Operating Systems

Jinghui Zhong （钟竞辉）
Office：B3-515
Email ： jinghuizhong@scut.edu.cn

File System Backup

- Backups are made to handle: recover from disaster or stupidity.
- Considerations of backups
  - ✓Entire or part of the file system
  - ✓**Incremental dumps**: dump only files that have changed
  - ✓Compression
  - ✓Backup an active file system
  - ✓Security

File System Backup

- **Two strategies for dumping a disk:**

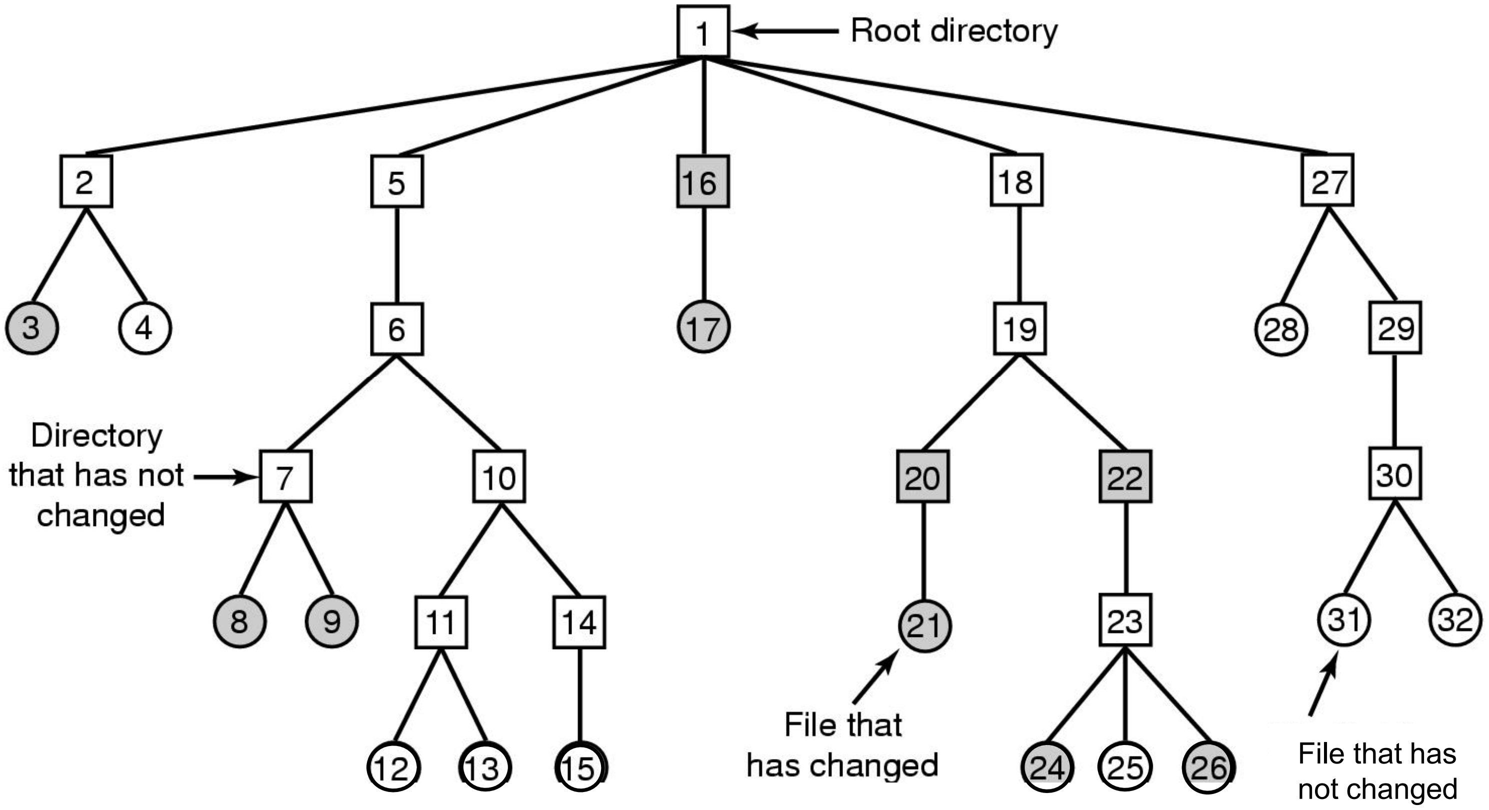
  ① **Physical dump**: starts at block 0 to the last one.

  Advantages: simple and fast

  Disadvantages: backup everything

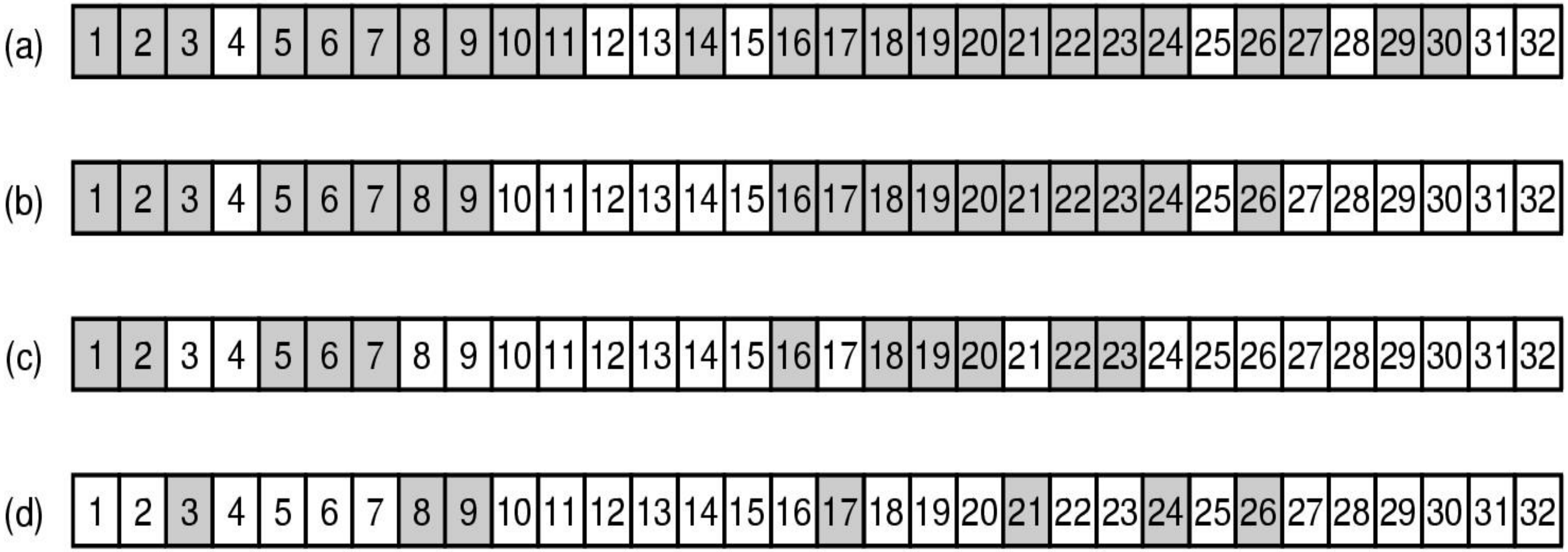
  ② **Logical dump**: starts at one or more specified directories and recursively dumps all files and directories found that have changed since some given base date.

File System Backup



A file system to be dumped (squares are directories, circles are files, shaded items are modified since last dump, each directory & file labeled by i-node number

File System Backup



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
(a)	▨	▨	▨		▨	▨	▨	▨	▨	▨	▨			▨		▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨		▨	▨		
(b)	▨	▨	▨		▨	▨	▨	▨	▨							▨	▨	▨	▨	▨	▨	▨	▨	▨		▨						
(c)	▨	▨			▨	▨	▨									▨		▨	▨	▨		▨	▨									
(d)			▨					▨	▨								▨				▨			▨		▨						

Bit maps used by the logical dumping algorithm

(After 4 phases, the dump is complete)

File System Consistency

- Most OS have a utility program, called a file system checker, to test the consistency of a file system.

E.g., *fsck* in UNIX, *sfc* in Windows

- Two types of consistency checks can be made:

  (a) block consistency

  (b) file consistency

File System Consistency

- **Block consistency:**
  ① Build two tables with a counter per block, initially set to 0 , The counters in the first table keep track of number of times each block is present in a file. The counters of the second table record the number of times in free list,
  ② Then, the program reads all the i-nodes and uses the i-nodes to build a list of all blocks used in the files (incrementing file counter as each block is read).
  ③ Check free list or bit map to find all blocks not in use (increment free list counter for each block in free list).

File System Consistency

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocks in use	1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Free blocks	0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1

(a)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocks in use	1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Free blocks	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1

(b)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocks in use	1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Free blocks	0	0	1	0	2	0	0	0	0	1	1	0	0	0	1	1

(c)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocks in use	1	1	0	1	0	2	1	1	1	0	0	1	1	1	0	0
Free blocks	0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1

(d)

- File system states
  - (a) consistent
  - (b) missing block – add it to the free list
  - (c) duplicate block in free list – rebuild the free list
  - (d) duplicate data block – copy the block to a free block

File System Consistency

- For checking directories – keep a list of counters per file starting at the root directory, recursively inspect each directory. For each file, increment the counter for the files i-node
- Compare computed value with link count stored in each i-node.
  - ✓ i-node link count > computed value = number of directory entries.

    Even if all files are removed, the i-node link count > 0. So the i-node will not be removed.

    Solution : set i-node link count = value computed
  - ✓ i-node link count < computed value

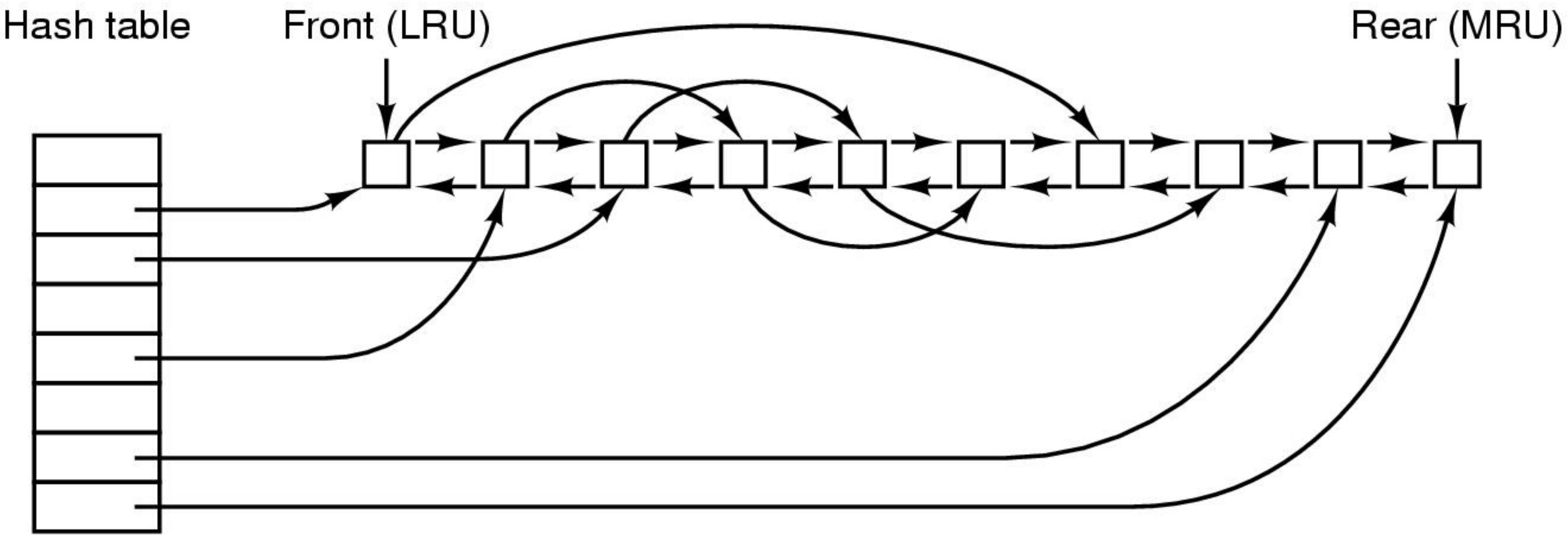
    The i-node may be freed even when there is another directory points to it

    directory will be pointing to unused i-node

    solution : set inode link count = computed value

File System Performance

- A **block cache** or **buffer cache** is a collection of blocks that logically belong on the disk, but are kept in memory to improve performance.
- All of the previous paging replacement algorithms can be used to determine which block should be written when a new block is needed and the cache is full.



File System Performance

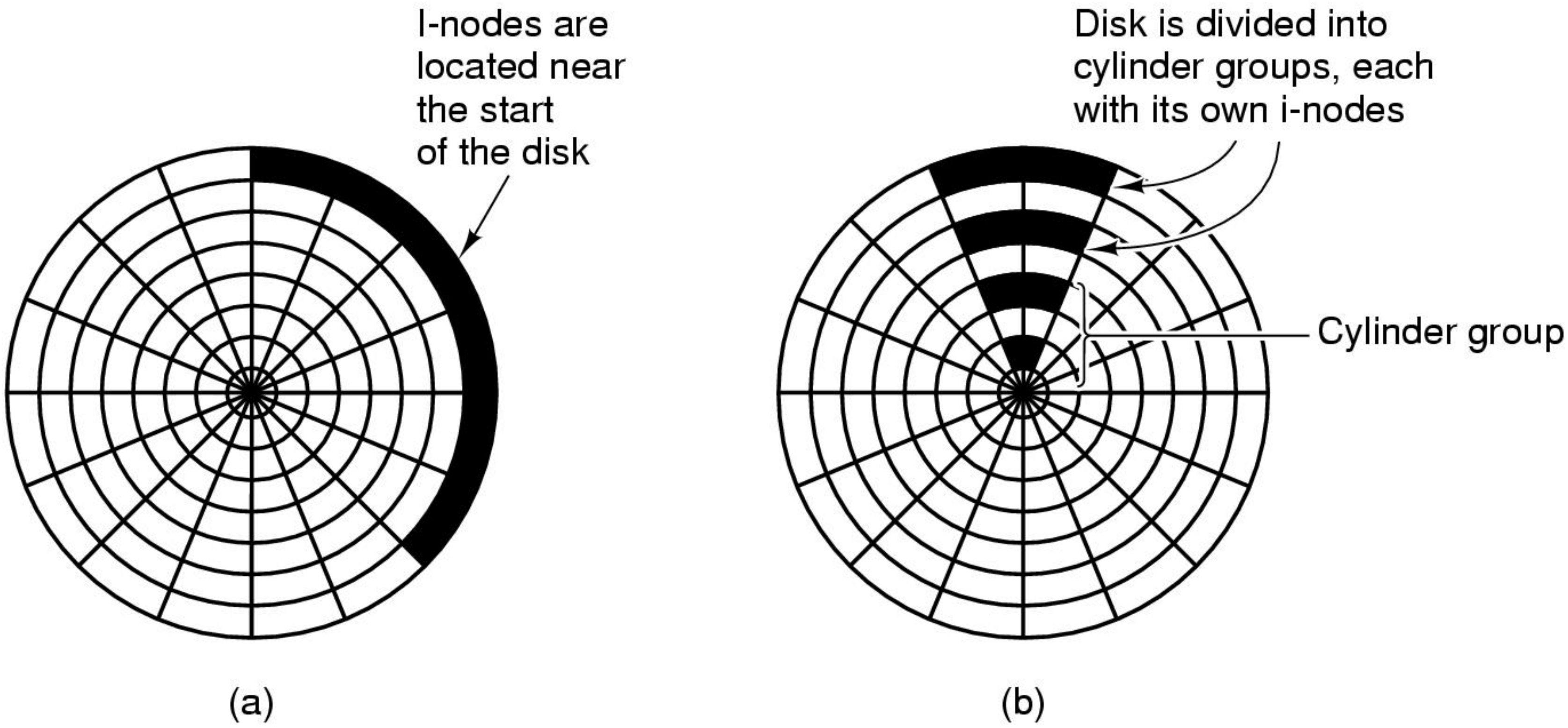
- Periodically, all data block should be written out (e.g. write works all day).
- UNIX - system call sync forces modified blocks out to the disk immediately. e.g. update runs in background during sync every 30 seconds
- MS-DOS - **write-through cache** => all modified blocks are written immediately.

e.g. write a 1K block one character at a time

UNIX collect them together

MS-DOS 1 at a time

File System Performance


I-nodes are located near the start of the disk
Disk is divided into cylinder groups, each with its own i-nodes
Cylinder group
(a)
(b)

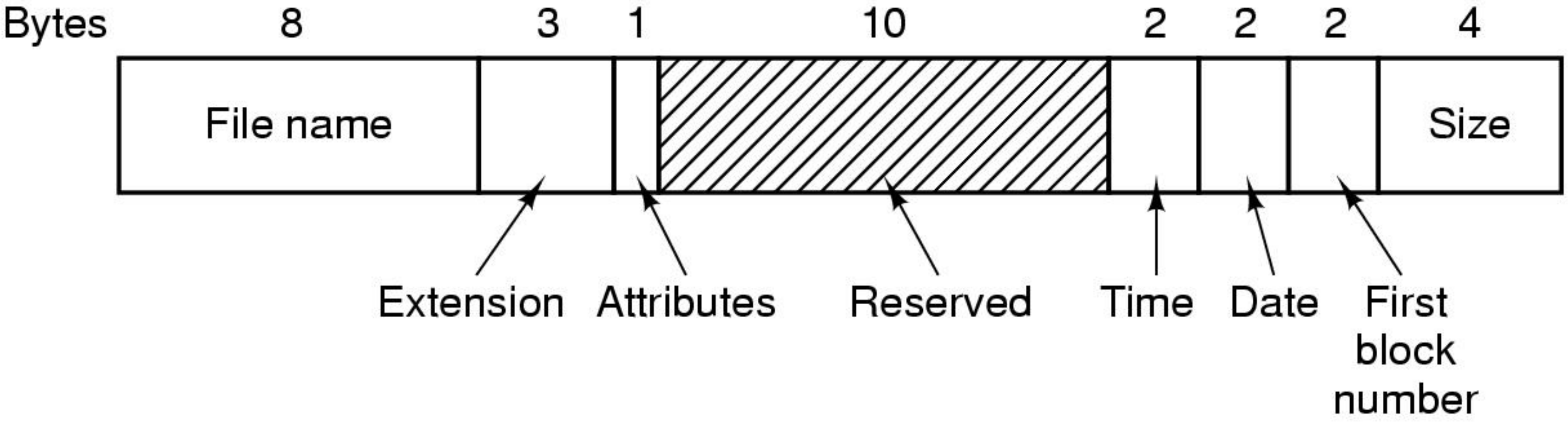
●Reading a block needs one access for the i-node and one for the block. Save i-node access time.

(a) I-nodes placed at the start of the disk

(b) Disk divided into cylinder groups, each with its own blocks and i-nodes.

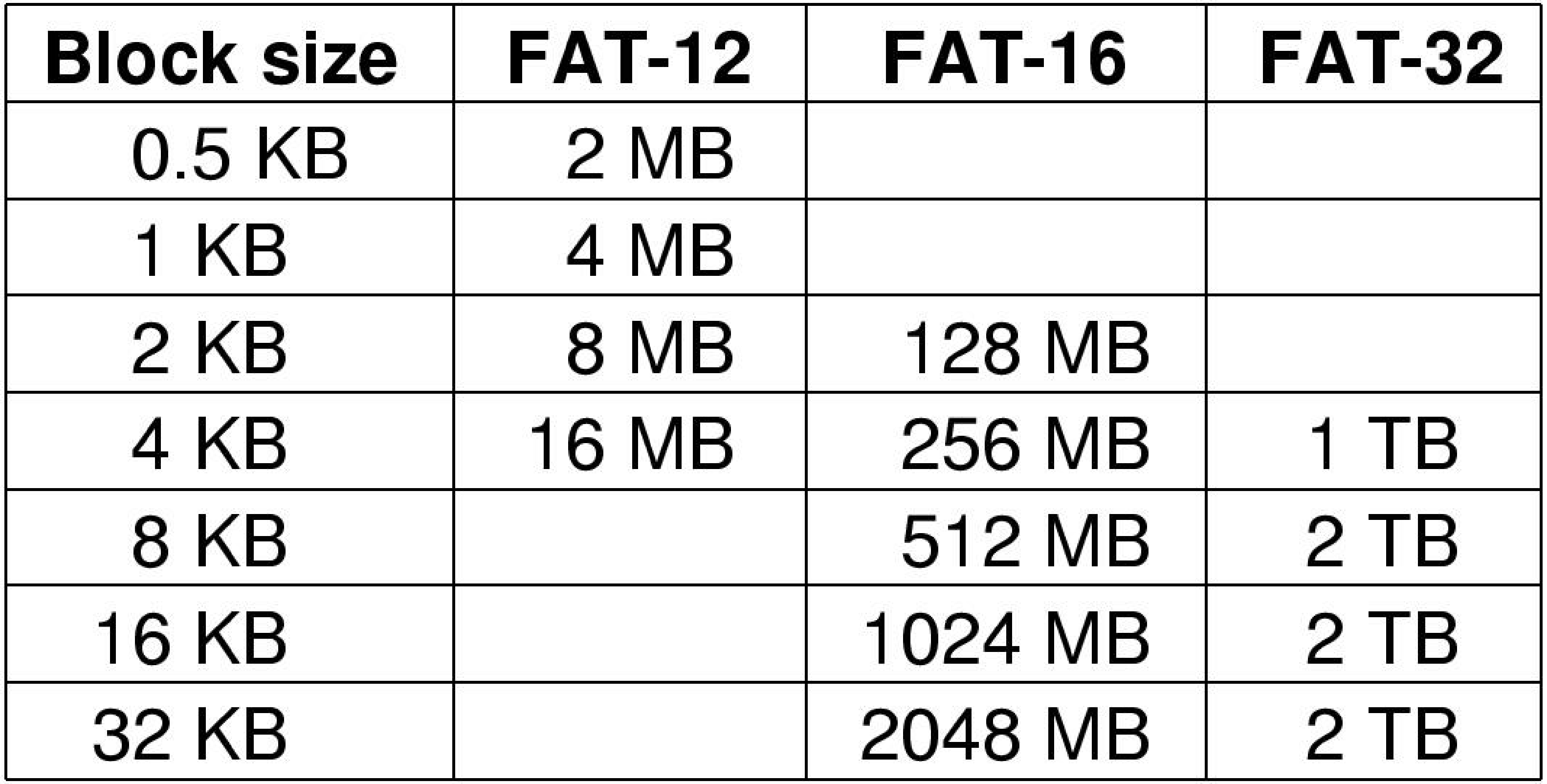
The MS-DOS File System

- Many digital cameras and MP3 players use it.
- Use a fixed-size 32 byte directory entry.



The MS-DOS directory entry

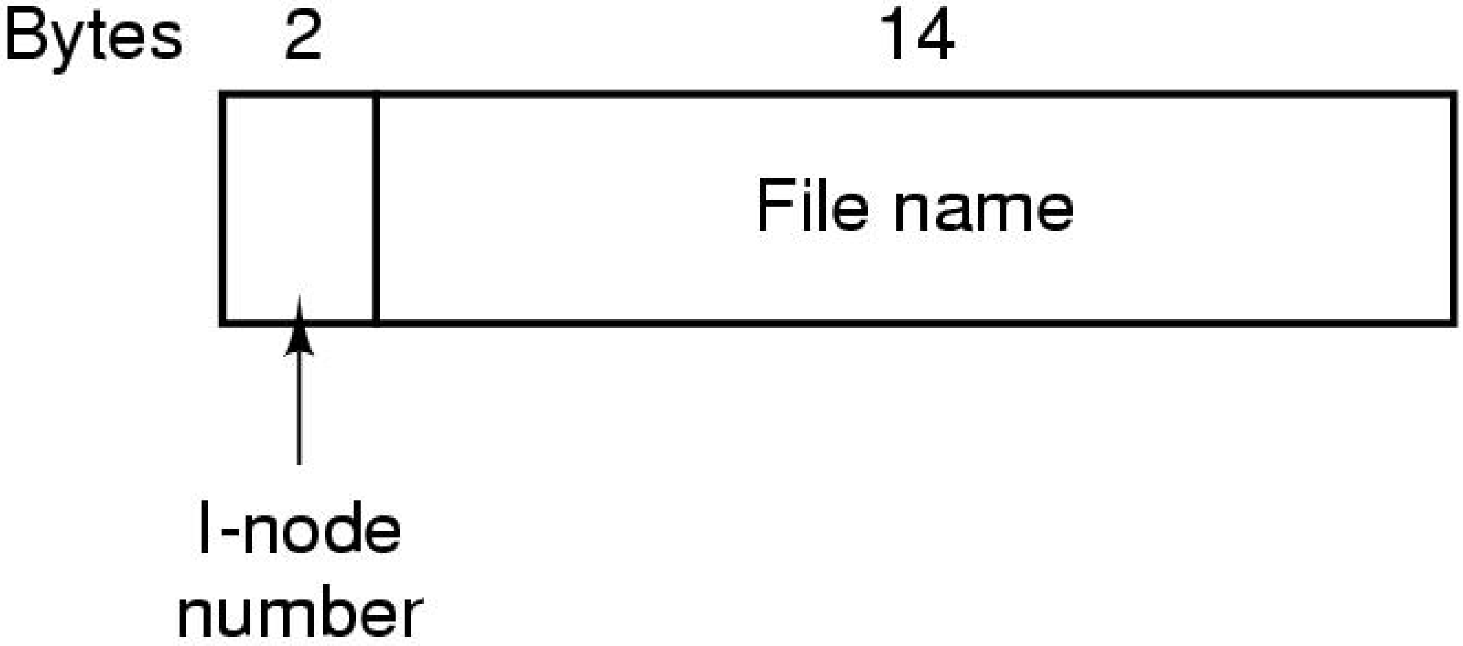
The MS-DOS File System



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

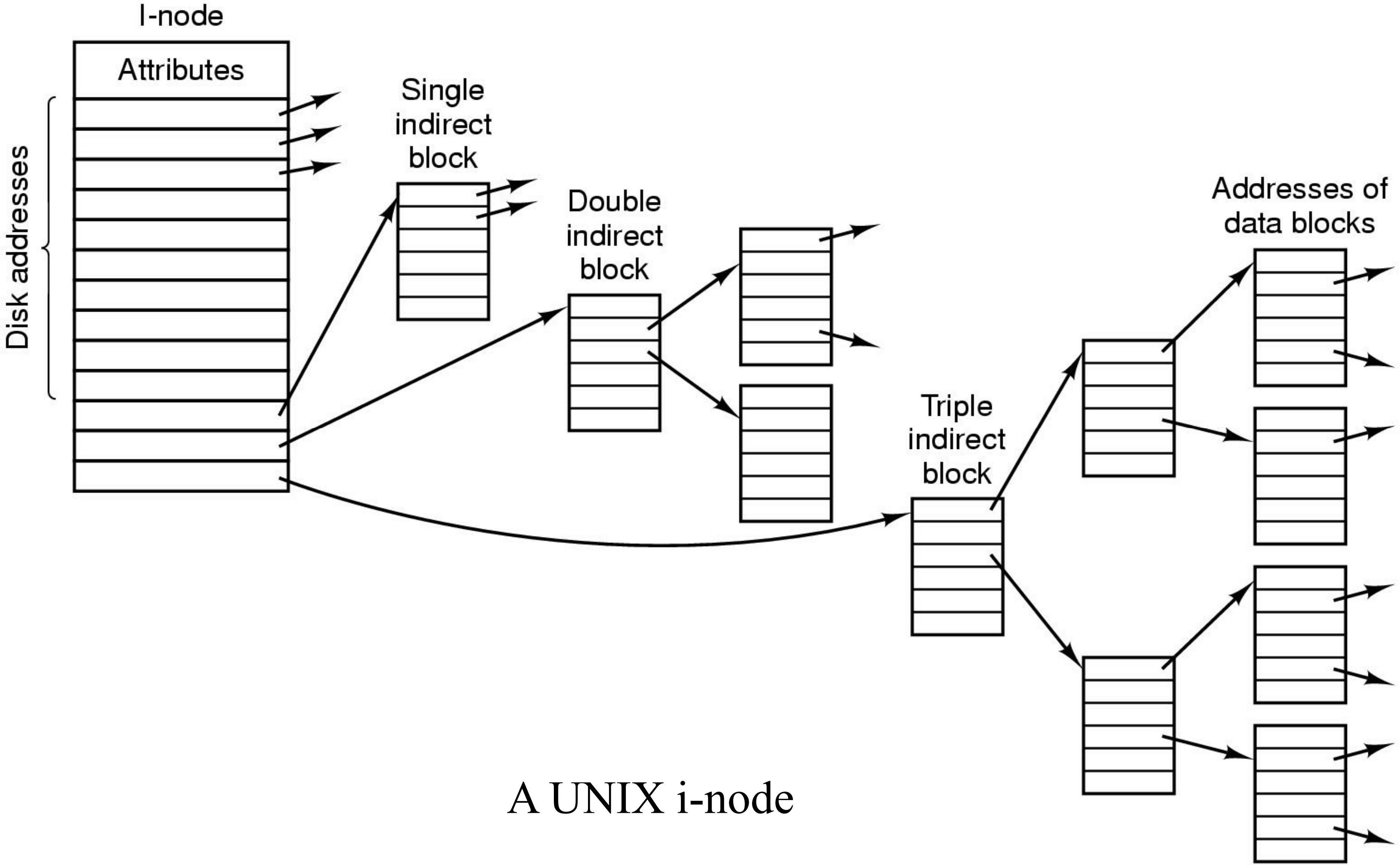
- Maximum partition for different block sizes
- The empty boxes represent forbidden combinations

The UNIX V7 File System



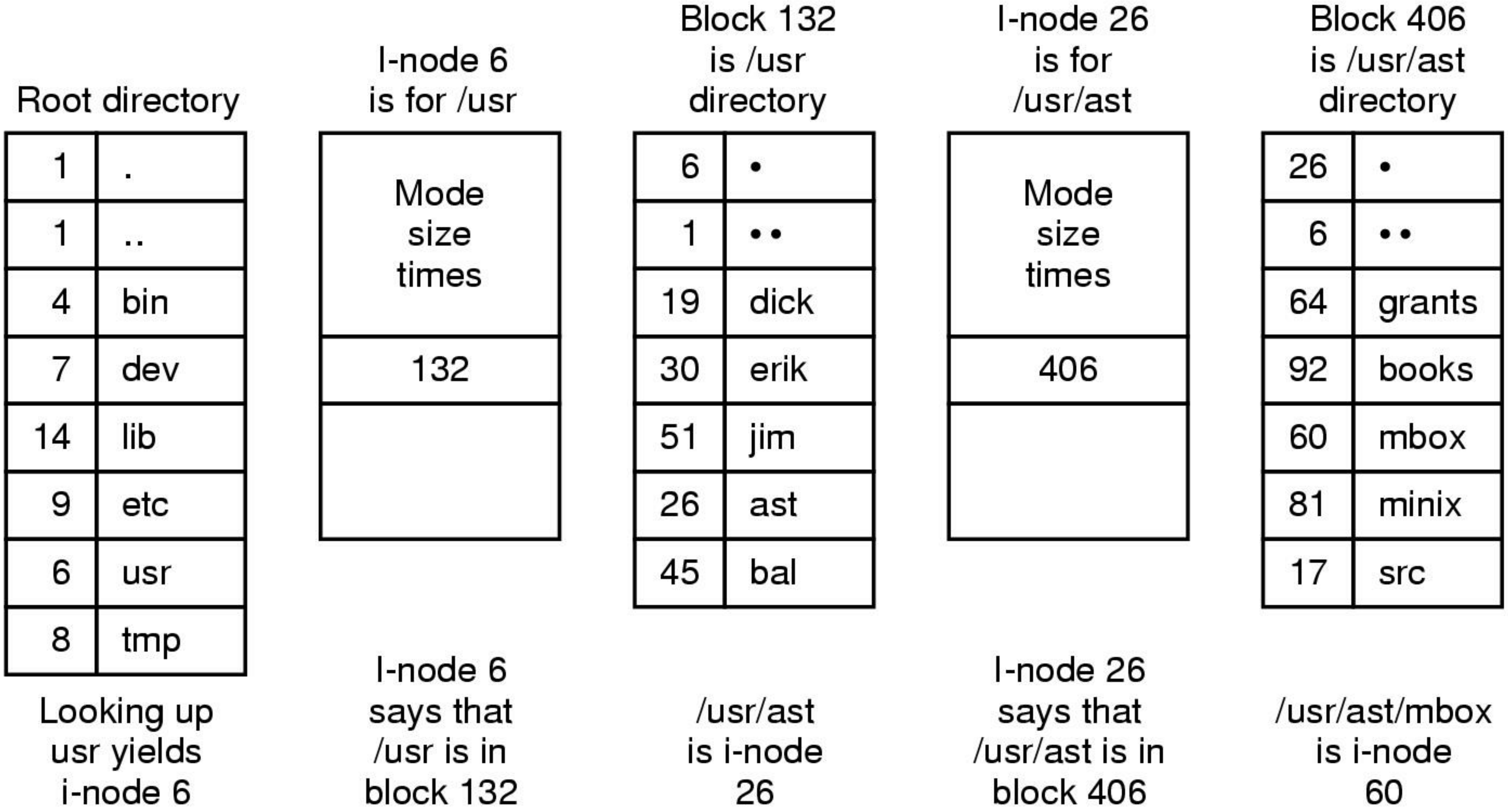
A UNIX V7 directory entry

The UNIX V7 File System



A UNIX i-node

The UNIX V7 File System



Root directory

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

Looking up usr yields i-node 6

I-node 6 is for /usr

Mode size times
132

I-node 6 says that /usr is in block 132

Block 132 is /usr directory

6	•
1	••
19	dick
30	erik
51	jim
26	ast
45	bal

/usr/ast is i-node 26

I-node 26 is for /usr/ast

Mode size times
406

I-node 26 says that /usr/ast is in block 406

Block 406 is /usr/ast directory

26	•
6	••
64	grants
92	books
60	mbox
81	minix
17	src

/usr/ast/mbox is i-node 60

The steps in looking up */usr/ast/mbox*

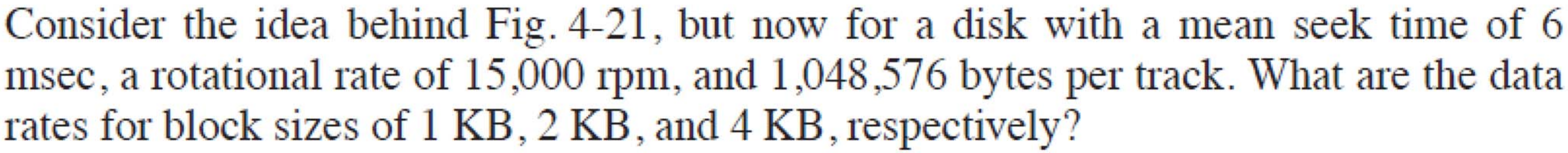
Check Points

① Please describe the two strategies for dumping a disk.

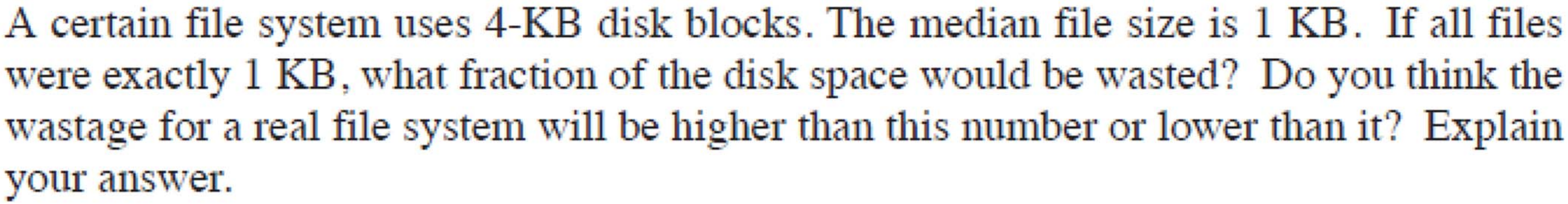
② Please describe the two types of consistency checks .

③ Please describe two methods to increase file-system performance.

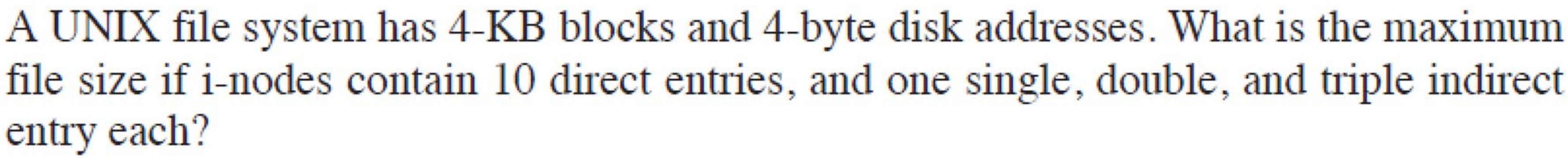
Check Points



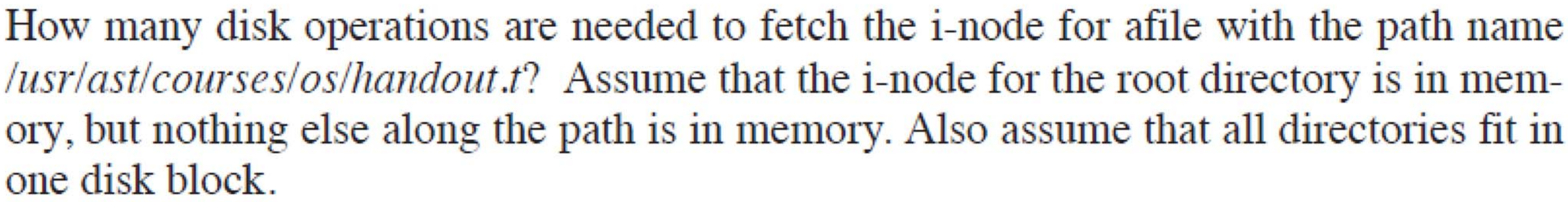
Consider the idea behind Fig. 4-21, but now for a disk with a mean seek time of 6 msec, a rotational rate of 15,000 rpm, and 1,048,576 bytes per track. What are the data rates for block sizes of 1 KB, 2 KB, and 4 KB, respectively?



A certain file system uses 4-KB disk blocks. The median file size is 1 KB. If all files were exactly 1 KB, what fraction of the disk space would be wasted? Do you think the wastage for a real file system will be higher than this number or lower than it? Explain your answer.



A UNIX file system has 4-KB blocks and 4-byte disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and triple indirect entry each?



How many disk operations are needed to fetch the i-node for afile with the path name */usr/ast/courses/os/handout.t*? Assume that the i-node for the root directory is in memory, but nothing else along the path is in memory. Also assume that all directories fit in one disk block.