操作系统

Operating system

孔维强

大连理工大学

内容纲要

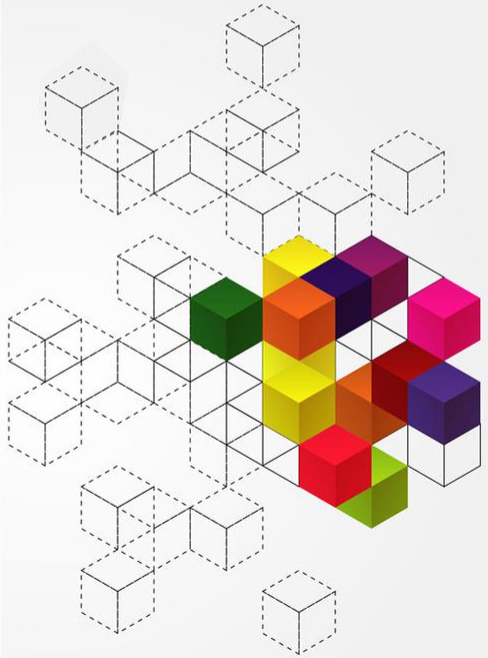
11.4 文件磁盘空间分配

一、 文件物理结构

二、 连续分配

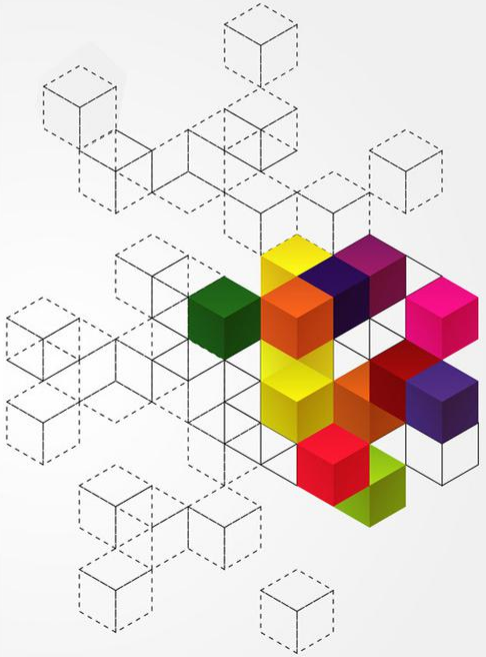
三、 链式分配

四、 索引分配



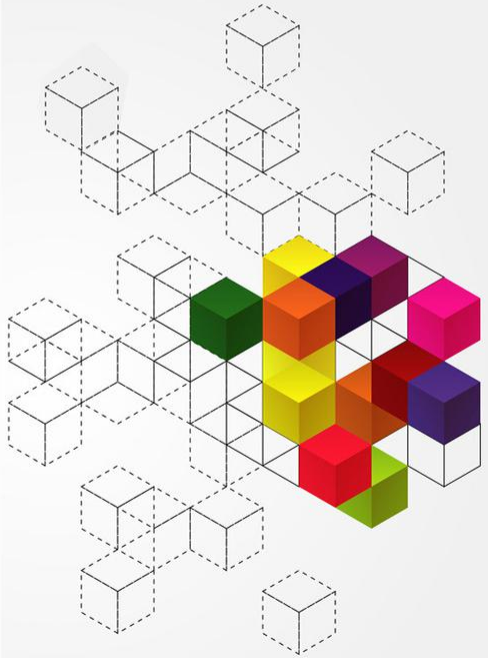
一、文件物理结构

- **文件物理结构：如何为文件逻辑空间进行实际的磁盘空间分配**
  - 操作系统需要决定为文件分配哪些磁盘块，以便将文件的逻辑数据完整地存在外存设备上
  - 如何分配对于高效利用磁盘非常重要



一、文件物理结构

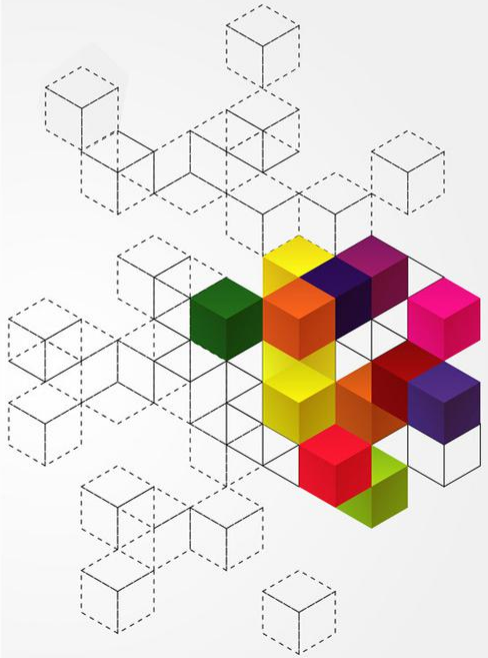
- 三种典型的文件物理结构
  - 连续分配（Contiguous Allocation）
  - 链接分配（Linked Allocation）
  - 索引分配（Indexed Allocation）



二、连续分配

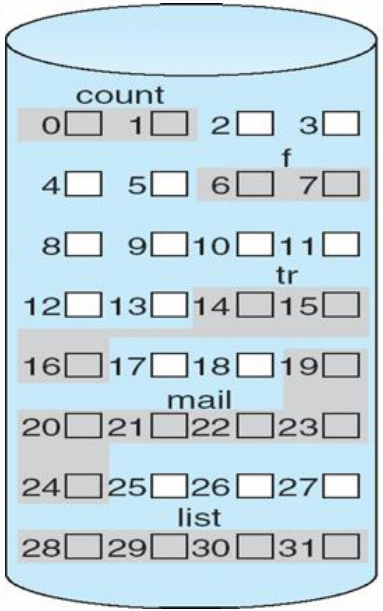
(1) Contiguous allocation

- Each file occupies set of contiguous blocks
  - Best performance in most cases
  - Simple – only starting location (block #) and length (number of blocks) are required (b, b+1, b+2, …, b+n-1)
  - Random access
  - Problems include
    - finding space for new file (dynamic allocation problem: first-fit, best-fit, worst-fit),
    - external fragmentation, need for **compaction off-line** (**downtime**) or **on-line**



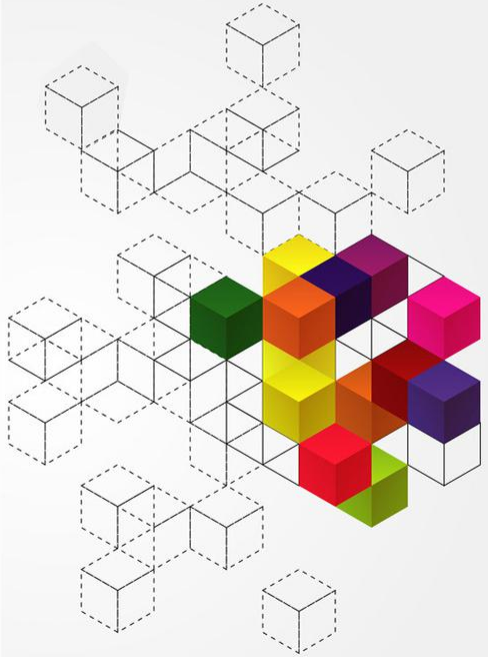
二、连续分配

- 连续分配磁盘空间



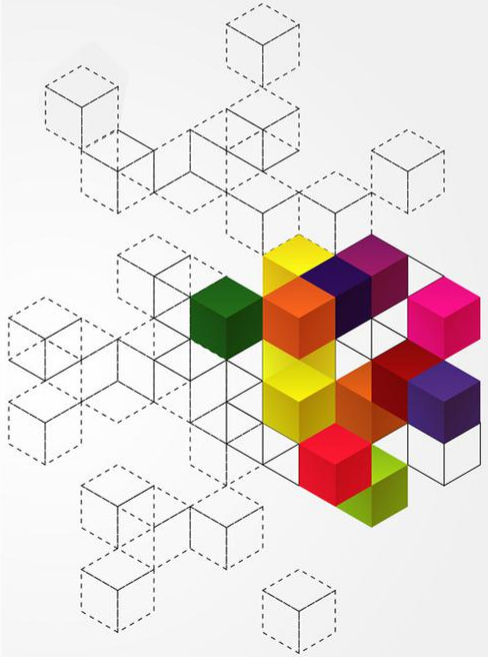
directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2



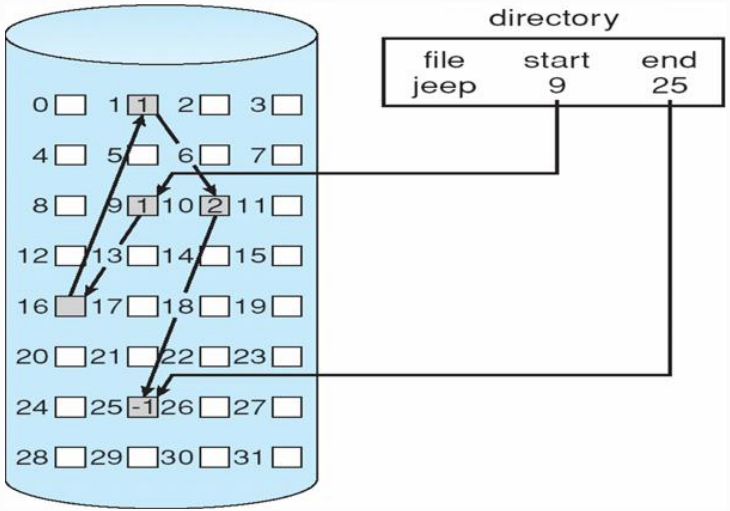
三、链接分配

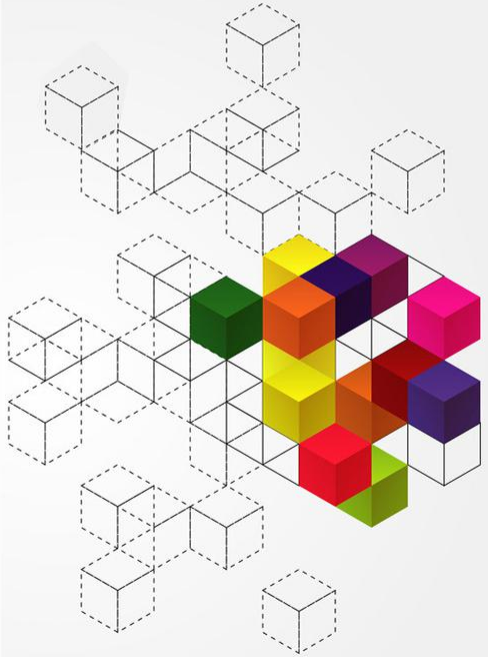
- **(2) Linked allocation** – each file is a linked list of disk blocks, blocks can be scattered anywhere in the disk
  - **File ends at nil pointer**
  - **No external fragmentation**
  - **Each block contains pointer to next block**
  - **No compaction, external fragmentation**
  - **Free space management system called when new block needed**
  - **Problems:**
    - **Reliability can be a problem**
    - **Locating a block can take many I/Os and disk seeks (only efficient for sequential-access)**
    - **Space for saving pointers**
      - **Improve efficiency by clustering blocks into groups but increases internal fragmentation**



三、链接分配

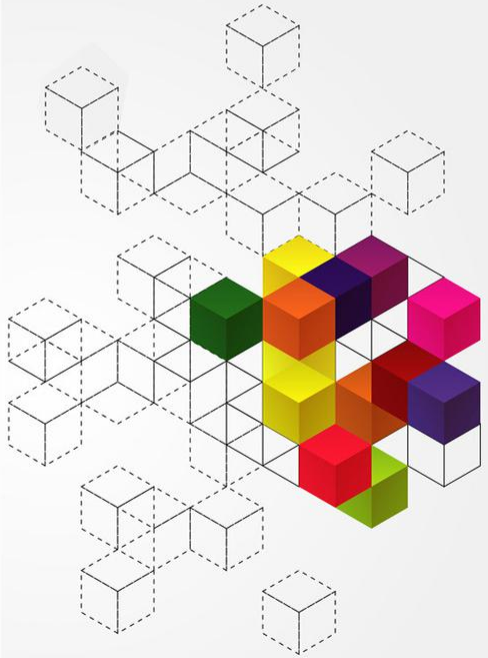
- 离散分配文件数据块，并以链表的形式组织



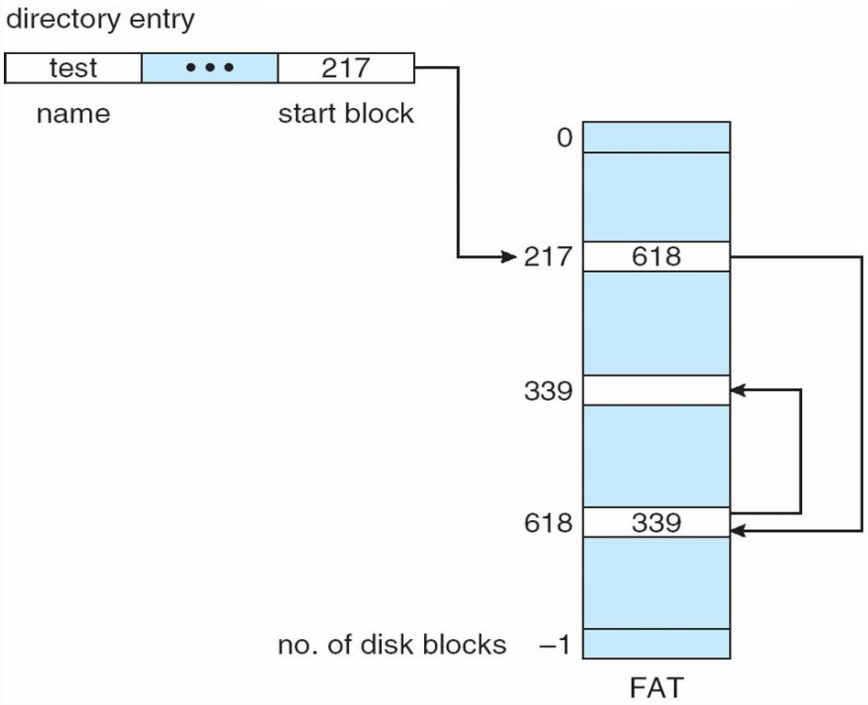


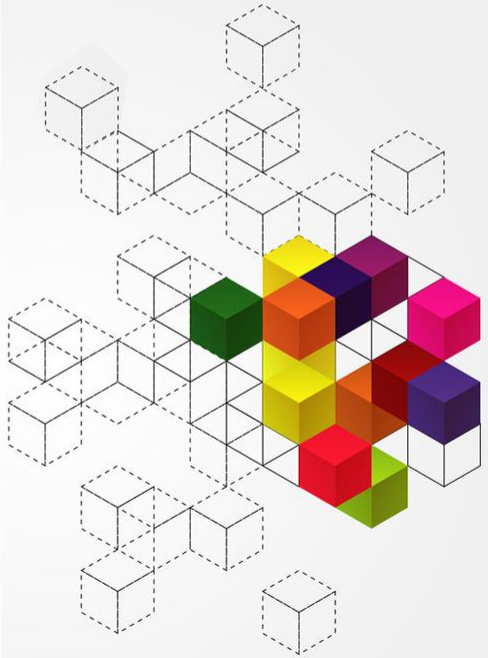
三、链接分配

- FAT (File Allocation Table) variation
  - **Beginning of volume has table, indexed by block number**
  - **Much like a linked list, but faster on disk and cacheable**
  - **New block allocation simple**
    - **Since unused blocks are indicated by a 0 table value;**
    - **Find the first 0-valued block location**



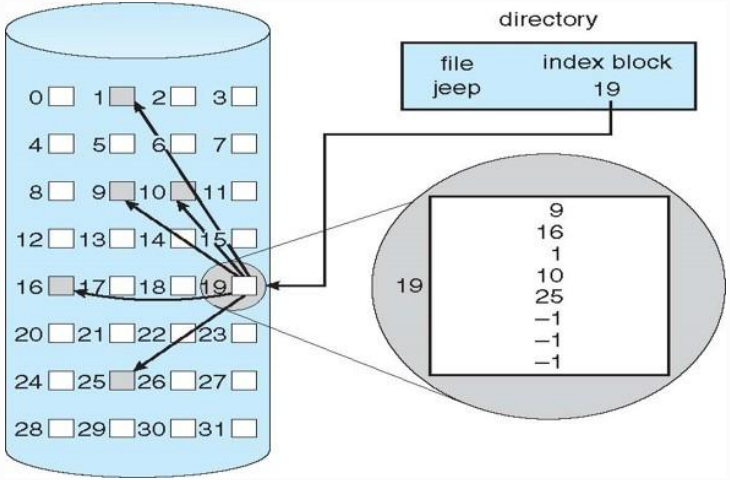
三、链接分配

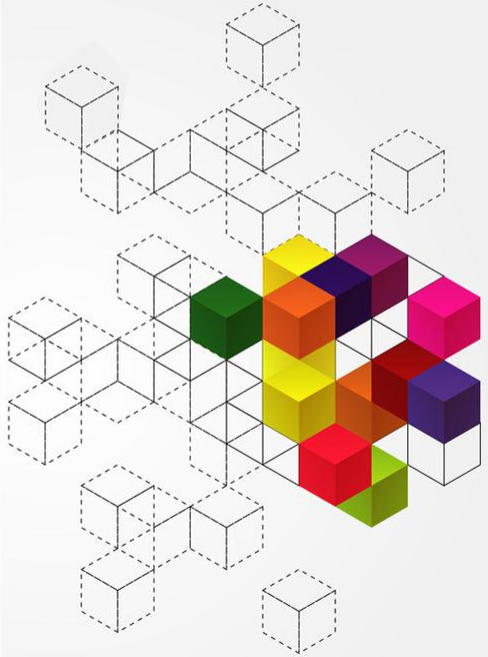

directory entry
test
• • •
217
name
start block
0
217
618
339
618
339
no. of disk blocks
–1
FAT



四、索引分配

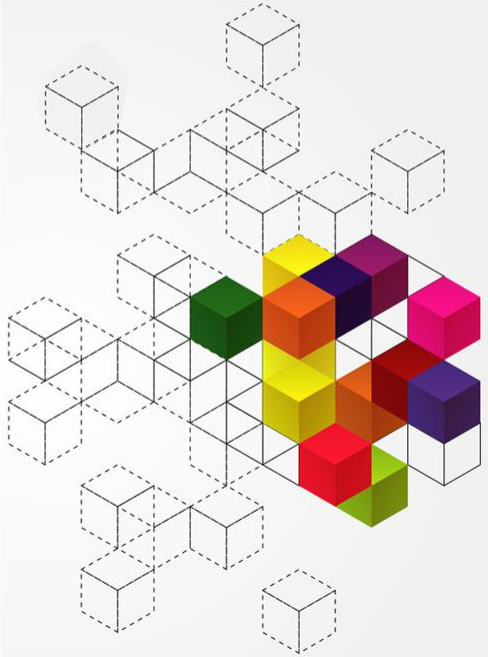
- 为文件的所有数据块建立索引，置于独立的索引磁盘块



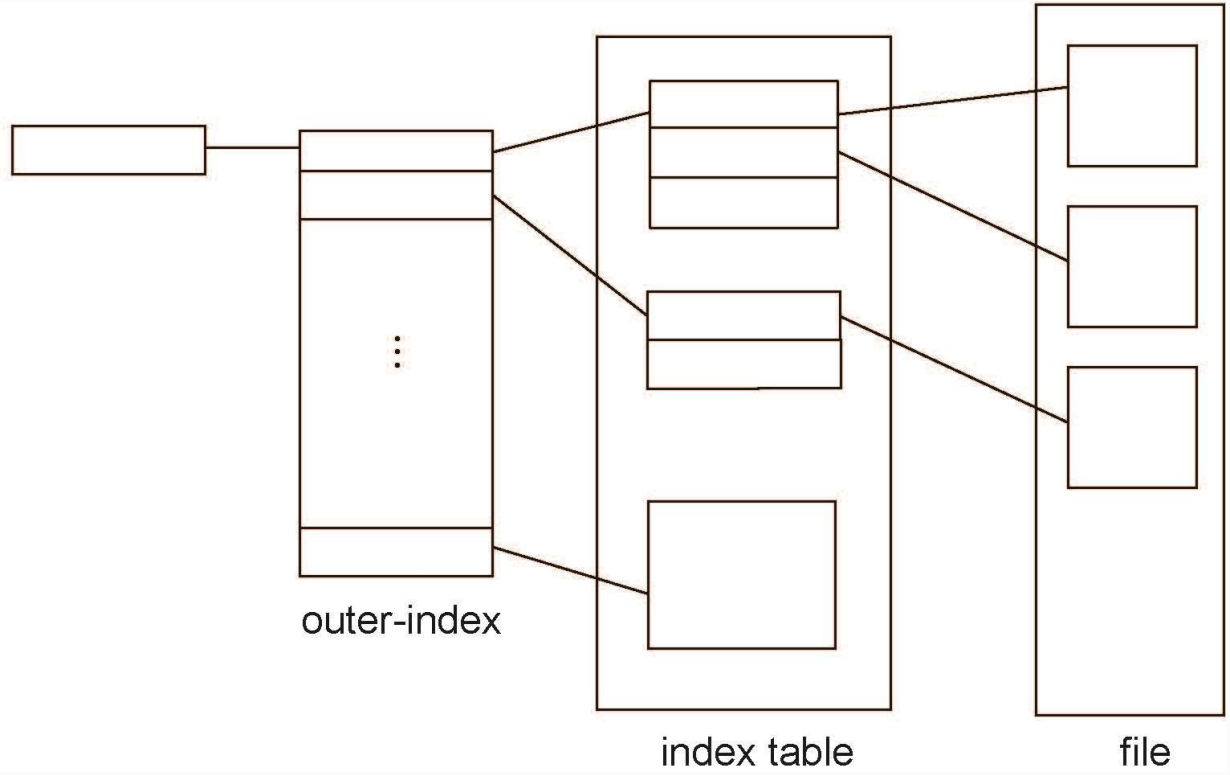


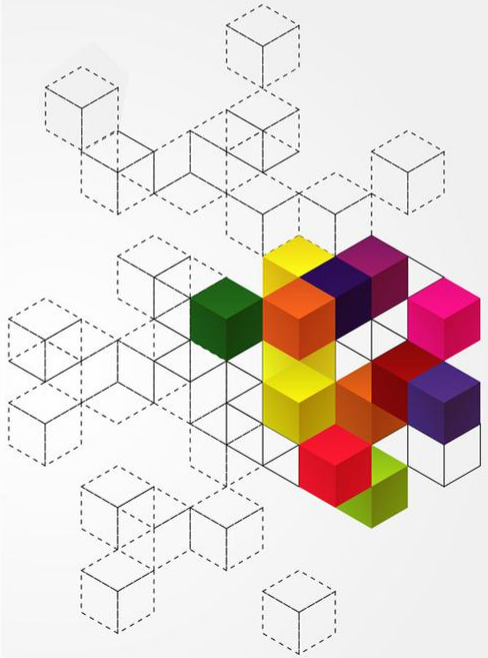
四、索引分配

- Need index table
- Random access
- Dynamic access without external fragmentation, but have overhead of index block – how large a file's index block should be?
  - **Linked scheme**
    - **Introduce more index blocks, which are linked**
  - **Multilevel index**
    - **Introduce more index blocks, index of index**
  - **Combined scheme**



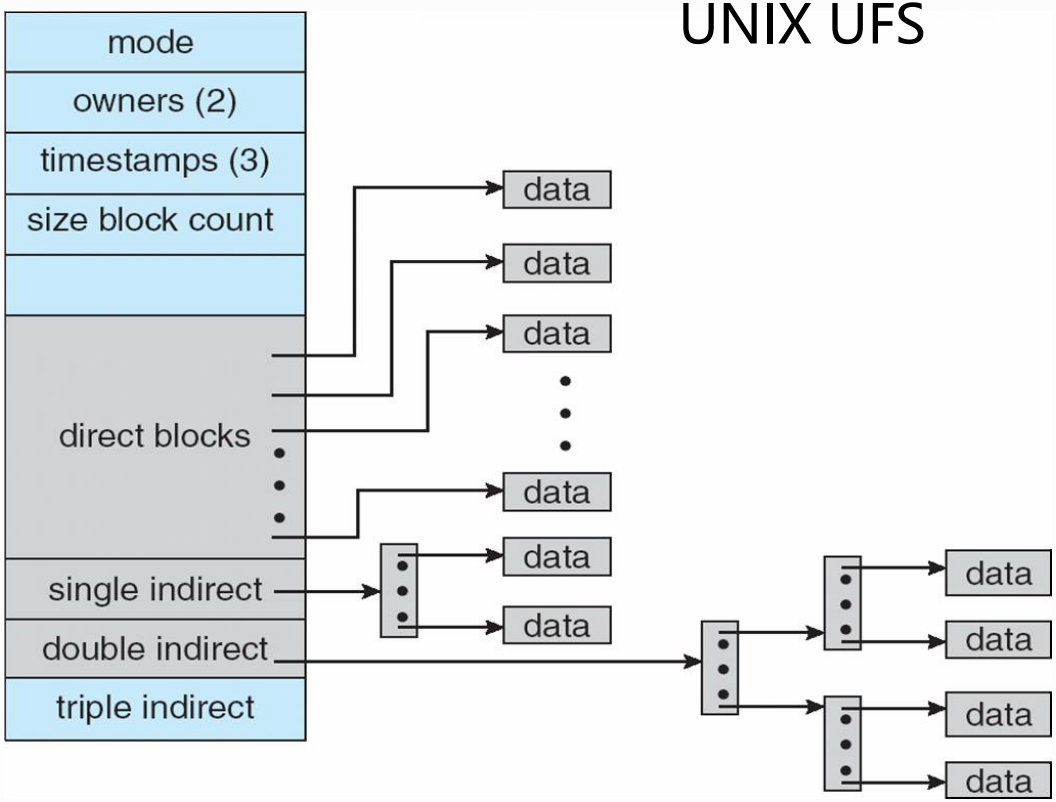
四、索引分配

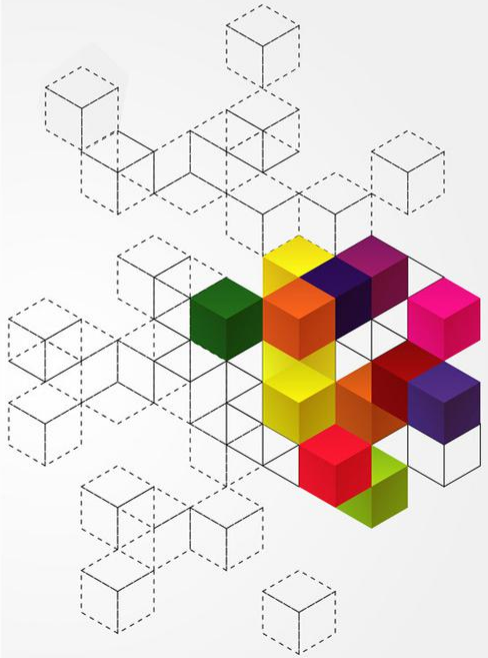




四、索引分配

- Combined Scheme: UNIX UFS

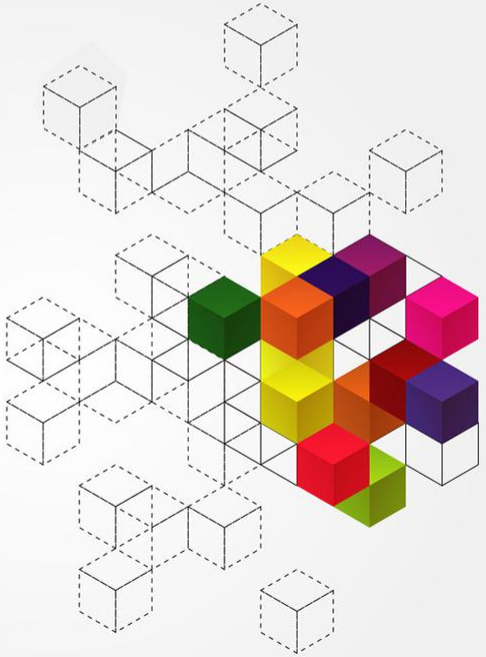




四、索引分配

Performance

- Best method depends on file access type
  - Contiguous great for sequential and random
  - Linked good for sequential, not random
- For some system that supports both, declare access type at file creation -> select either contiguous or linked
- Indexed more complex
  - Single block access could require 2 index block reads then data block read
  - Multilevel indexes may require more index reads (depends on file size)



本讲小结

- 文件物理结构
- 连续分配
- 链接分配
- 索引分配