Allocation of Free Disk Blocks in UNIX using Bitmaps

S.Ayesha Firadose, S.Khadeeja Banu

Abstract— The utility program mkfs organizes the data blocks of a file system in a linked list. Each link of the list is a disk block that contains an array of free disk block numbers, and one array entry is the number of the next block of the linked list. The file system super block contains an array that is used to cache the numbers of free disk blocks in the file system. Because of this technique, there is a wastage of memory. To overcome this disadvantage we can use bitmaps. In bitmaps only one bit can be used to indicate the status of disk block. Using this technique memory space is utilized efficiently and wastage of processor time for reading one or more disk blocks in main memory is reduced.

Index terms— disk block, super block, bit map.

I. INTRODUCTION

Unix is a multitasking, multi-user computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, Michael Lesk and Joe Ossanna.

Unix operating systems are widely used in servers, workstations, and mobile devices. The Unix environment and the client–server program model were essential elements in the development of the Internet and the reshaping of computing as centered in networks rather than in individual computers.

The file subsystem is one of the component in UNIX operating system also called kernel. The file subsystem accesses file data using a buffering mechanism that regulates data flow between the kernel and secondary storage devices.

A file system consists of a sequence of logical blocks, each containing 512, 1024, 2048 or any convenient multiple of 512 bytes, depending on the system implementation.

A File system has the following structure.

Boot block Super block Inode list Data blocks

File system layout

- ✓ The boot block is the first sector and contains the boostrap code that is read into the machine.
- ✓ The super block describes the state of a file system like how large it is, how many files it can store, where to find free space on the file system and other information.
- ✓ The inode list is a list of inodes that follows the super block in the file system.
- ✓ The data blocks contain file data and administrative data.

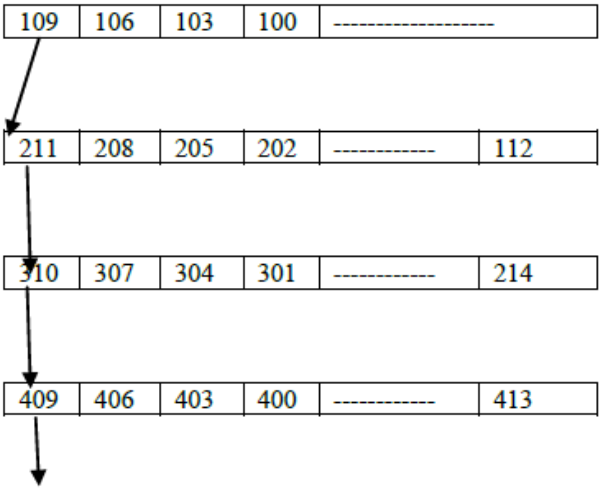
The super block consists of the following fields

- ✓ The size of the file system
- ✓ The number of free blocks in the file system
- ✓ A list of free blocks available on the file system
- ✓ The index of the next free block in the free block list
- ✓ The size of the inode list
- ✓ The number of free inodes in the file system
- ✓ A list of free inodes in the in the file system
- ✓ The index of the next free inode in the free inode list
- ✓ Lock fields for the free block and free inode list
- ✓ A flag indicating that super block has been modified

A. Process of allocating a block

When the kernel wants to allocate a block from a file system, it allocates the next available block in the super block list. Once allocated, the block cannot be reallocated until it becomes free.

Super block list



Linked list of free disk block numbers

If the allocated block is the last available block in the super block cache, the kernel treats it as a pointer to a block that contains a list of free blocks. It reads the block, populates the super block array with the new list of block numbers, and then proceeds to use the original block number. It allocates a buffer for the block and clears the buffer's data. The disk block has now been assigned, and the kernel has a buffer to work with. If the file system contains no free blocks, the calling process receives an error.

The disadvantages in this technique for organizing free disk block numbers are

1. Wastage of memory space for maintaining large number of free disk blocks
2. One or more disk blocks should be read into main memory for allocating or freeing disk blocks.

Manuscript received October, 2013.

S.Ayesha firadose, Computer Science and Engineering, Srenivasa Institute of Technology and Management Studies,Chittoor, Andhra Pradesh, India.

S.Khadeeja Banu, Computer Science and Engineering, Srenivasa Institute of Technology and Management Studies,Chittoor, Andhra Pradesh, India.

B. Process of freeing a block

If the super block list is not full, the block number of the newly freed block is placed on the super block list.
If the super block list is full, the newly freed block becomes a link block. The kernel writes the super block list into the free block and writes the block to disk. It then places the block number of the newly freed block in the super block list.

II. BITMAPS

To overcome the above disadvantages, bitmap can be used to maintain the free disk block numbers. Here a single bit will represent the status of the block.
Following figure shows a bitmap for disk block numbers up to 16.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	0	1	1	0	0	0	1	0	1	1	0	1

Bitmap for disk block numbers .

In this bitmap , the disk block numbers 0,1,4,7,8,9,11 and 14 are free .
The disk block numbers 2,3,5,6,10,12,13 and 15 are allocated.
Suppose, if there are 256 block numbers. Then the index of the bit positions in the bit map is 0 to 255. Bit 0 represents the status of disk block 0, bit 1represents the status of disk block 1 and so on.
If the bit in the bitmap at position 'b' is 0, then the disk block number b is free , otherwise it is allocated. In this technique, instead of storing 16 bit or 32 bit block numbers, only a single bit will indicate whether the block is allocated or free.

A. Algorithm for disk block allocation

1. Scan the bitmap for the bit 0.
2. Get the index of the bit 0 and set that bit to 1.
3. Return the index as the number of the disk block that is allocated.

B. Algorithm for freeing a disk block

1. Convert the number of disk blocks as index into the bitmap.
2. Using the index value, set the corresponding bit in the bitmap to 0.
3. Zero indicates that the block is free.

C. Advantages of bitmaps for disk block allocation

- ✓ In bitmaps only one bit can be used to indicate the status of disk block. Using this technique memory space is utilized efficiently
- ✓ Wastage of processor time for reading one or more disk blocks in main memory is reduced.

III. CONCLUSION

Bitmaps can be used for the allocation of memory pages, inodes, disk sectors, etc. Another application of bit arrays is the Bloom filter, a probabilistic set data structure that can store large sets in a small space in exchange for a small probability of error. It is also possible to build probabilistic hash tables based on bit arrays that accept either false positives or false negatives.

REFERENCES

[1] The design of UNIX operating system by Maurice J. Bach
[2] Christian, K., The UNIX operating system, John Wiley & Sons Inc., New York, NY, 1983.
[3] Cole, C.T., P.B. Flinn and A.B. Atlas, "An Implementation of an Extended File System for UNIX", Proceedings of the USENIX Conference, Summer 1985, pp. 131-149.
[4] Johnson, S. C. and D. M. Ritchie, "Portability of C programs and the UNIX system", The Bell System Technical Journal, Vol. 57, No. 6, Part 2, July-August, 1978, pp. 2021-2048.
[5] Kernighan, B.W., and R. Pike, The UNIX Programming Environment, Prentice-Hall, Englewood cliffs, NJ, 1984.
[6] Sandberg, R., D. Goldberg, S. Kleiman, D.Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem" Proceedngs of the USENIX conference, Summer 1985, pp. 119-131.