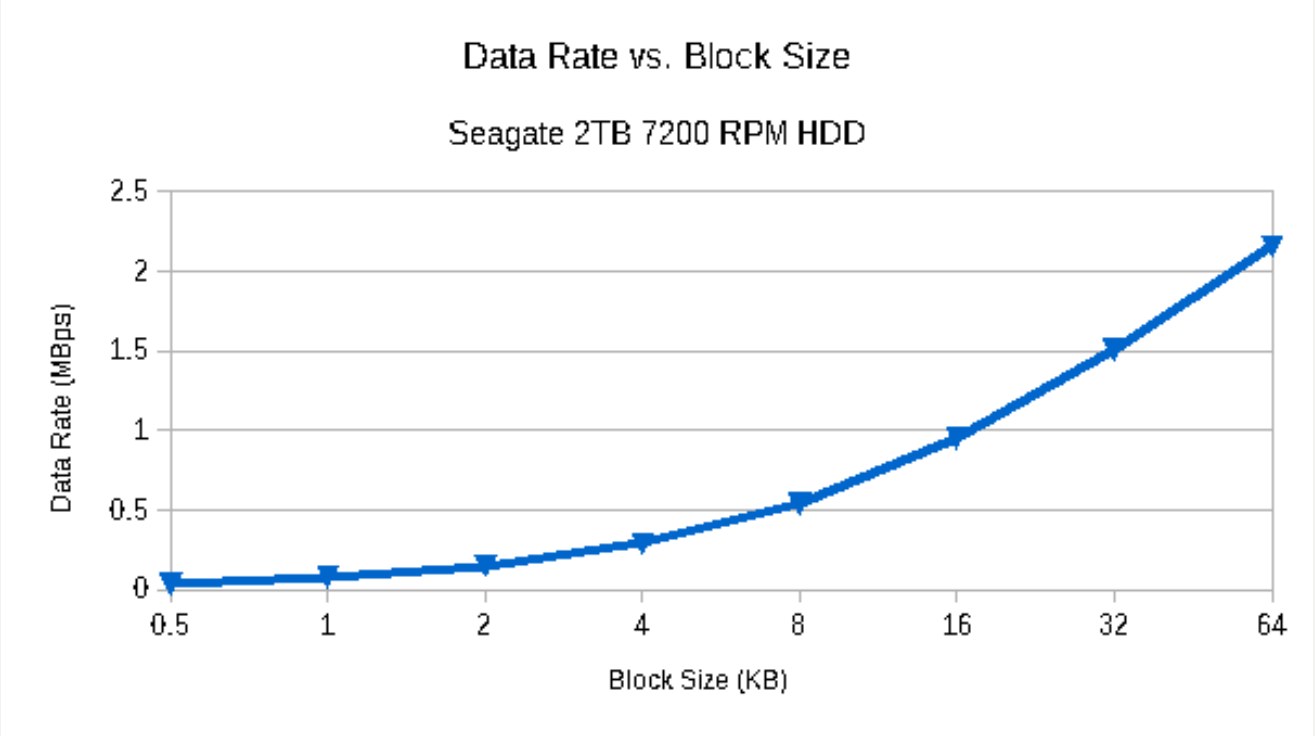
Percentage of files smaller than a given size

BLOCK SIZE (3)



Approximate disk space utilization (due to internal fragmentation) as function of block size for VU 2005 file size distribution

BLOCK SIZE (4)



Data rate (MBps) as function of block size for Seagate 2TB HDD (7200 RPM rotation speed, 63 sectors/track, 8.5 ms mean seek)

BLOCK SIZE (4)

2013 study (Welch & Noer):

- 25-90% of files are < 65KB in size
- But these only account for 3% on average (5-15% extreme) of total file space used….
- Average lengths ran from 75KB to over 16 GB for the installations studied, typical was a few MB average.
- Maximum file sizes ran 13 MB to 85 TB.
- They suggest using Solid State Drive (SSD) for small files and HDD for large files….

THE MS-DOS FILE SYSTEM

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Figure 4-31. Maximum partition size for different block sizes. The empty boxes represent forbidden combinations.

KEEPING TRACK OF FREE BLOCKS

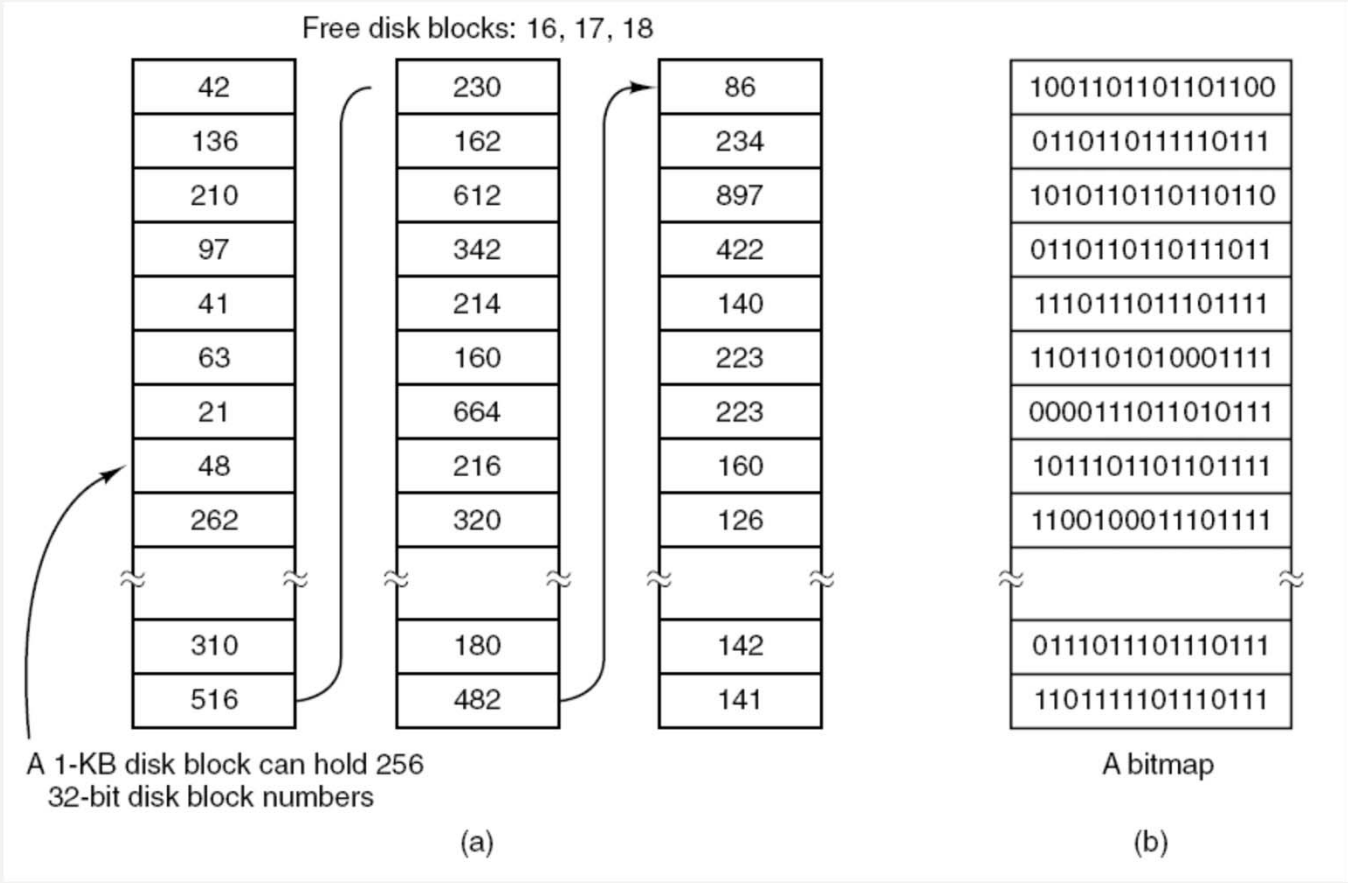
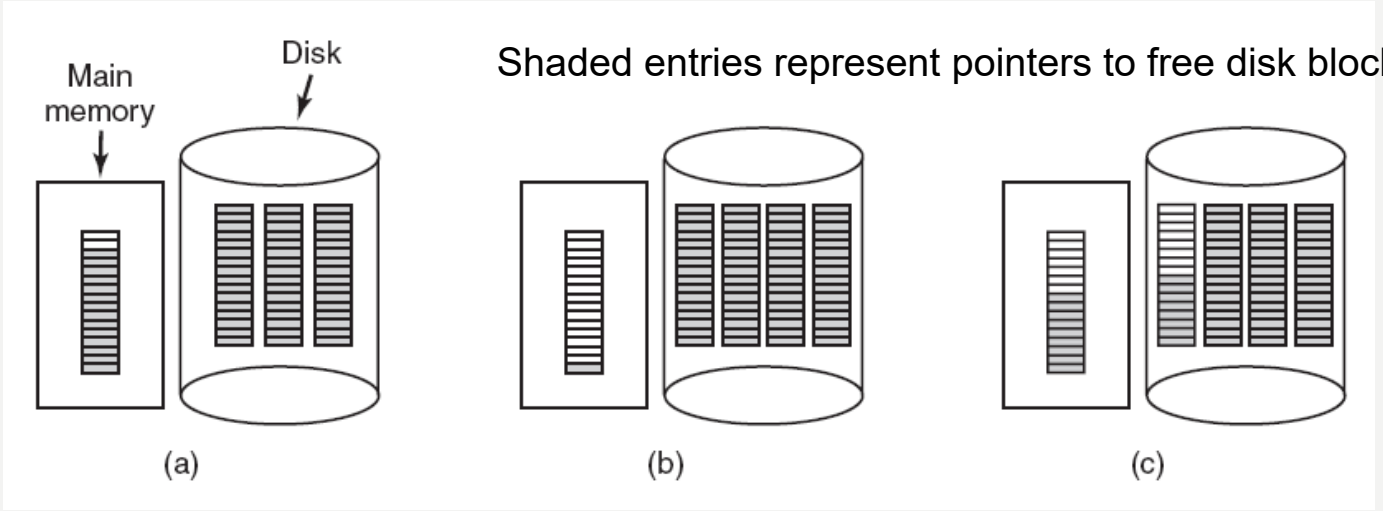


Figure 5-18. (a) Storing the free list on a linked list. (b) A bitmap.

KEEPING TRACK OF FREE BLOCKS (2)



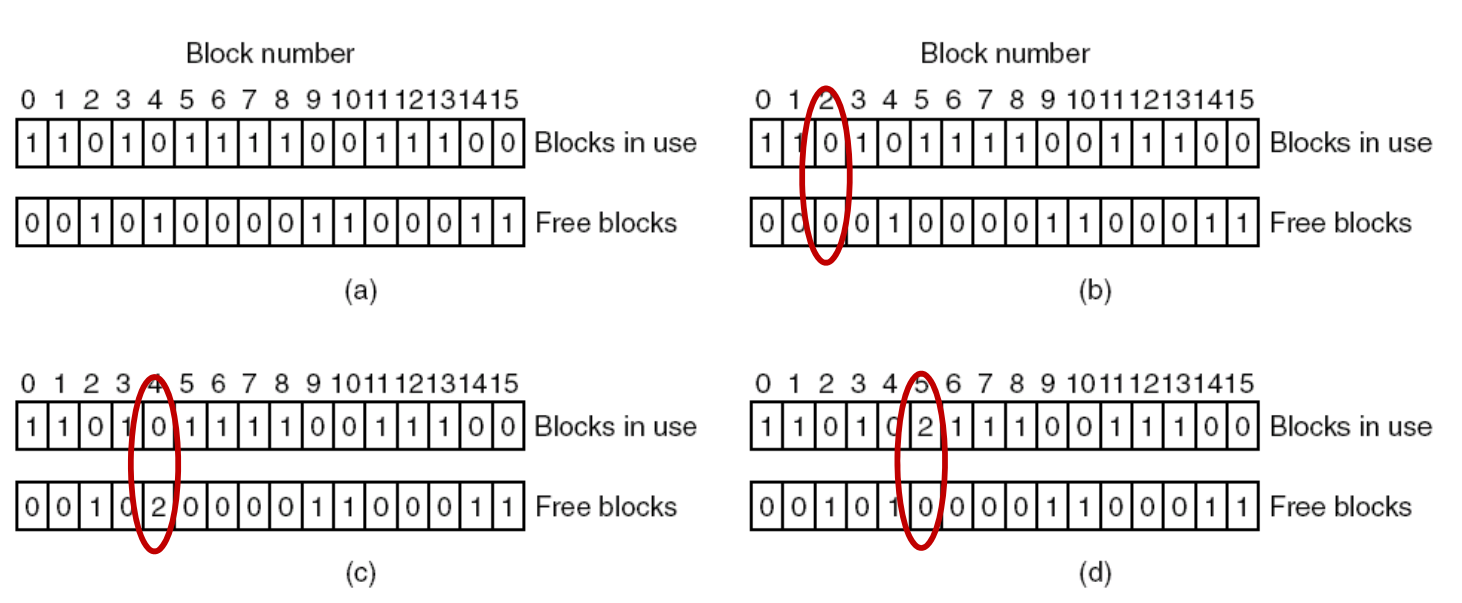
(a) An almost-full block of pointers to free disk blocks in memory and three blocks of pointers on disk.

(b) Result of freeing a three-block file. (c) An alternative strategy.

Why (c)?

Hint: Think about allocating a three-block file, then freeing it, repeatedly.

FILE SYSTEM CONSISTENCY



Can also have block listed as in-use and free! (c)<->(d)

Figure 5-19. File system states. (a) Consistent. (b) Missing block. (c) Duplicate block in free list. (d) Duplicate data block.

How to generate these lists?

How are (b) and (c) even possible? How to fix?

BLOCK MANAGEMENT

Procedure	Function
get_block	Fetch a block for reading or writing
put_block	Return a block previously requested with get_block
alloc_zone	Allocate a new zone (to make a file longer)
free_zone	Release a zone (when a file is removed)
rw_block	Transfer a block between disk and cache
invalidate	Purge all the cache blocks for some device
flushall	Flush all dirty blocks for one device
rw_scattered	Read or write scattered data from or to a device
rm_lru	Remove a block from its LRU chain

Figure 5-40. Procedures used for block management.

Blocks are used for disk access, but zones are used for allocation.

WRITING A FILE

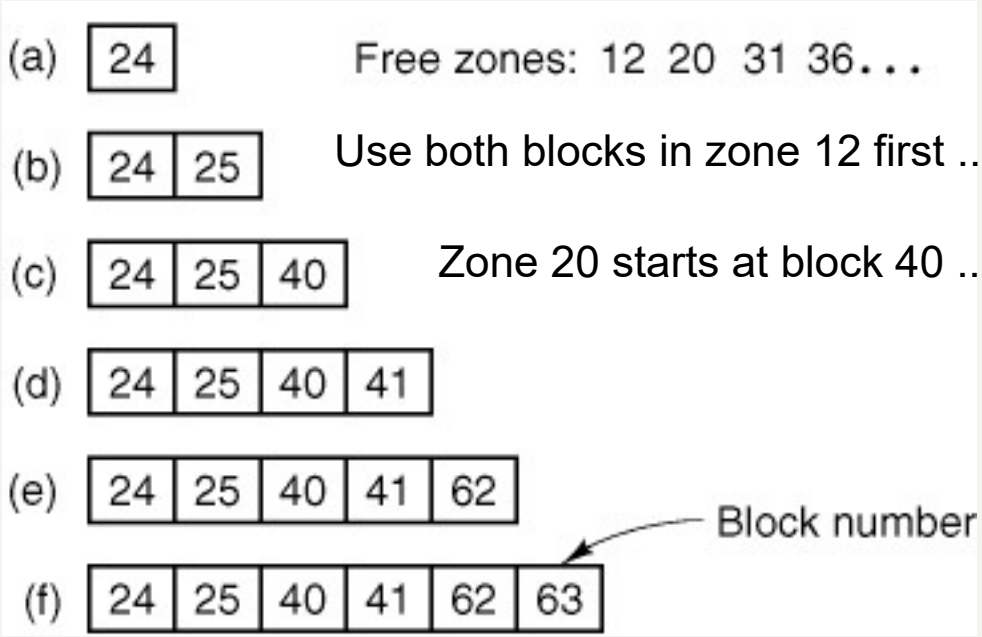


Figure 5-47. (a) – (f) The successive allocation of 1-KB blocks with a 2-KB zone.

THE BLOCK CACHE

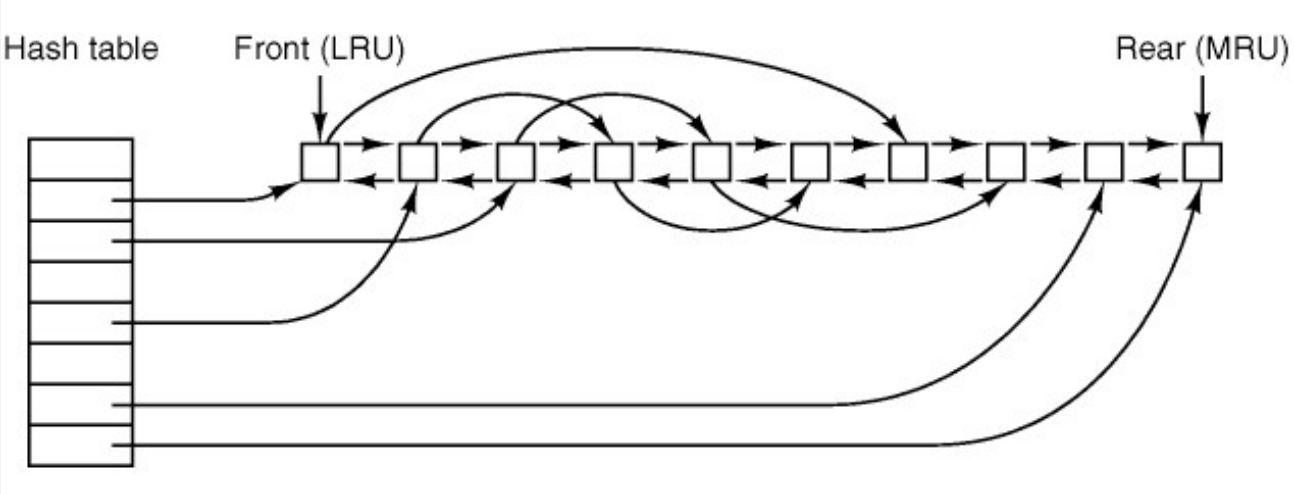


Figure 5-37. The linked lists used by the block cache.

BLOCK CACHE REPLACEMENT POLICY

- Some blocks rarely referenced two times within a short interval.
- Leads to a modified LRU scheme, taking two factors into account:
  1. Is the block likely to be needed again soon?
  2. Is the block essential to the consistency of the file system?

THE BLOCK CACHE

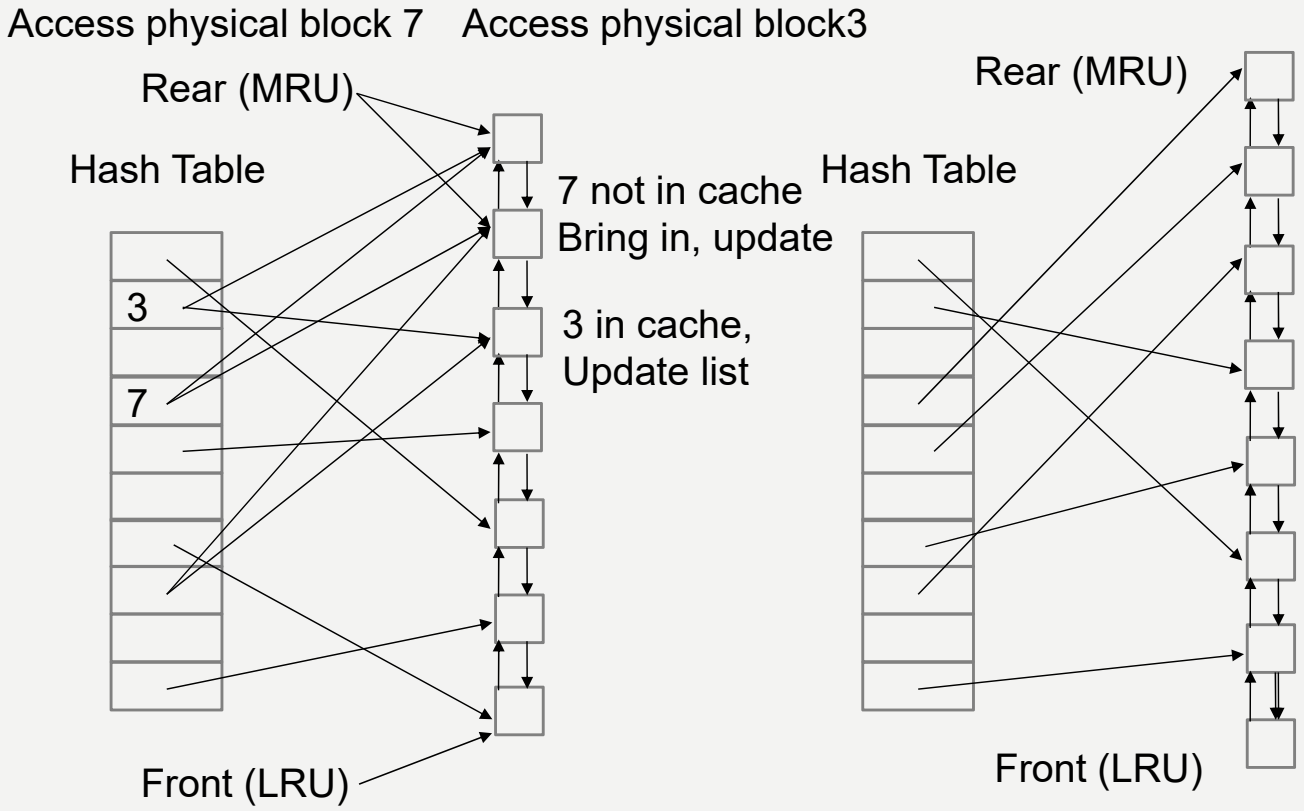


Figure 5-37. The linked lists used by the block cache.

INITIALIZATION OF THE FILE SYSTEM (1)

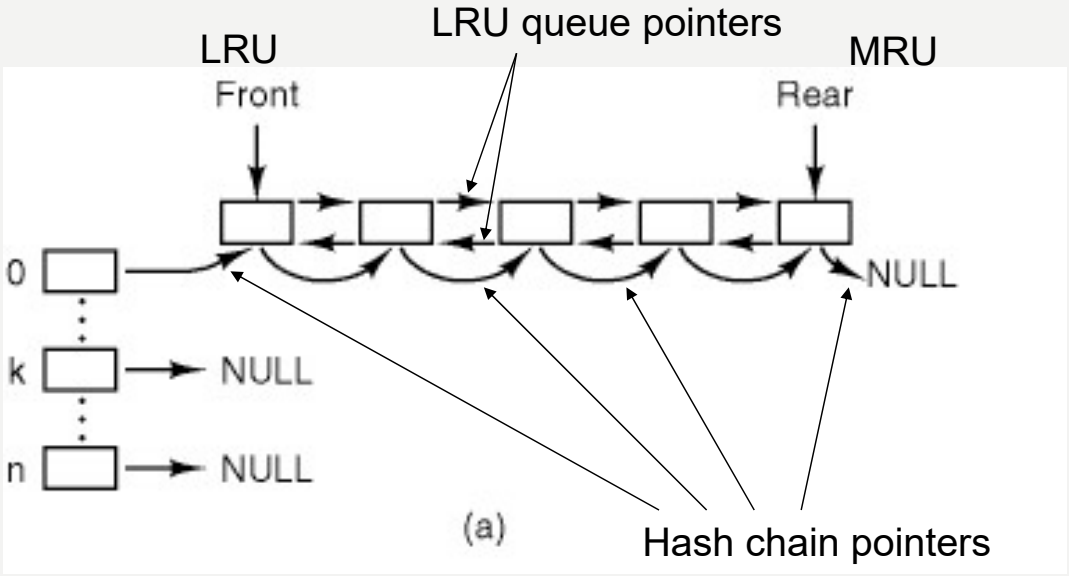


Figure 5-44. Block cache initialization.
(a) Before any buffers have been used.

INITIALIZATION OF THE FILE SYSTEM (2)

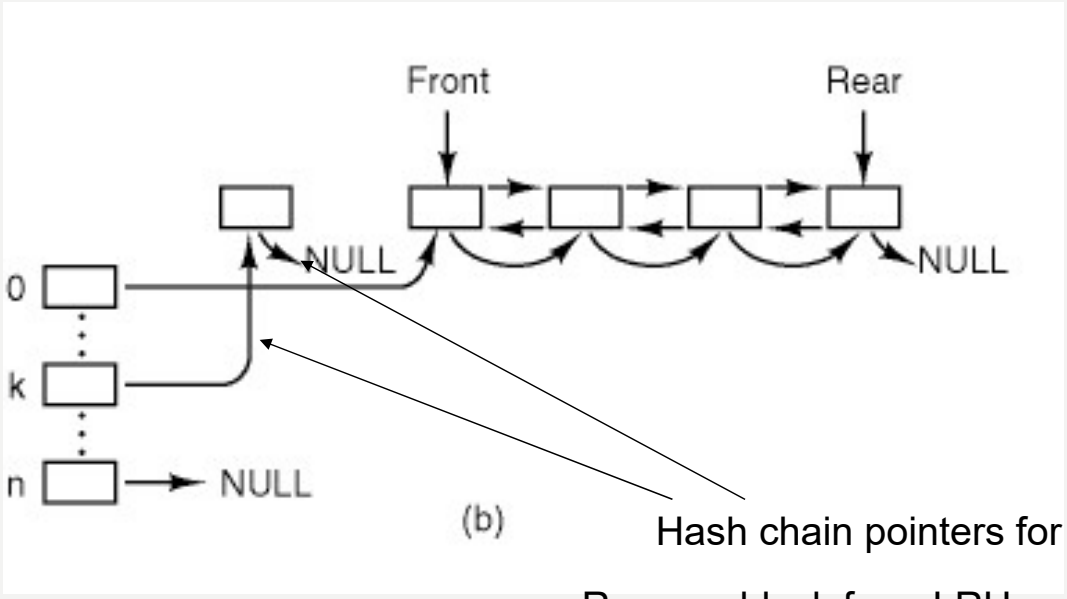


Figure 5-44. Block cache initialization.

(b) After one block (that hashes to k) has been requested.

INITIALIZATION OF THE FILE SYSTEM (3)

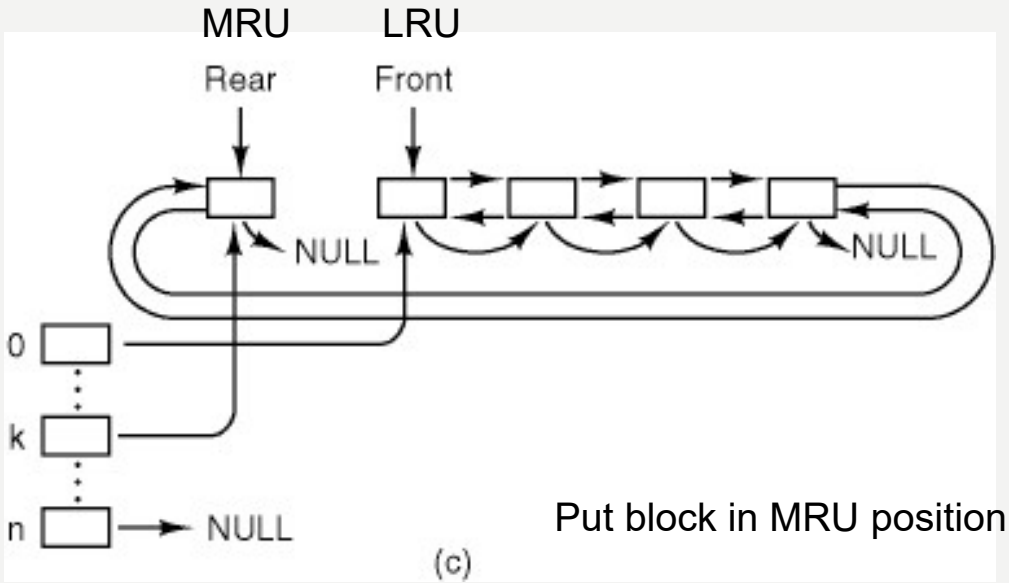


Figure 5-44. Block cache initialization.
(c) After the block has been released.

CHUNKED ACCESSES

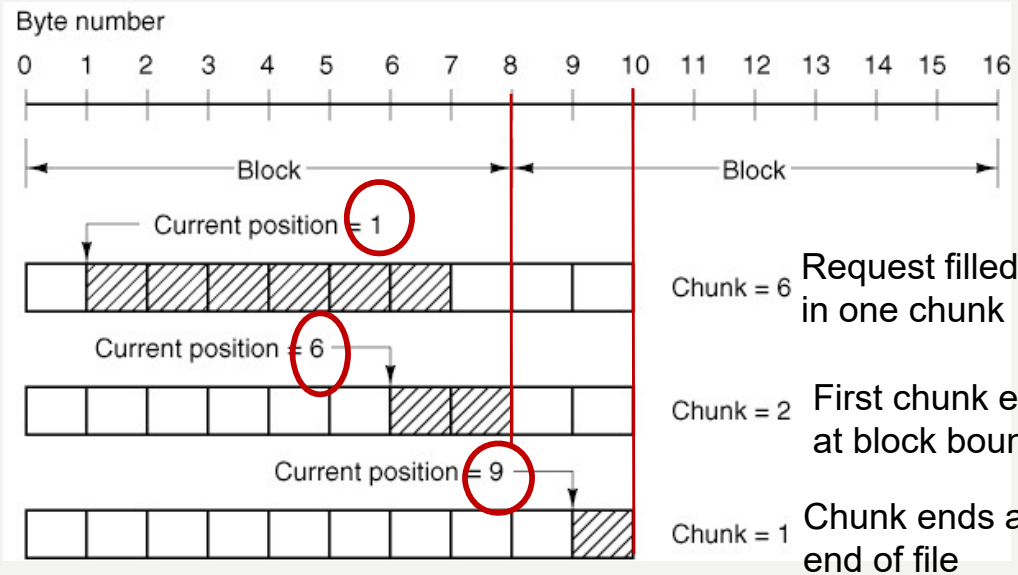


Figure 5-45. Three toy examples of how the first chunk size is determined for a 10-byte file. The block size is 8 bytes, and the number of bytes requested is 6. The chunk is shown shaded. A chunk will never exceed the current request, straddle a block boundary, or pass the end of the file. An operation on a chunk can be handled with one block access.

SUMMARY

- Tradeoffs of block size on performance
  - Bigger = faster transfer rate per byte transferred
  - Smaller = more precise, less waste
- Effect of block size on maximum file, partition size
- Keeping track of free blocks
  - Reliability/consistency
  - Allocation: zones vs. blocks vs. sectors
- Block cache
  - Another place where block size matters!
- Chunked access
  - Block boundary effects