OPERATING SYSTEMS

File Management

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

File Management – Free space management

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all the PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

Free-Space Management

- Need to reuse the disk space from deleted files for new files
- To keep track of free disk space, the system maintains a **free-space list**
- The free-space list records all **free disk blocks**
- When a file is created, the free-space list is searched for the required amount of space and allocate that space to the new file. This space is then removed from the free-space list.
- When a file is deleted, its disk space is added to the free-space list

Bit Vector

- The free-space list is implemented as a **bit map** or **bit vector**.
- Each block is represented by 1 bit.
- If the block is free, the bit is 1; if the block is allocated, the bit is 0
- For example, consider a disk where blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18, 25, 26, and 27 are free and the rest of the blocks are allocated.
- The free-space bit map would be 00111100111111000110000011100000 ...
- The main advantage of this approach is its relative simplicity and its efficiency in finding the first free block or n consecutive free blocks on the disk

Bit Vector

- One technique for finding the first free block on a system that uses a bit-vector to allocate disk space is to sequentially check each word in the bit map to see whether that value is not 0, since a 0-valued word contains only 0 bits and represents a set of allocated blocks
- The first non-0 word is scanned for the first 1 bit, which is the location of the first free block.
- The calculation of the block number is

  (number of bits per word) × (number of 0-value words) + offset of first 1 bit

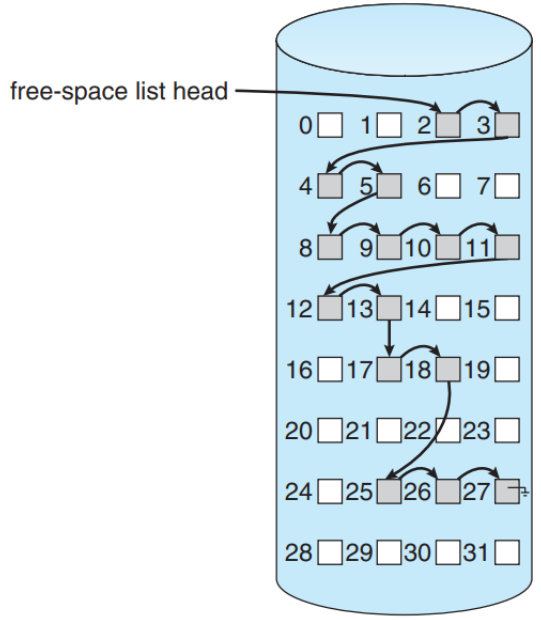
Bit Vector

- Bit vectors are inefficient unless the entire vector is kept in main memory
- Keeping the bit vector in main memory is possible for smaller disks but not necessarily for larger ones
- A 1.3-GB disk with 512-byte blocks would need a bit map of over 332 KB to track its free blocks
- A 1-TB disk with 4-KB blocks requires 32 MB to store its bit map.

OPERATING SYSTEMS

Linked List

- Link together all the free disk blocks, keeping a pointer to the first free block in a special location on the disk and caching it in memory.
- The first free block contains a pointer to the next free disk block, and so on
- **Example:** Consider blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18, 25, 26, and 27 are free and the rest of the blocks are allocated.
- The pointer would point to block 2 as the first free block. Block 2 would contain a pointer to block 3, which would point to block 4 and so on
- This scheme is not efficient; to traverse the list, one must read each block, which requires substantial I/O time.



Grouping

- A modification of the free-list approach stores the addresses of n free blocks in the first free block.
- The first n−1 of these blocks are actually free.
- The last block contains the addresses of another n free blocks, and so on.
- The addresses of a large number of free blocks can now be found quickly, unlike the situation when the standard linked-list approach is used.

Counting

- Generally, several contiguous blocks may be allocated or freed simultaneously, particularly when space is allocated with the contiguous-allocation algorithm or through clustering.
- Rather than keeping a list of n free disk addresses, we can keep the address of the first free block and the number (n) of free contiguous blocks that follow the first block
- Each entry in the free-space list then consists of a disk address and a count.
- The entries can be stored in a balanced tree, rather than a linked list, for efficient lookup, insertion, and deletion.

THANK YOU

Suresh Jamadagni
Department of Computer Science Engineering

sureshjamadagni@pes.edu