Preview

- Disk Space Management
- Free Block Management
  - Linked List
  - Bit Map
- Disk Quota
- File System Backup
  - Physical Backup
  - Logical Backup

Disk-Space Management

- Two General Strategies for storing an n byte file.
  1. n consecutive bytes of disk space are allocated.
  2. The file is split up into a number of fixed size blocks. And also a disk space is also divided into same size block (similar with paging in memory management). Then, a file is saved in disk with number of blocks.
- Second idea is common in Computer System. Block size is one of issue need to be decided.
  - Big size block – internal fragmentation
  - Small size block – need multiple seek and rotation time for read

Disk-Space Management

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

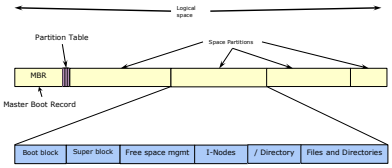
Length	VU 1984	VU 2005	Web
16 KB	92.53	78.92	86.79
32 KB	97.21	85.87	91.65
64 KB	99.18	90.84	94.80
128 KB	99.84	93.73	96.93
256 KB	99.96	96.12	98.48
512 KB	100.00	97.73	98.99
1 MB	100.00	98.87	99.62
2 MB	100.00	99.44	99.80
4 MB	100.00	99.71	99.87
8 MB	100.00	99.86	99.94
16 MB	100.00	99.94	99.97
32 MB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

Table: Percentage of files smaller than a given size (in bytes).
From file size distribution (%) research by Tanenbaum (2006)

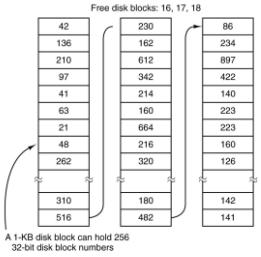
Free Block Management

- System reserve spaces (blocks) to save free block information.
  - Linked-List – free block information is saved in the blocks
  - Bitmap – system keep Bitmap for saving free block information (1 for used block 0 for available block)

Free Block Management



Free Block Management (Linked List)



Free Block Management
(Linked List)

- Each block for free blocks list holding as many free disk block numbers as will fit.

Ex)

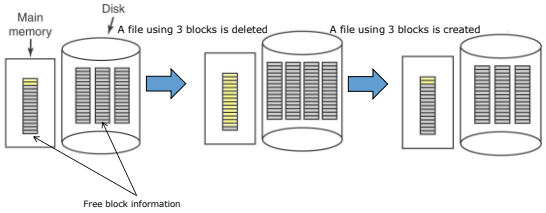
- With a 1 KB block size and a 32 bit disk block number.
- A block can hold 8×2^{10} / 32 = 256 -1 = 255 free blocks numbers (1 slot for next free block pointer)!!!
- 16 GB disk has $2^{34}/2^{10} = 2^{24}$ blocks
- Needs reserve $2^{24}/255$ = 65794 blocks to hold free block numbers

Free Block Management
(Linked List Implementation)

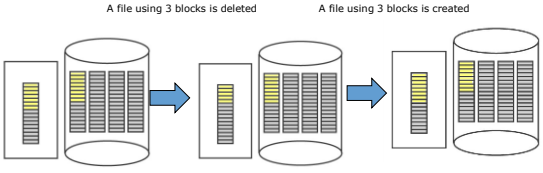
- When the free list method is used, only one block of pointers need be kept in main memory.
- When a file is created, the needed block blocks are taken from the block of pointers.
- When it runs out, a new block of pointer is read in from the disk.
- When a file is deleted, its blocks are freed and added to the block of pointer in the main memory.

Free Block Management
(Linked List Implementation)

Under certain circumstances, linked method leads to unnecessary disk I/O

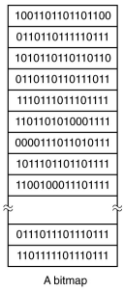


Free Block Management
(Linked List Implementation)



Free Block Management
(Bit Map)

A disk with n block requires a bitmap with n bits. a free blocks are represented by 1s in the map, allocated blocks by 0 (or vice versa).



A bitmap

Free Block Management
(Bit Map)

Ex)Same example with Linked-List

- 16 GB disk
- Size of block = 1KB
- There are $2^{34}/2^{10} = 2^{24}$ blocks
- Need 2^{24} bit (for bit map) = 2^{24} / 2^3 Byte = $2^{11} \times 2^{10}$ Byte = 2^{11} KB = 2048 blocks for the bitmap.

Free Block Management

Ex)Linked List vs. Bit map

- Linked List
  - With a 4 KB block size and a 64 bit disk block number.
  - A block can hold $8 \times 4 \times 2^{10}$ / 64 = 512 -1 = 511 free blocks numbers (1 slot for next free block pointer)!!!
  - 2 TB disk has 2×2^{40}/ $4 \times 2^{10} = 2^{41} / 2^{12} = 2^{29}$ blocks
  - 2^{29}/511 = 1050628.007. Needs reserve 1050629 blocks to hold free block numbers
- Bit map
  - 2 TB disk has 2×2^{40}/ $4 \times 2^{10} = 2^{41} / 2^{12} = 2^{29}$ blocks
  - Size of bit map = 2^{29} bit = $2^{29} / 2^3 = 2^{26}$ byte = 2^{16} KB
  - 2^{16} KB / 4 KB = 2^{14} = 16384 blocks

Free Block Management

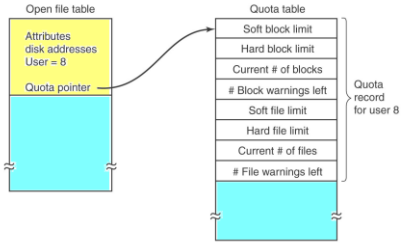
In the file system of an operating system, two methods are widely used to keep track of free blocks: a linked list and a bitmap. Let's say a block size is 8-KB and 32-bit disk block number in a file system.

- How many maximum blocks are needed for keep track 128-GB disk with linked list?
  - Size of Each block = $8 \times 8 \times 2^{10}$ bits = 2^{16} bits
  - One block can keep = size of block/size of a block address = 2^{16} bits / 32 bits = $2^{16} / 2^5 = 2^{11}$ -1 = 2047 block information
  - Total # of blocks in the disk = size of disk / block size = 128GB / 8KB blocks = $128 \times 2^{30} / 8 \times 2^{10}$ = $2^7 \times 2^{30} / 2^3 \times 2^{10} = 2^{37} / 2^{13} = 2^{24}$ blocks
  - # of blocks need to keep track of free blocks = 2^{24} blocks /2047 = 8196.002
  - ∴ 8197 blocks
- How many blocks are needed for keep track of 128-GB disk with bitmap?
  - Total # of blocks in the disk == 2^{24} blocks
  - Need 2^{24} bits for bit map= $2^{24} / 8 = 2^{24} / 2^3 = 2^{21}$ Byte
  - # of blocks need for bitmap = $2^{21} / (8 \times 2^{10}) = 2^{21} / 2^{13}) = 2^8$ blocks =256 blocks

Disk Quota

- The system administrator assigns each user a maximum allotment of files and blocks.
- The operating system makes sure that each user do not exceed their quotas
- When user opens a file, the attributes and disk addresses are located and put into an **open file table** in main memory.
- The **quota table** contains the quota record for every user with a currently open file.
- This record is an extract from a quota file on disk for the users whose files are currently open.
- When all the files are closed, the record is written back to the quota file

Disk Quota



File System Backup

File Backup System design issues

1. Should the entire file system be backed up or only part of it?
2. It is wasteful to back up files that have not changed since the last backup
3. Since big amount of data are typically dumped, it may be desirable to compress the data before writing them to tape or disk.
4. It is difficult to perform a backup on an active file system – making rapid snapshots of the file system.
5. Backup introduces many nontechnical problems into an organization – security problem

File System Backup

(Physical Dump)

There are two strategies for dumping a disk to tape or disk: **Physical** and **Logical Dump**

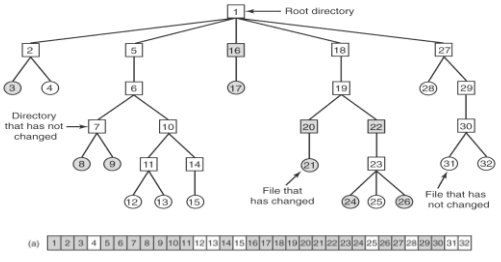
- **Physical Dump** – start at block 0 of the disk (or SSD), writes all the disk blocks onto the output tape or disk in order, and stop when it has copied the last one.
  - How to handle skipping unused blocks?
  - How to handle bad block?
- Advantage : Simplicity, speed
- Disadvantage: disable to skip selected directories, make incremental dumps and restore individual files.

File System Backup
(Logical Dump: Unix)

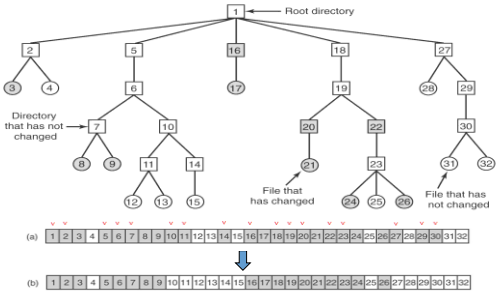
- **Logical Dump** – starts at one or more specified directories and recursively dumps all files and directories found there that have changed since some given base date

File System Backup
(Logical Dump: Unix)

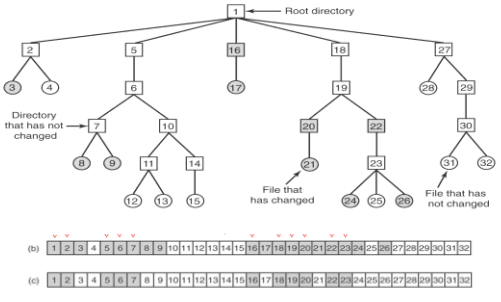
- The dump algorithm maintains a bitmap indexed by i-node number with several bits per i-node The algorithm operates in four phases.(Phase 1 & 2 : prepare dump)
- Phase 1
  - begins at the starting directory and examines all the entries in it. For each modified file, its i-node is marked in the bitmap. Each directory is also marked and recursively inspected.
- Phase 2
  - unmarking any directories that have no modified files or directories in them or under them.
- Phase 3
  - all marked directory is dumped
- Phase 4
  - all marked files is dumped



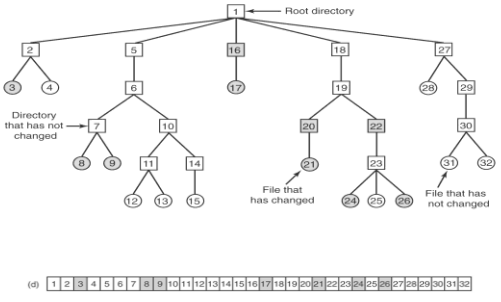
Phase 1:
Begins at the starting directory and examines all the entries in it. For each modified file, its i-node is marked in the bitmap. Each directory is also marked and recursively inspected



Phase 2: **unmarking** any directories that have no modified files or directories in them or under them.



Phase 3: all marked directory is dumped



Phase 4: all marked file is dumped