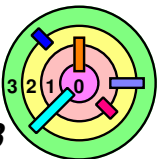


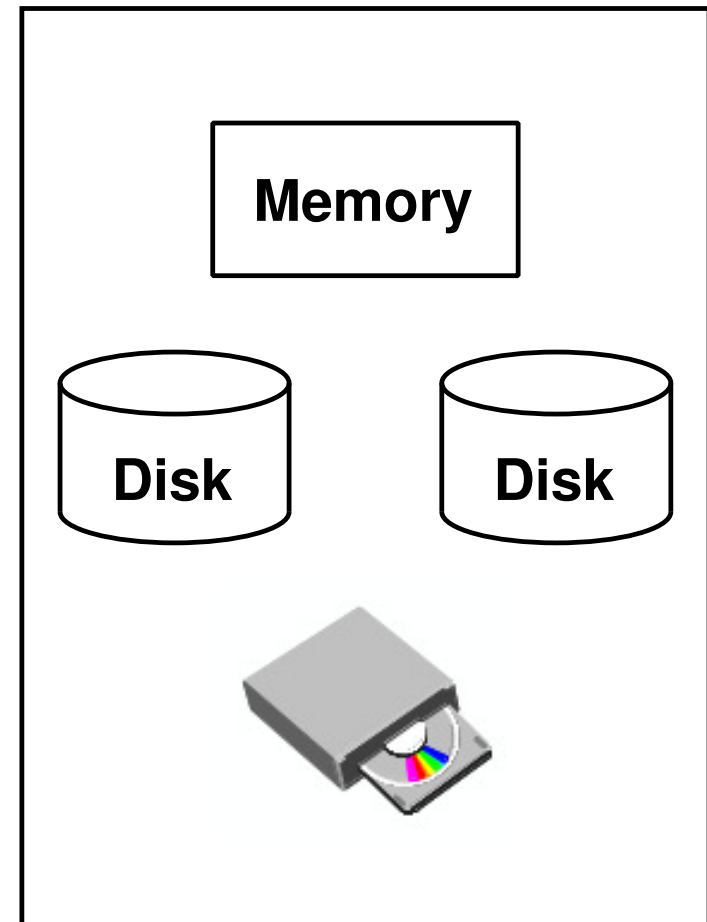
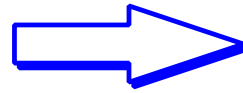
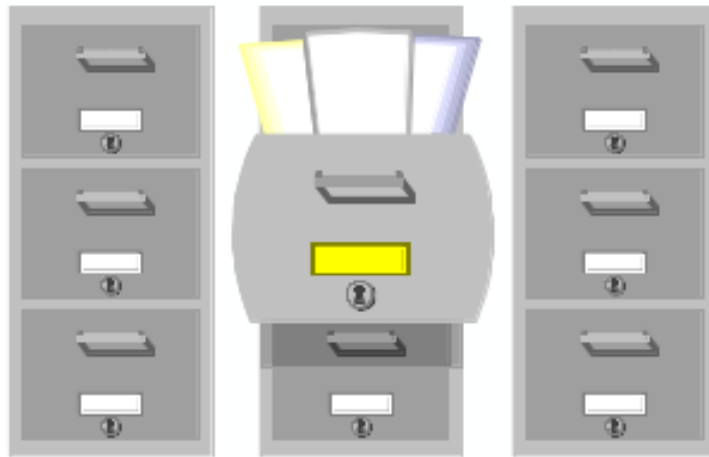
Ch 6: File Systems

Bill Cheng

<http://merlot.usc.edu/william/usc/>

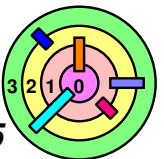


Files



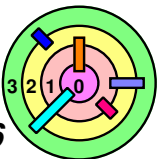
Requirements

- ➡ **Permanent storage**
 - ▬ resides on disk (or alternatives)
 - ▬ survives software and hardware crashes
 - (including loss of disk?)
- ➡ **Quick, easy, and efficient**
 - ▬ satisfies needs of most applications
 - how do applications use permanent storage?



Applications

- ➡ **Software development**
 - ▬ text editors
 - ▬ linkers and loaders
 - ▬ source-code control
- ➡ **Document processing**
 - ▬ editing
 - ▬ browsing
- ➡ **Web stuff**
 - ▬ serving
 - ▬ browsing
- ➡ **Program execution**
 - ▬ paging



Needs



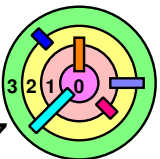
Directories

- convenient naming
- fast lookup



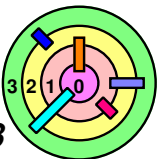
File access

- sequential is very common!
- "random access" is relatively rare



6.1 The Basics of File Systems

- ➡ *UNIX's S5FS*
- ➡ Disk Architecture
- ➡ Problems with S5FS
- ➡ Improving Performance



S5FS



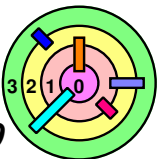
A simple file system

- slow
- not terribly tolerant to crashes
- reasonably efficient in space
 - no compression

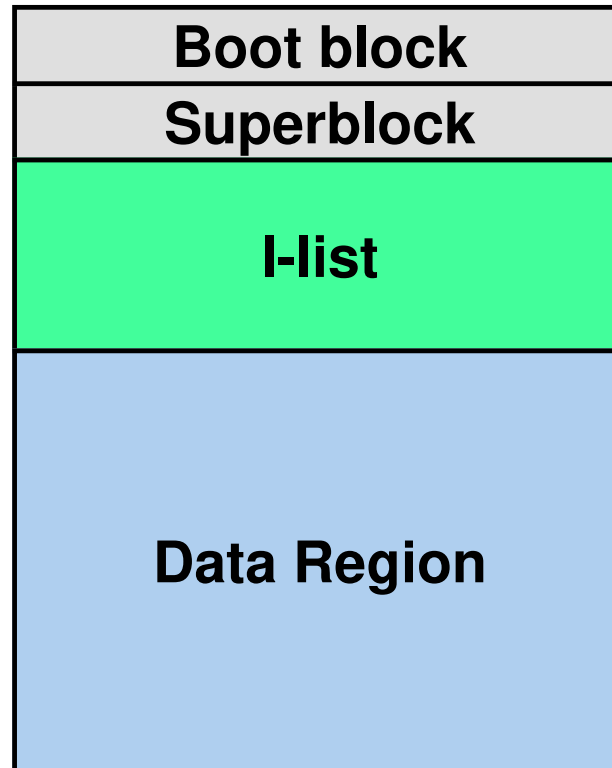


Concerns

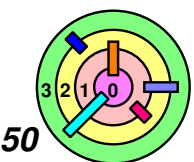
- on-disk data structures
 - file representation
 - ◆ recall that AFS translates an inode number to disk address
 - free space



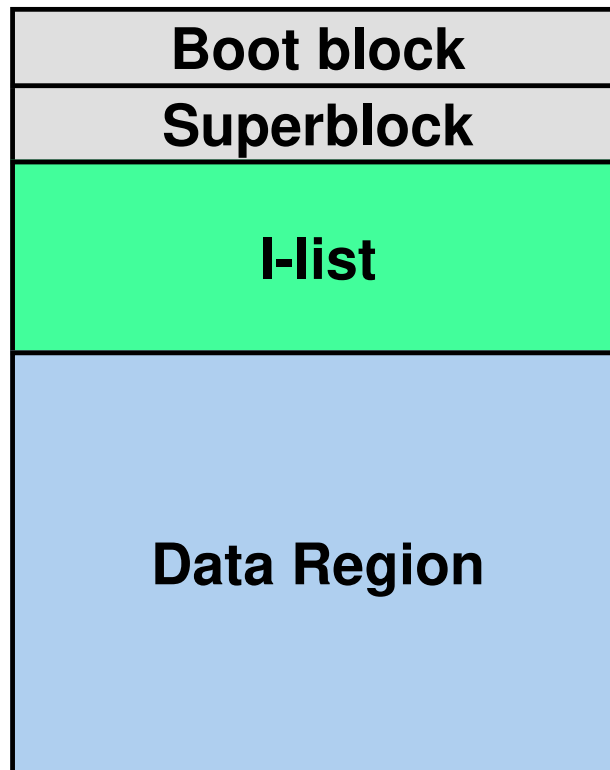
S5FS Layout



- ➡ A disk is simply an array of blocks of 1KB each (old Unix: 512B)
 - a disk block is a *logical* unit (usually multiple of page size)
- ➡ A "linear view" (1-D array of blocks) of the disk



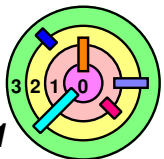
S5FS Layout



where are the "free space"?

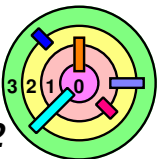
- inodes can be free or not
- data block can be free or not

- ➡ The *superblock*
 - describes the layout of the rest of the file system
 - contains the *head* of the *free list*
- ➡ The *i-list* is an *array* of *index nodes (inodes)*
 - each representing a *file*



S5FS: Inode

Device
Inode Number
Mode (File Type)
Link Count
Owner, Group
File Size
Disk Map



Disk Map

inode

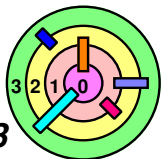
...
0
1
2
3
...
7
8
9
10
11
12

- ▢ assuming blocksize = 1KB
- to make math easier

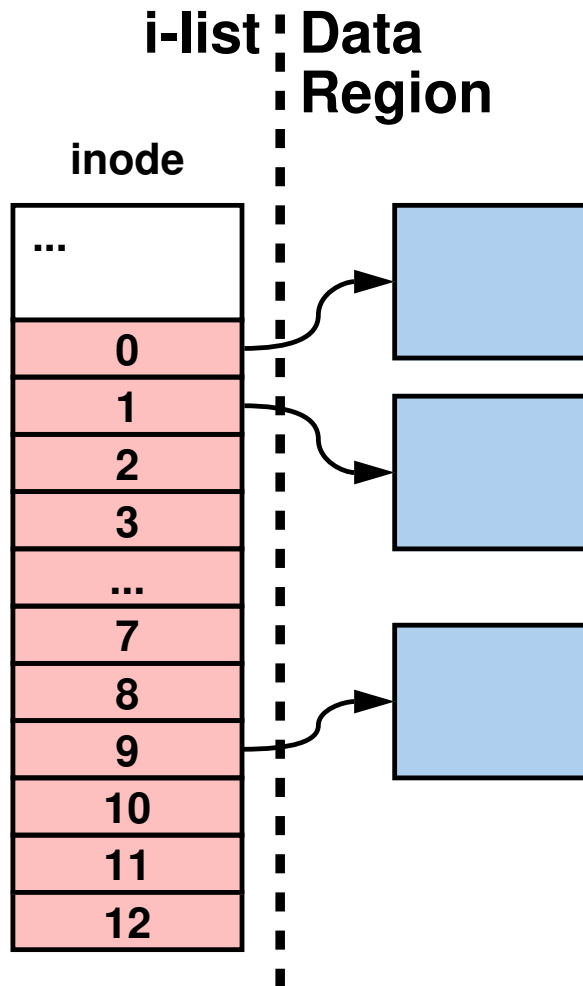


Disk Map contains 13 disk block pointers

- ▢ a disk block pointer is just a block number
 - pointers 0 through 9 are direct pointers
 - pointer 10 is an indirect pointer
 - pointer 11 is an double-indirect pointer
 - pointer 12 is an triple-indirect pointer

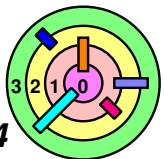


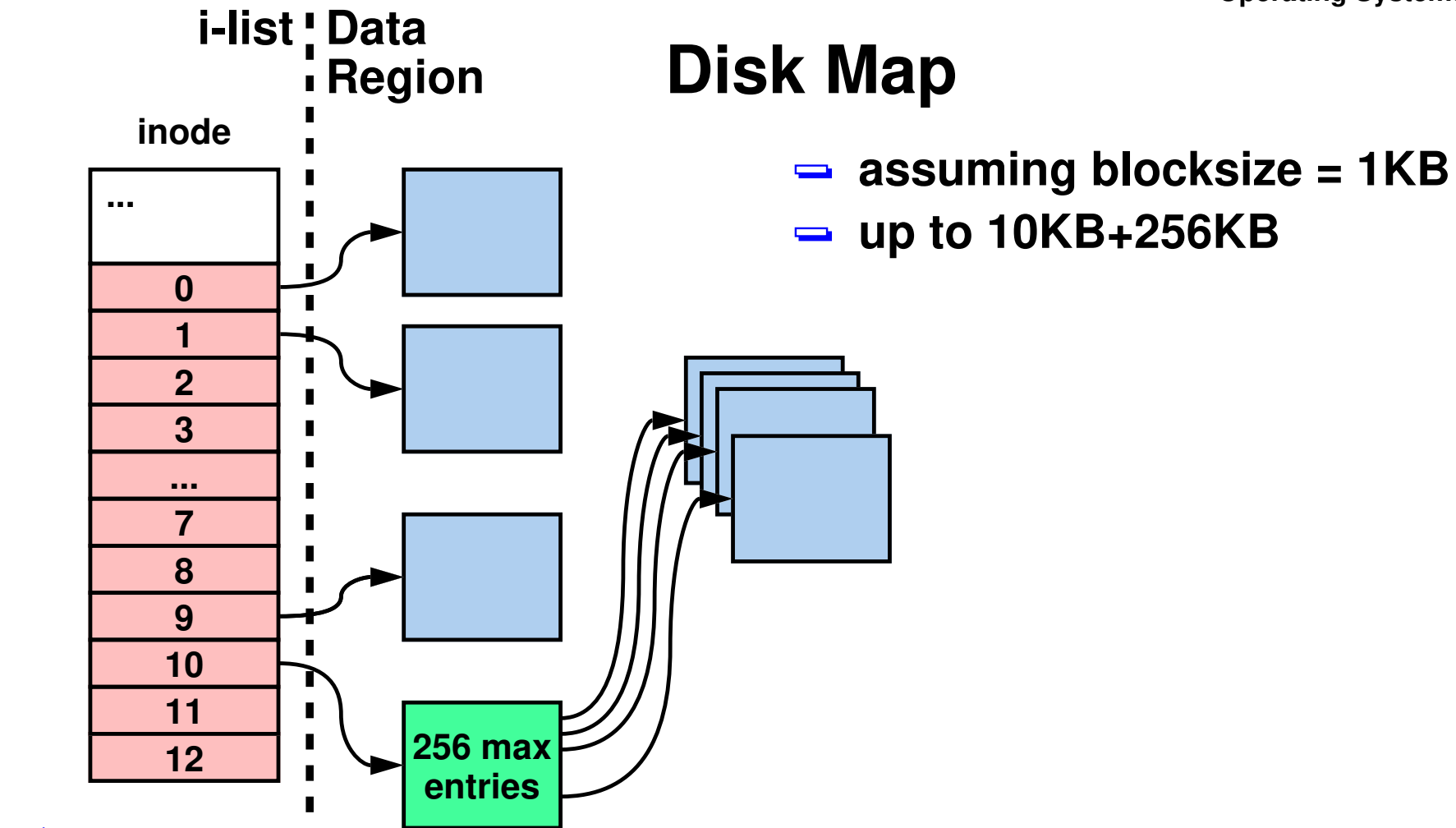
Disk Map



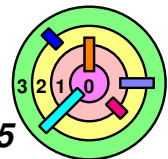
- assuming blocksize = 1KB
- up to 10KB

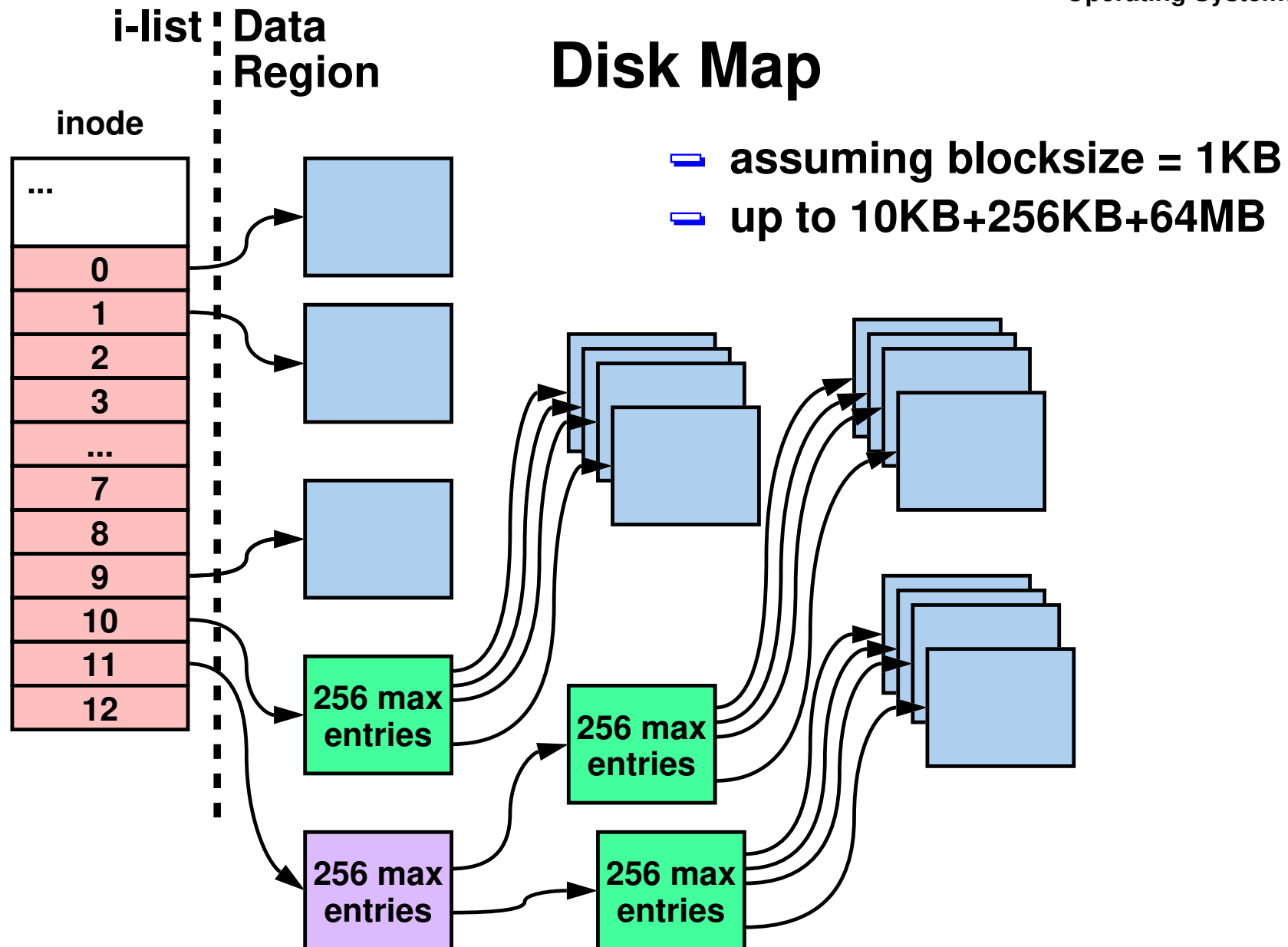
- ➡ Disk Map contains 13 disk block pointers
 - a disk block pointer is just a block number
- ➡ Some blocks in the data region contains data

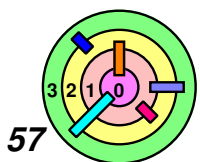
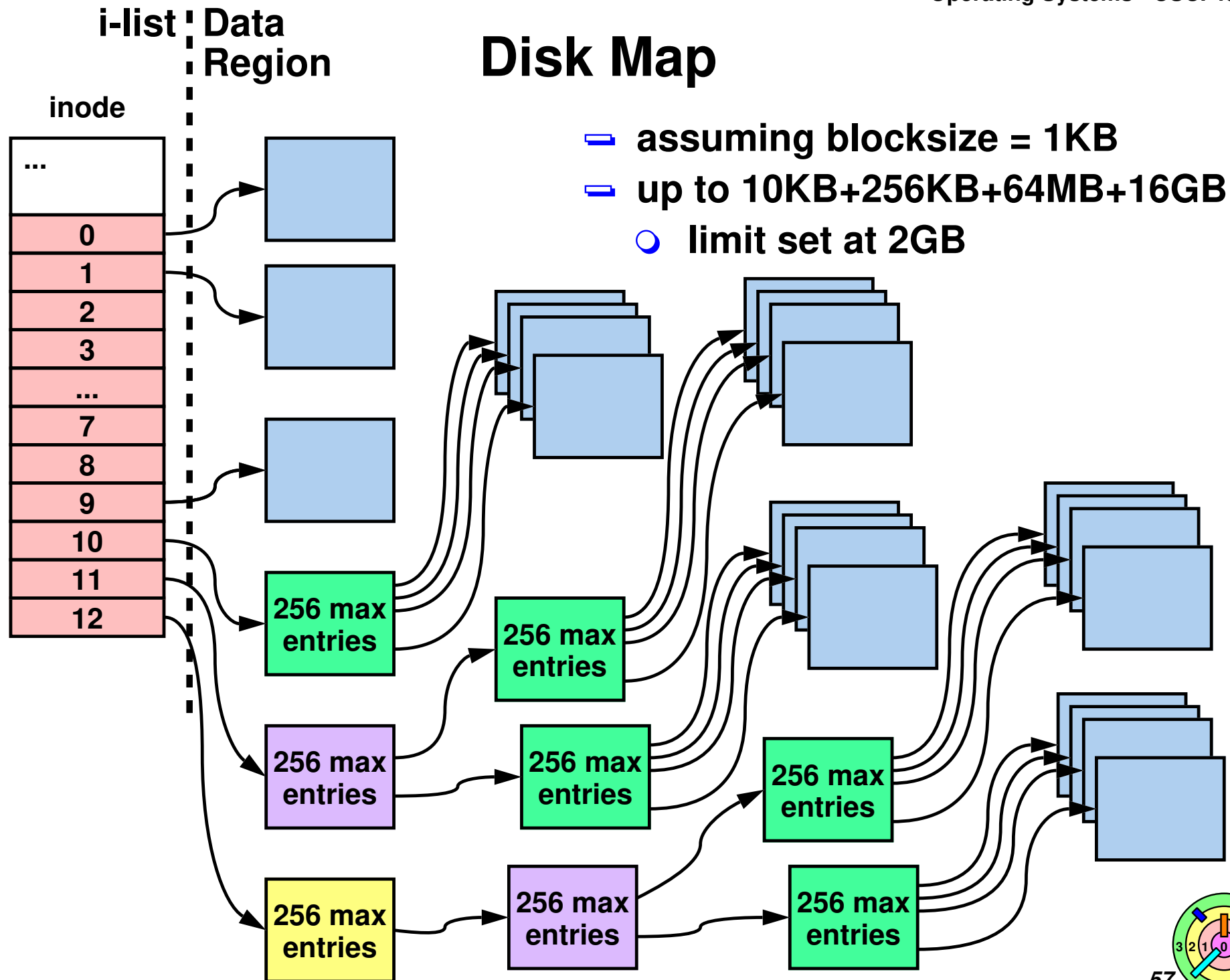


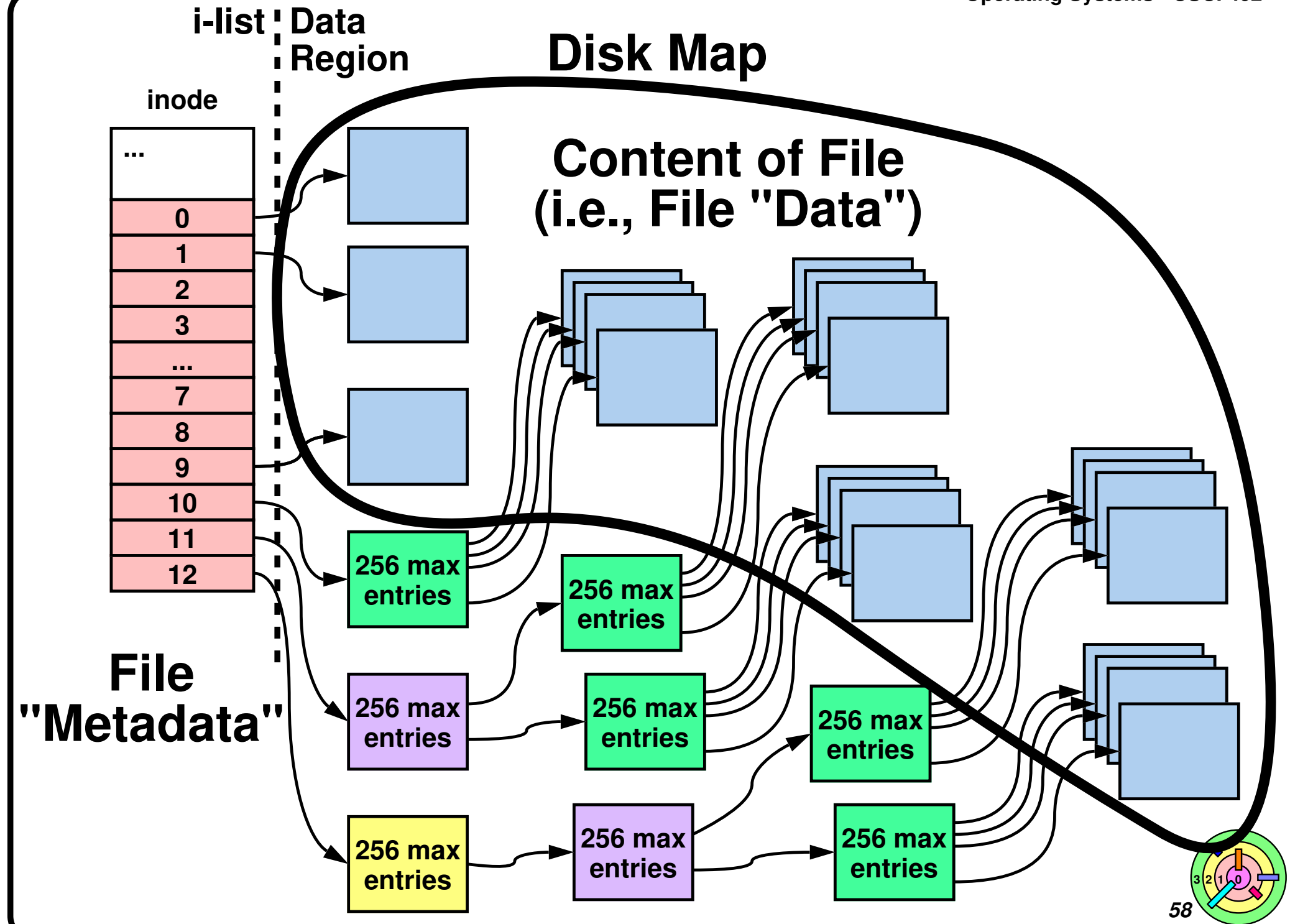


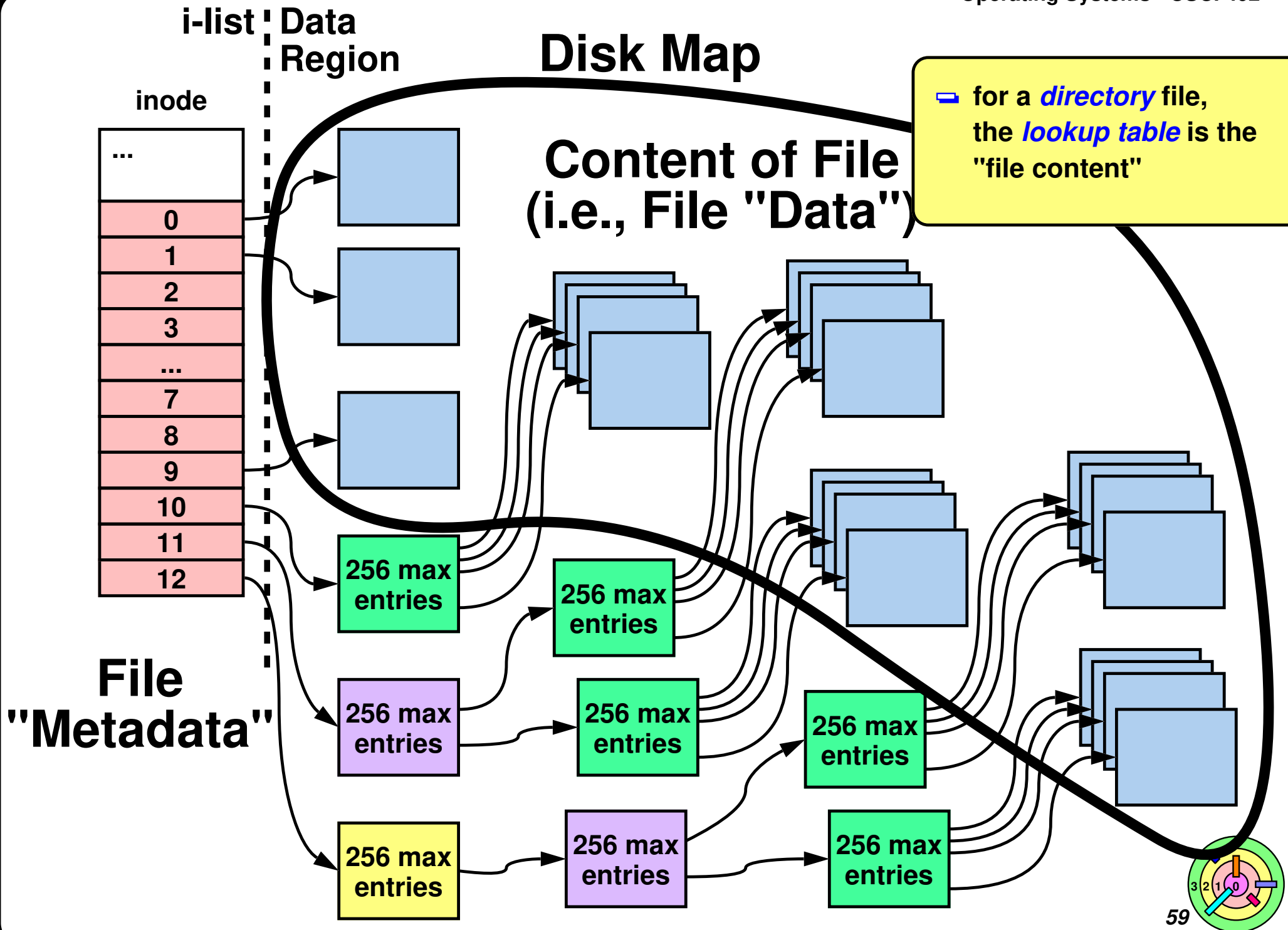
- ➡ Disk Map contains 13 disk block pointers
 - a disk block pointer is just a block number
- ➡ Some blocks in the data region contains data
 - some blocks in the data region contains data structures



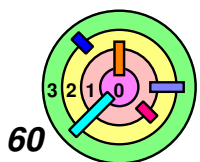
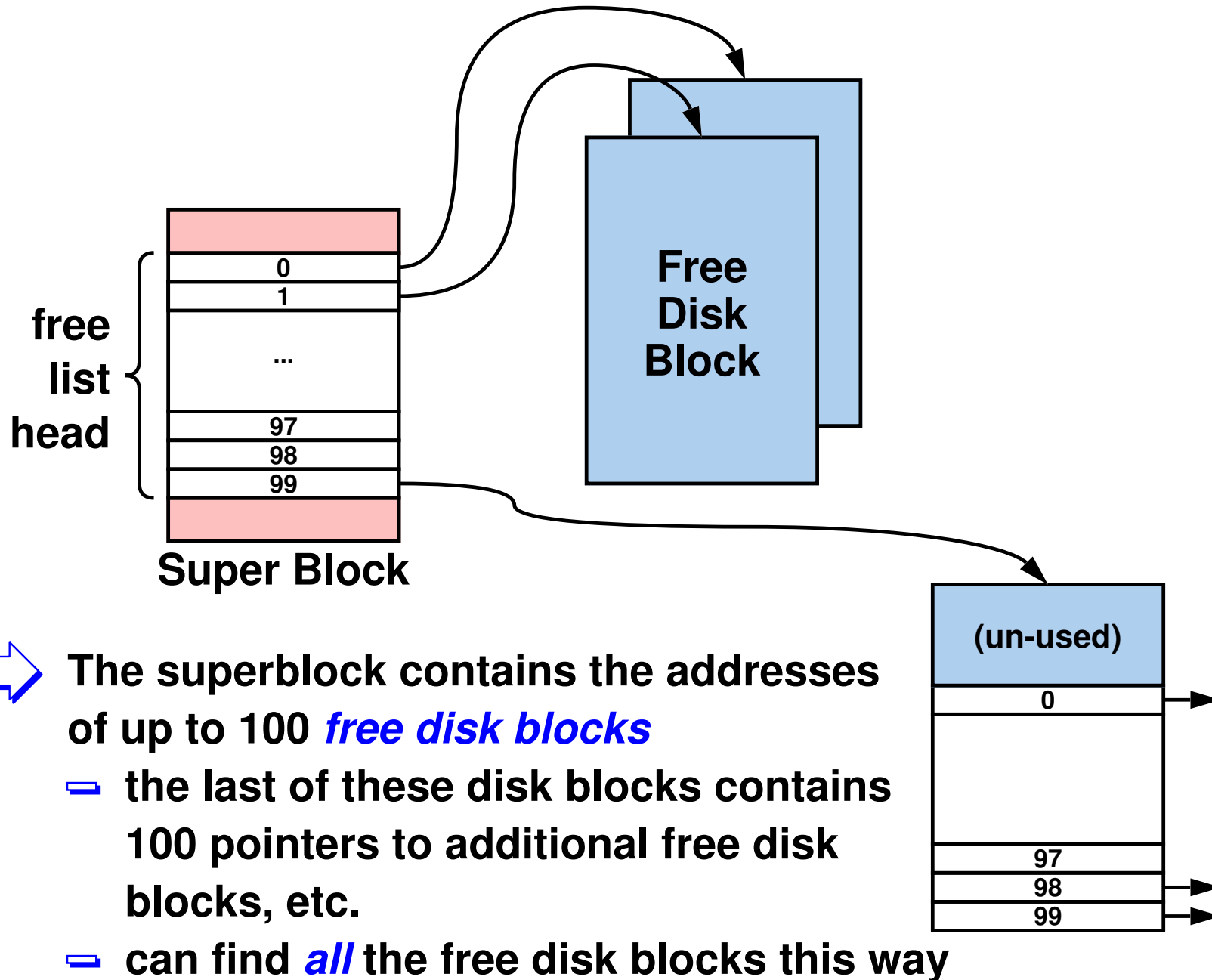




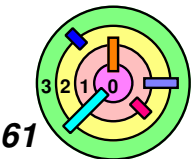
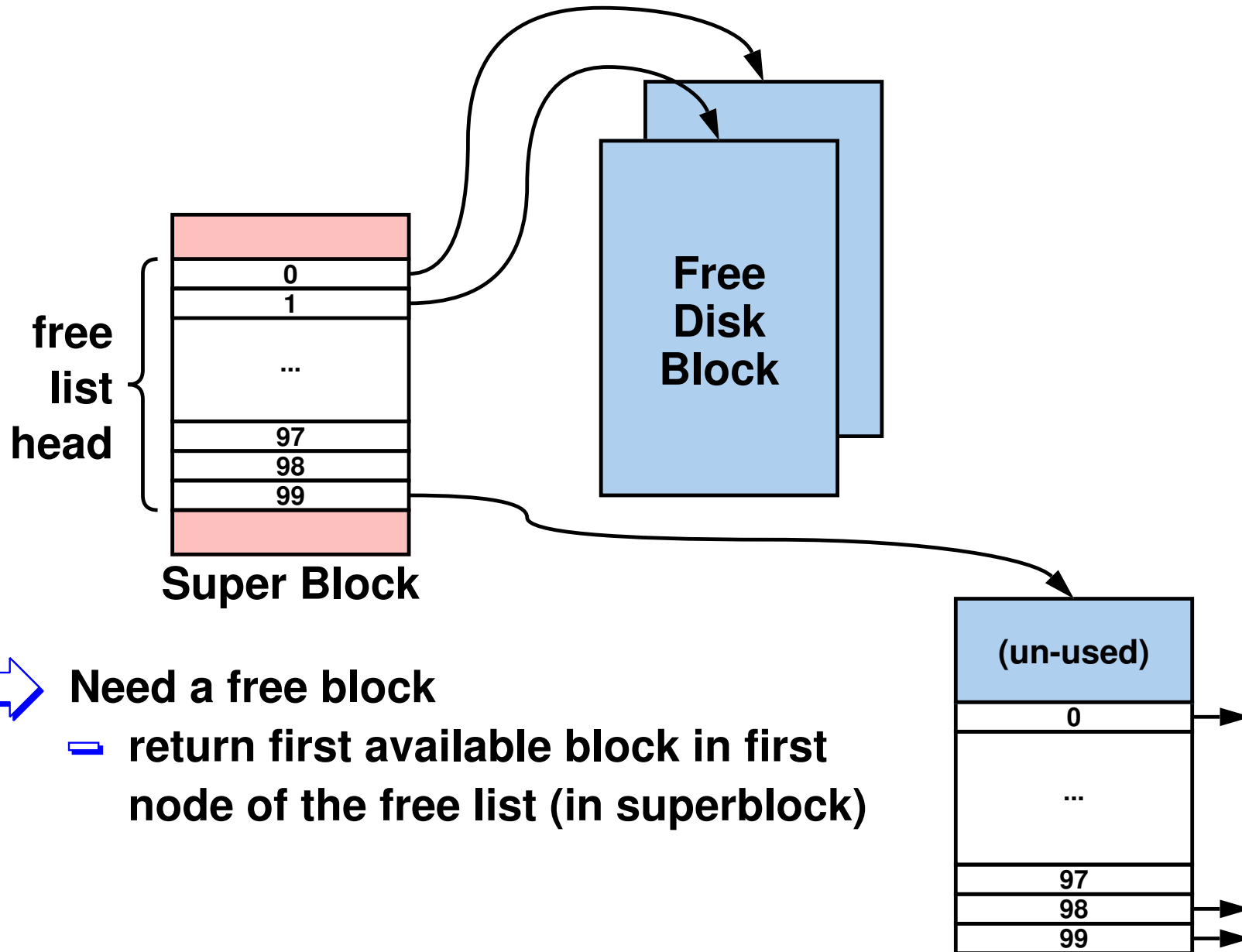




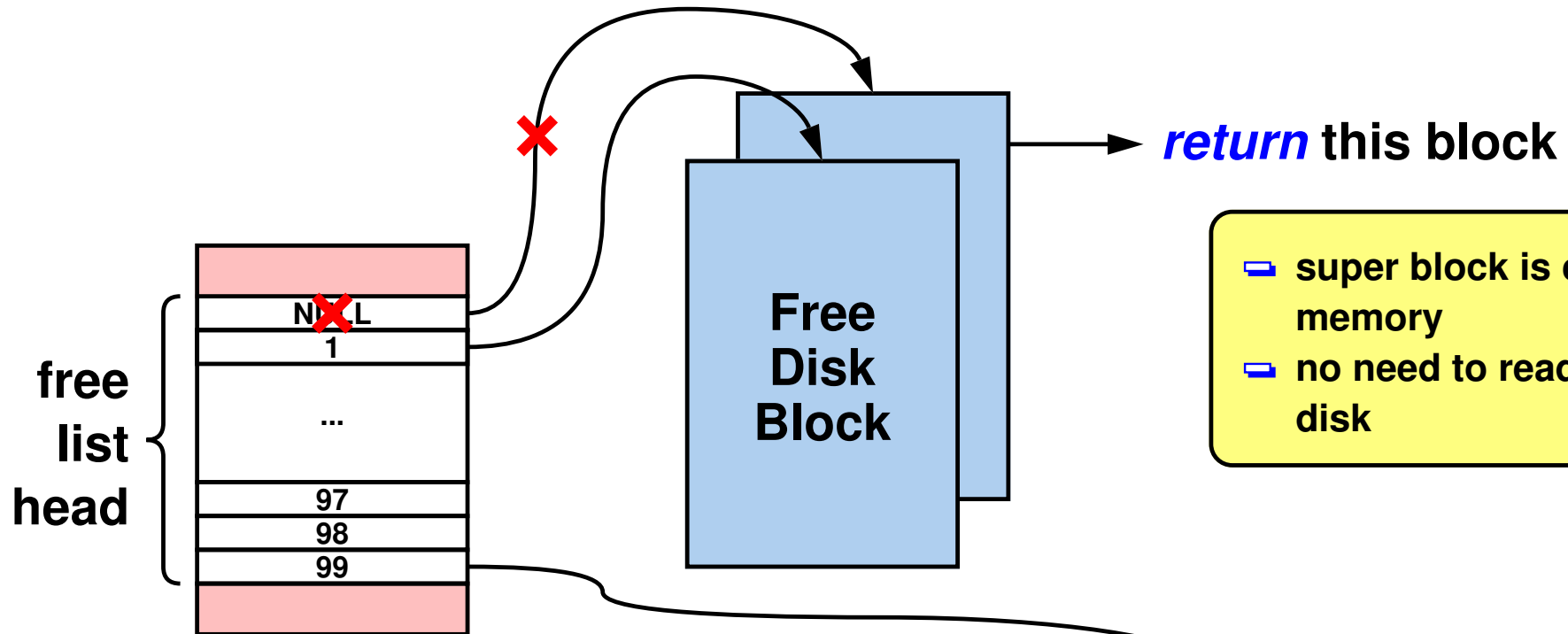
S5FS Free List



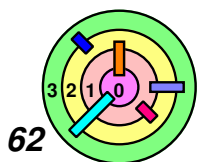
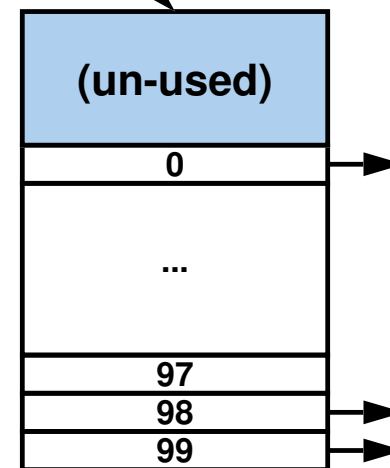
S5FS Free List: Allocation - Case (1)



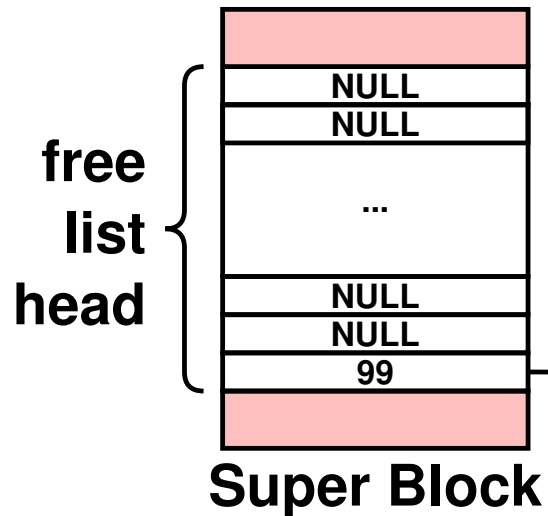
S5FS Free List: Allocation - Case (1)



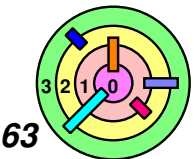
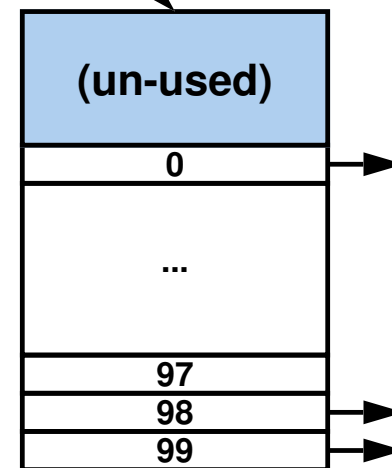
- ➡ **Need a free block**
- return first available block in first node of the free list (in superblock)



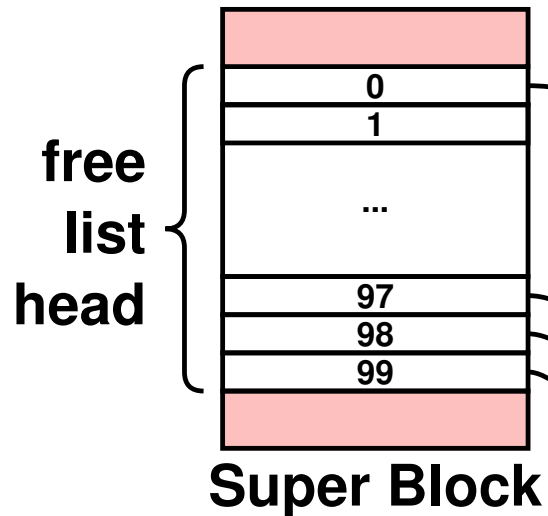
S5FS Free List: Allocation - Case (2)



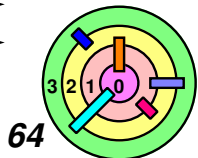
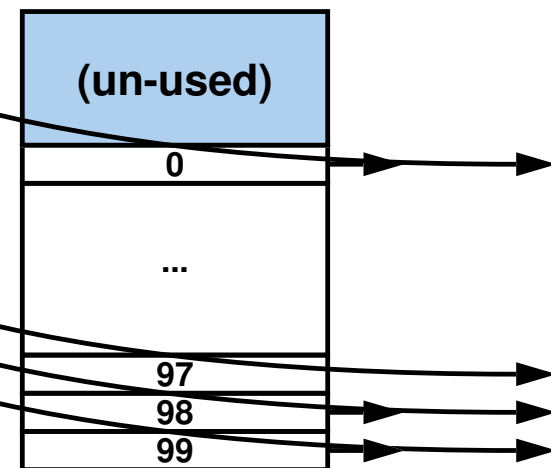
- ➡ Need a free block
- return first available block in first node of the free list (in superblock)
 - if no free block in first node
 - first copy 100 pointers into the superblock with one disk access



S5FS Free List: Allocation - Case (2)

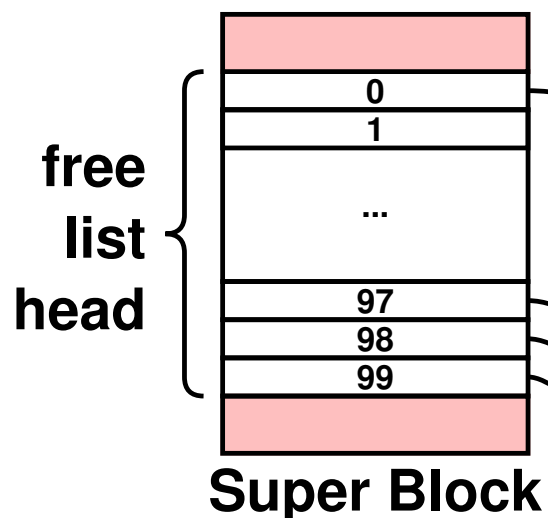


- ➡ Need a free block
- return first available block in first node of the free list (in superblock)
 - if no free block in first node
 - first copy 100 pointers into the superblock with one disk access



S5FS Free List: Allocation - Case (2)

- one disk read to copy disk block pointers
- only happens 1 out of 100 allocations



return this block since it has been **unlinked** from the free list

- ➡ Need a free block
- return first available block in first node of the free list (in superblock)
 - if no free block in first node
 - first copy 100 pointers into the superblock with one disk access

