Operating Systems

Chapter 12 File Management

Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.1 Overview(3/6)

- File System Properties(文件系统的特性)
  - Long-term existence (长期存在)
  - Sharable between processes (进程共享)
  - Structure (结构化存储)
    - Have specific structure according to application

12.1 Overview(5/6)

- File Management Systems (文件管理系统)
  - The way a user or application may access files(用户和程序使用文件的唯一方式)
  - Programmer does not need to develop file management software
  - objectives for a file management system:12.1.3
  - a minimal set of requirements in a general-purpose system :12.1.3

12.1 Overview(6/6)

File Management Functions

1.Identify (标识) and locate (定位) a selected file

- Use file name to identify files
- Use a directory (目录) to describe the location of all files plus their attributes (属性)

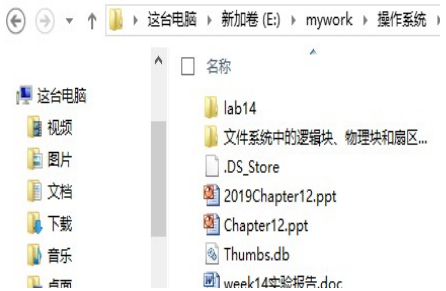
2.On a shared system describe user access control (存取控制)

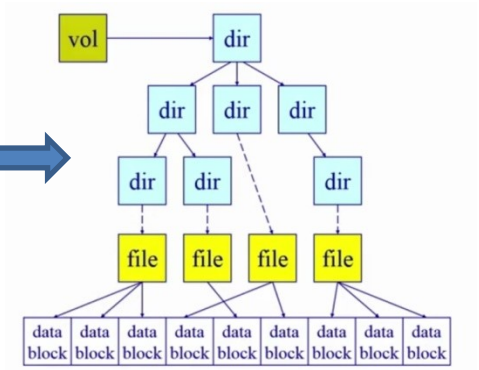
3.Blocking (块化) for access to files

4.Manage free blocks

- Allocate free blocks to files (空闲块分配)
- Reclaim free blocks (空闲块回收)

12.1 Overview(1/6)

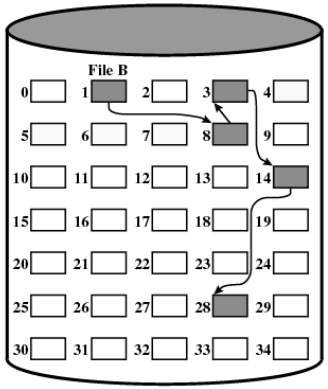




一个文件 m 块，
每块 (n 个扇区)

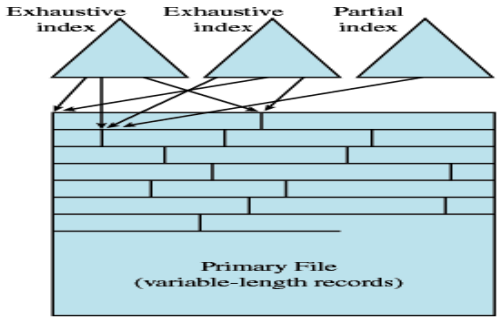
Record.xls

学号	姓名	籍贯
1001	A	SC
1002	B	BJ



File Allocation Table

File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...



(d) Indexed File

Figure 12.9 Chained Allocation

12.1 Overview(2/6)

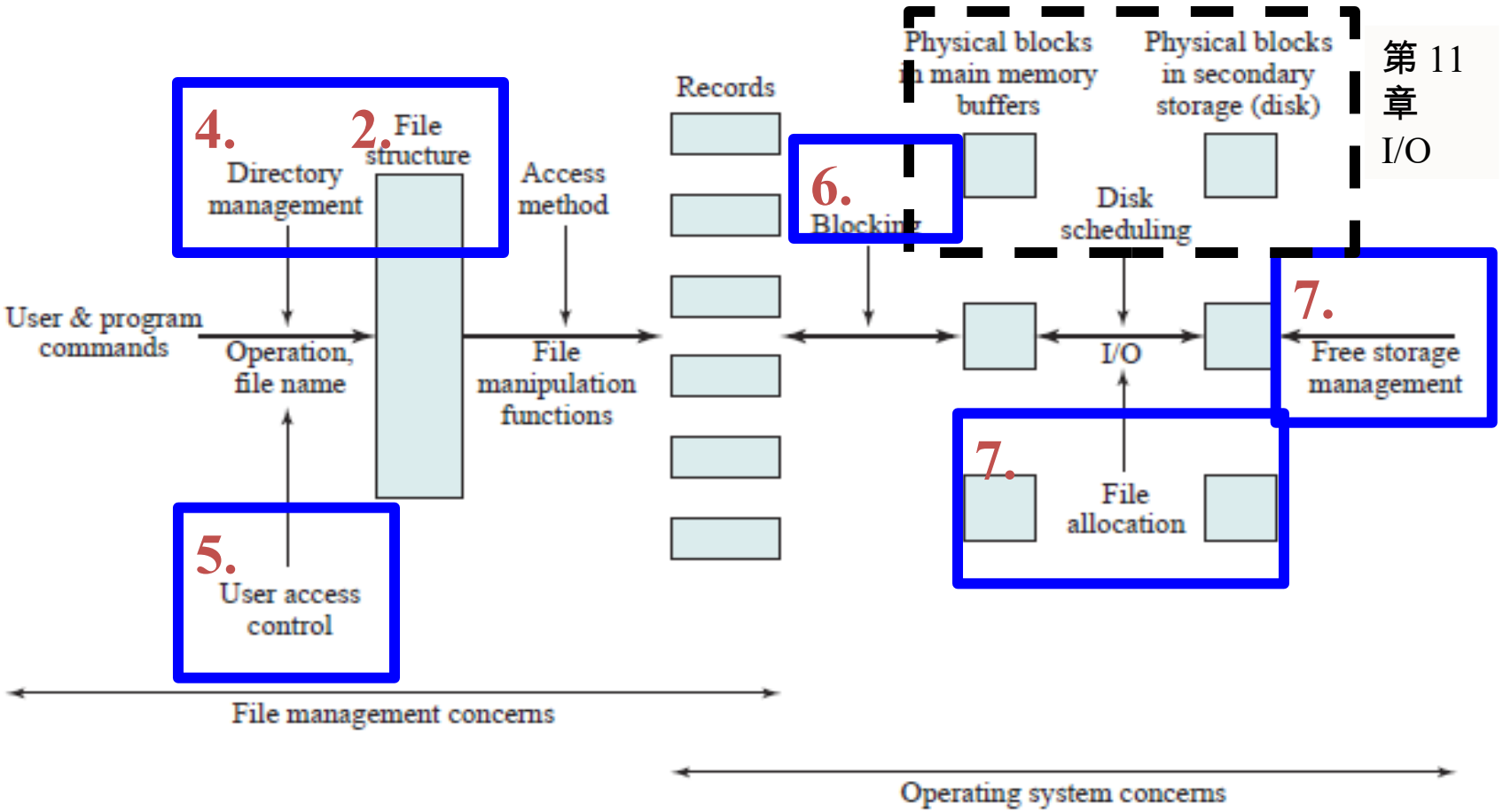
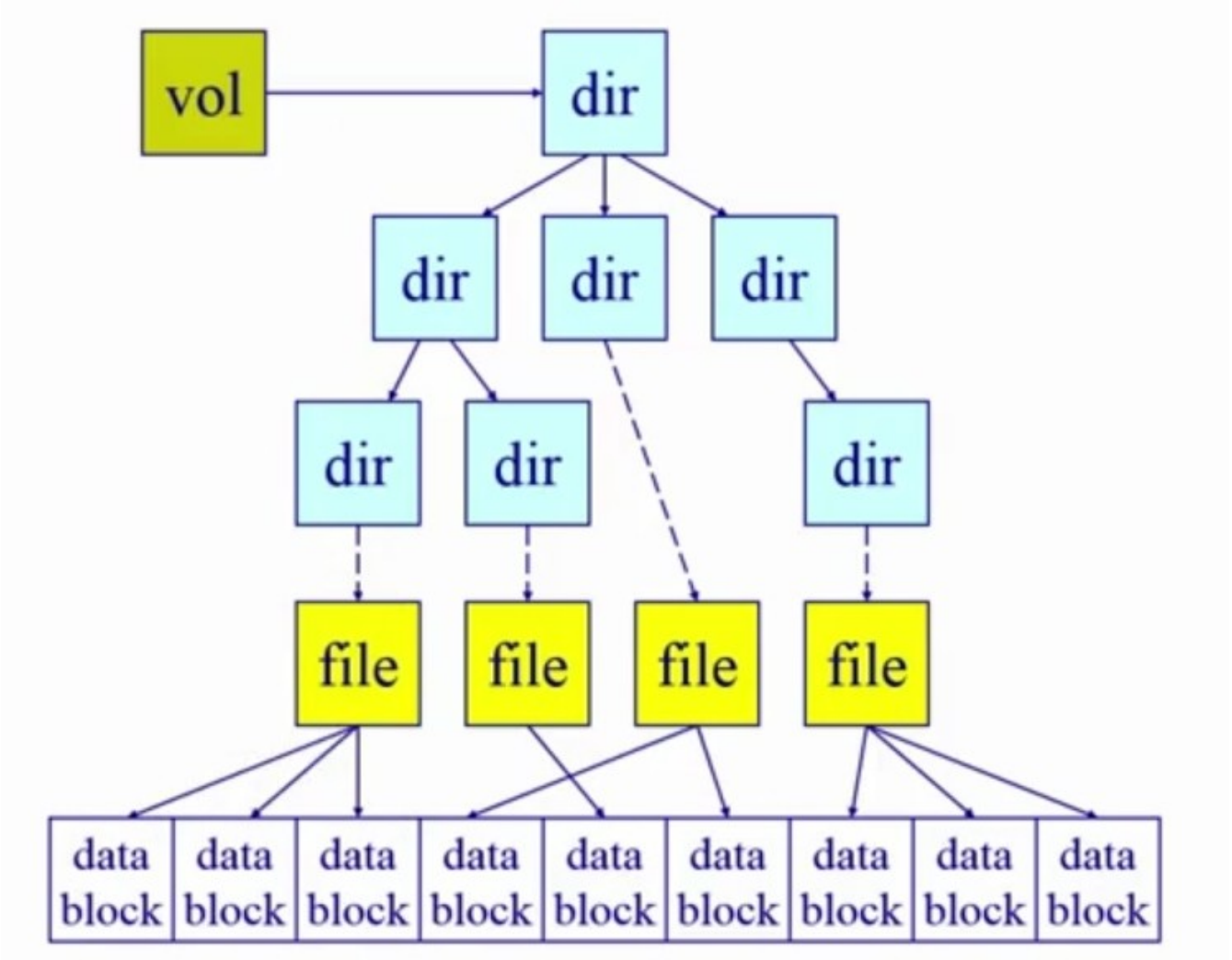


Figure 12.2 Elements of File Management

12.1 Overview(4/6)

- Unix



Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.2 File Organization and Access

- 12.2.1 Criteria for File Organization
- 12.2.2 5 Different File Organizations

12.2.1 Criteria for File Organization(1/1)

Criteria for File Organization (文件组织评价标准)

1.Short access time (短的存取时间)

2.Ease of update (易于修改)

3.Economy of storage (存储经济性)

4.Simple maintenance (维护简单)

5.Reliability (可靠性)

12.2 File Organization and Access

- 12.2.1 Criteria for File Organization
- 12.2.2 5 Different File Organizations
  -

12.2.2 5 Different File Organizations(1/11)

- File Operations(文件操作)
  - Create/Delete/Open/Close/Read/Write
- Terms Used with Files (文件术语)
  - Field (域)<Record (记录)<File (文件) < Database (数据库)

- File Organizations
  - 文件的存储结构是指文件在外存上的组织方式

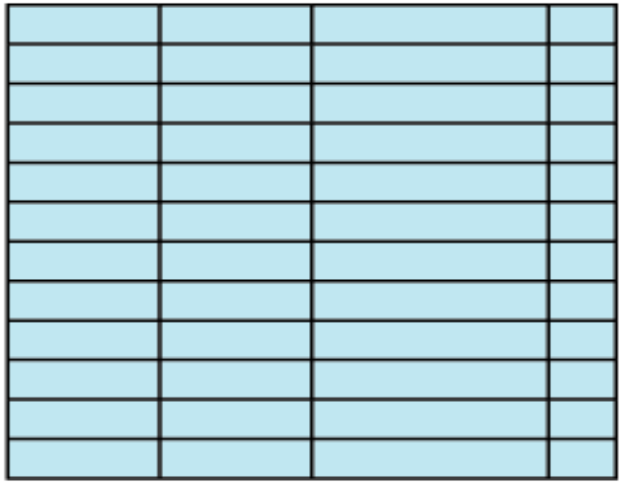
12.2.2 5 Different File Organizations (2/11)

2、 The Sequential File (顺序文件)

1. All fields the same (order and length)
   - Field names and lengths are attributes of the file
2. The First (only one) field is the key filed (关键域 / 关键字)
   - Uniquely identifies the record
   - Records are stored in key sequence
3. New records are placed in a log file(日志文件) or transaction file(事务文件) (the pile 堆)
4. Batch update (成批更新) is performed to merge(合并) the log file with the master file.

12.2.2 5 Different File Organizations (3/11)

- 顺序文件多用于磁带。
  - 一切存储在顺序存储器 (磁带) 上的文件都只能顺序文件 。只能按顺序查找法存取。
  - 存储在 直接存取存储器 (磁盘) 上的顺序文件可以顺序查找法存取，也可以用分块查找法或二分查找法存取。



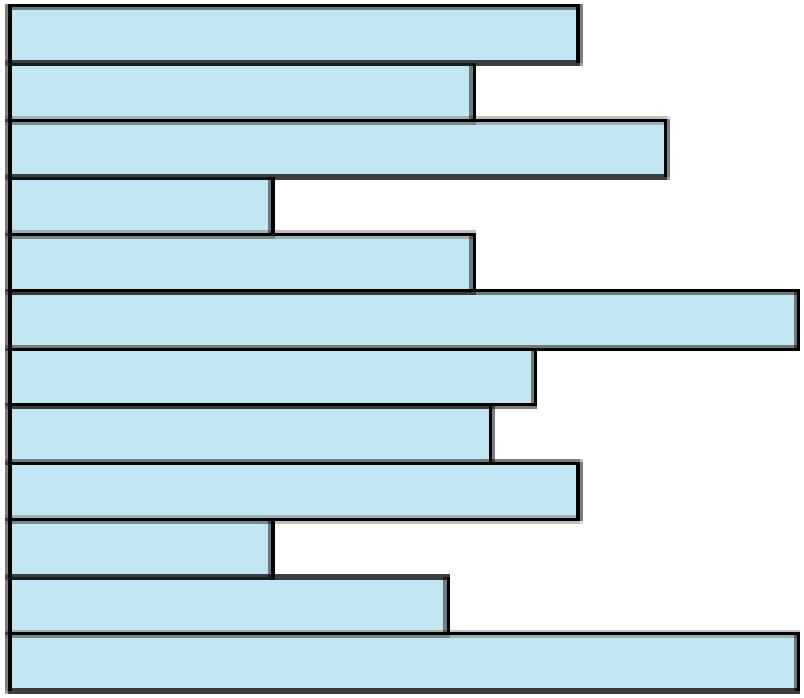
Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

(b) Sequential File

12.2.2 5 Different File Organizations(4/11)

- 1、 The Pile 堆
  - Data are collected in the order they arrive
  - Purpose is to accumulate a mass of data and save it
  - Records may have different fields
  - Disadvantages:
    - No structure
    - Record access is by exhaustive search (穷举搜索)

12.2.2 5 Different File Organizations(5/11)



Variable-length records
Variable set of fields
Chronological order

(a) Pile File

12.2.2 5 Different File Organizations (6/11)

- 3、 The Indexed Sequential File (索引顺序文件)
  - Sequential file + index + overflow file
  - Index: quickly reach the vicinity 邻近 of the desired record (索引提供了快速接近目标记录的查询能力)
    - Index contains a key field and a pointer to the main file
    - Indexed is searched to find highest key value that is equal to or precedes the desired key value (索引查找关键字小于或者等于目标关键字的最大记录)
  - Multiple level indexes (多级索引) for the same key field can be set up to increase efficiency
    - 1 级索引： 100 万条记录，查找某记录平均时间 50 万条
    - 2 级索引： 1 级 1000 条， 2 级 1000 条，则时间为 500+500

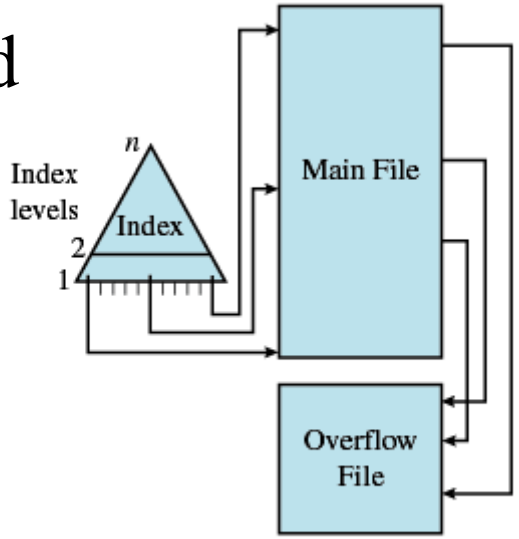
12.2.2 5 Different File Organizations(7/11)

- Overflow File of The Indexed Sequential File
  - New records are added to an overflow file(溢出文件)
  - Record in main file that precedes it is updated to contain a pointer to the new record
  - The overflow is merged (合并) with the main file during a batch update

12.2.2 5 Different File Organizations (8/11)

Disadvantage:

Based on a single field

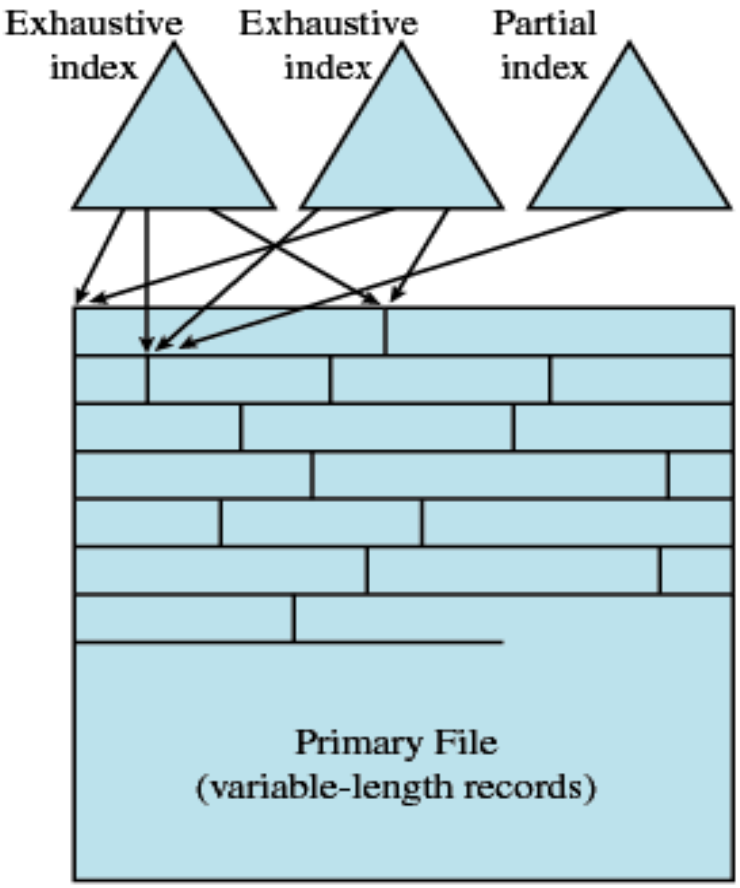


(c) Indexed Sequential File

12.2.2 5 Different File Organizations(9/11)

- 4、 The Indexed File (索引文件)
  - Uses multiple indexes for different key fields
  - May contain an exhaustive index(完全索引) that contains one entry for every record in the main file
  - May contain a partial index(部分索引)
  - Index itself are sequential file 索引是顺序的
    - When a new record is added to the main file, all of the index files must beupdated.
  - Records no longer no restrict to be sequential 记录非顺序
  - Pointer in one index refers to that record 用索引查找记录

12.2.2 5 Different File Organizations(10/11)



(d) Indexed File

12.2.2 5 Different File Organizations (11/11)

- 5、 The Direct or Hashed File(直接文件或者散列文件)
  - Key field required for each record
  - Hash based on the key field
    - 根据文件中关键字的特点，设计一个散列函数和处理冲突的方法，将记录散列到存储设备上。
  - 优点：文件随机存放，记录不需要排序；插入删除方便；存取速度快；不需要索引区，节省存储空间。
  - 缺点是：不能进行顺序存取，只能按关键字随机存取

Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.4 File Directories(1/9)

Contents ：

- Directory itself is a file owned by OS(一个目录本身是一个文件)
- Provides mapping between file names and the files themselves(提供文件名和文件之间的映射)

- Information Elements of a File Directory Table12.1
  - Basic Information
  - Address Information
  - Access Control Information
  - Usage Information

12.4 File Directories(2/9)

Structure ：

- File Directories
  - Contain ： Directory entry 目录表项
  - Operations ：
    - Search
    - Create file
    - Delete file
    - List directory
    - Update directory

12.4 File Directories(3/9)

- Structure 1 ： Simple Structure for a Directory (简单目录结构) ：目录项列表
  - Represented by a simple sequential file with the name of the file serving as the key (用顺序文件代表目录，该目录下的文件名做该顺序文件的关键字)
  - Forces user to be careful not to use the same name for two different files (文件不能重名)

12.4 File Directories(4/9)

- Structure 2 ： Two-level Scheme for a Directory(两级目录方案)
  - A master directory (主目录)+One directory for each user (用户目录)
  - Master directory contains entry for each user
    - Provides address and access control information
  - Each user directory is a simple list of files for that user
  - Still provides no help in structuring collections of files （ 不能建子目录 ）

12.4 File Directories(5/9)

- Structure 3 ： Hierarchical, or Tree-Structured Directory (层次 / 树状结构目录)
  - Master directory with user directories underneath it
  - Each user directory may have subdirectories and files as entries

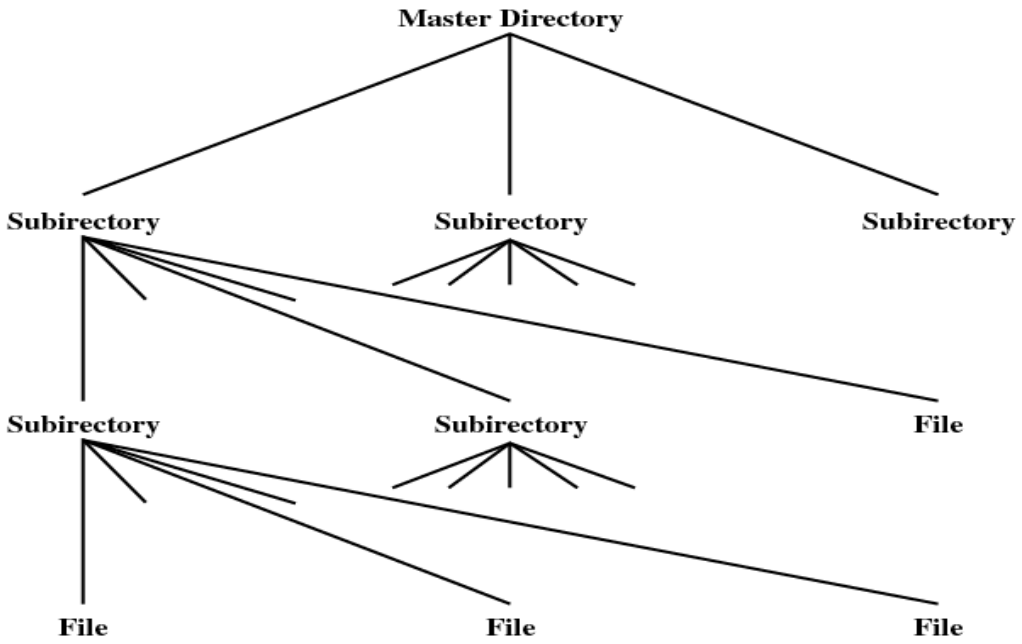


Figure 12.4 Tree-Structured Directory

12.4 File Directories(6/9)

- In Hierarchical, or Tree-Structured Directory
  - Files can be located (文件定位) by following a path from the root, or master, directory down various branches
    - This is the **pathname for the file**
  - Can have several files with the same file name (文件同名) as long as they have **unique path** names

12.4 File Directories(7/9)

- userB
  - Word/Unit_A/ABC
  - Draw/Unit_A/ABC

- 每个磁盘根目录位置固定

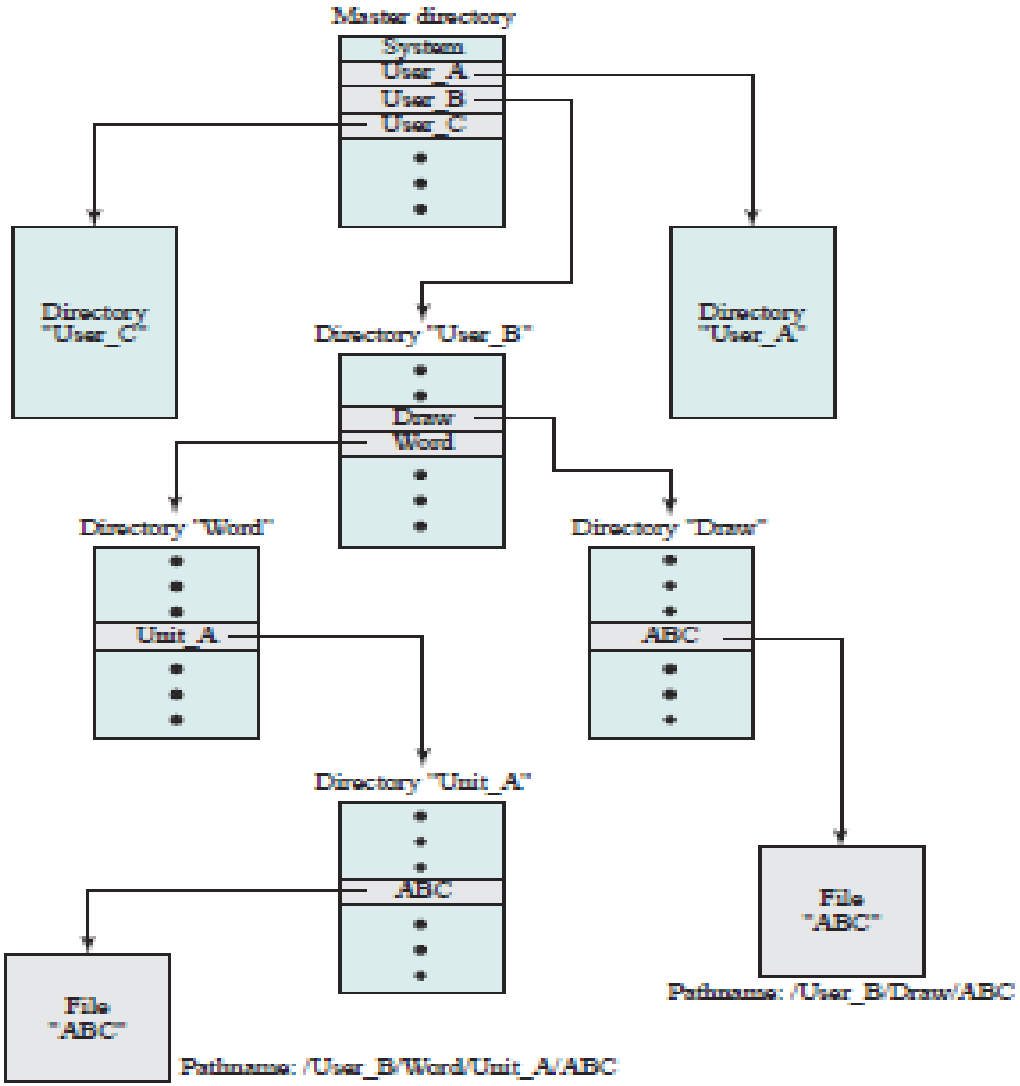


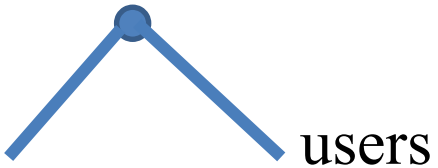
Figure 12.7 Example of Tree-Structured Directory

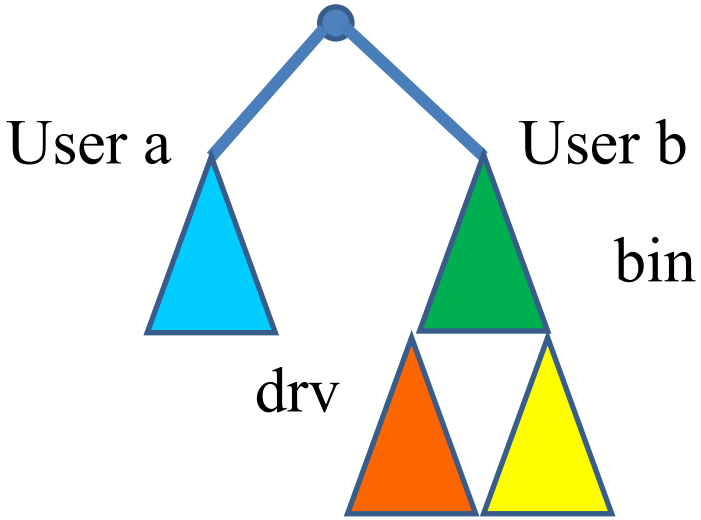
12.4 File Directories(8/9)

- Effectively locate the file
  - Current directory is the working directory (当前工作目录)
    - Files are referenced relative to the working directory (相对路径)
    - 每个进程都会指向一个文件目录，用于解析文件名

12.4 File Directories(9/9)

- 文件系统挂载：启动时挂入根节点





Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.5 File Sharing

- In multiuser system, allow files to be shared among users
- Two issues
  - Access rights (存取权限)
  - Management of simultaneous access （同时存取控制)

12.5 File Sharing

Access Rights

- None(无)
  - User may not know of the existence of the file
  - User is not allowed to read the user directory that includes the file
- Knowledge(知道)
  - User can only determine that the file exists and who its owner is

12.5 File Sharing

Access Rights

- Execution(执行)
  - The user can load and execute a program but cannot copy it
- Reading(读)
  - The user can read the file for any purpose, including copying and execution
- Appending(追加)
  - The user can add data to the file but cannot modify or delete any of the file's contents

12.5 File Sharing

Access Rights

- Updating(更新)
  - The user can modify, deleted, and add to the file's data. This includes creating the file, rewriting it, and removing all or part of the data
- Changing protection(更改保护)
  - User can **change access rights** granted to other users
- Deletion(删除)
  - User can delete the file

12.5 File Sharing

Access Rights

- Owners （所有者）
  - Has all rights previously listed
  - May grant rights to others using the following classes of users
    - Specific user(指定用户)
    - User groups(用户组)
    - All for public files(所有用户)

12.5 File Sharing

- Simultaneous Access(同时访问)
  - User may lock entire file when it is to be updated
  - User may lock the individual records during the update
  - Mutual exclusion and deadlock are issues for shared access

Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.6 Record Blocking(1/3) 记录组块

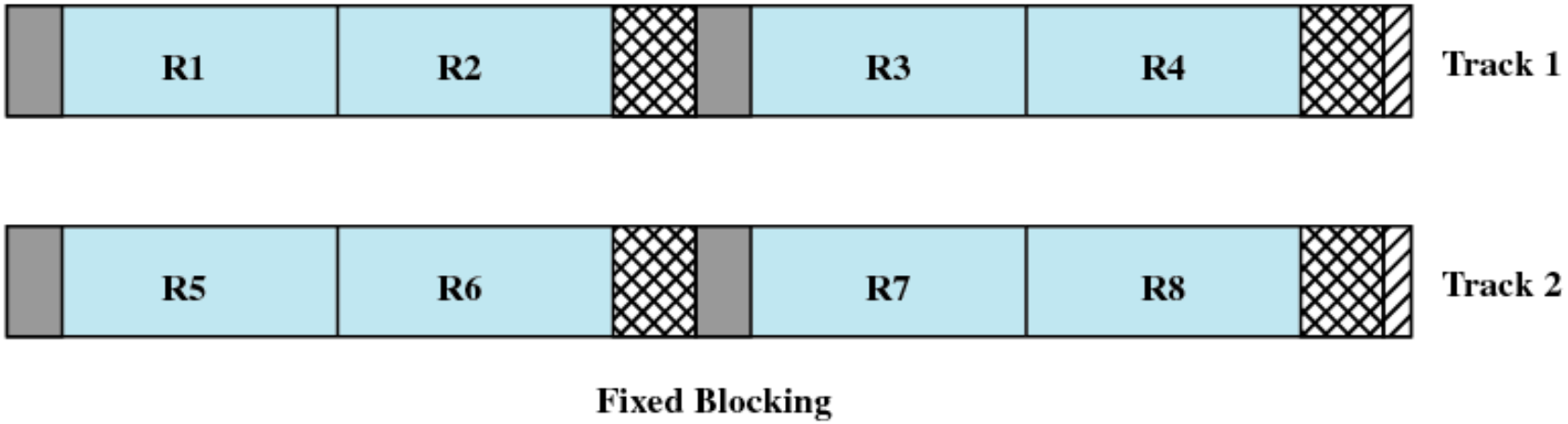
- 扇区 (sector):
  - 硬件 (磁盘) 上的最小的操作单位 , 是操作系统和块设备 (硬件、磁盘) 之间传递单位

- 逻辑块 Block ：
  - OS 的虚拟文件系统从硬件设备上读取一个 block ，实际为从硬件设备读取一个或多个 sector 。

- 对于文件管理来说，每个文件对应的多个 block 可能是不连续的 ;block 最终要映射到 sector 上，所以 block 的大小一般是 sector 的整数倍。不同的文件系统 block 可使用不同的大小，操作系统会在内存中开辟内存，存放 block 到所谓的 block buffer 中。

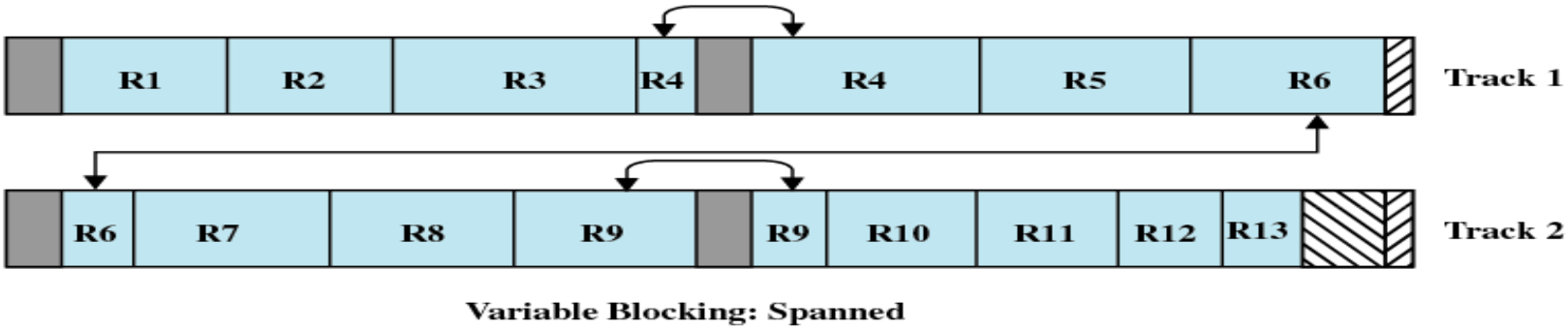
12.6 Record Blocking(2/3)

定长组块



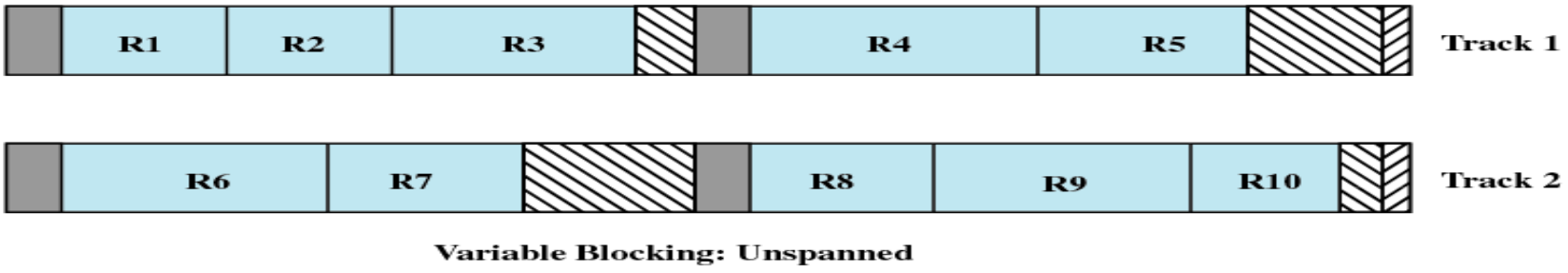
12.6 Record Blocking(3/3)

变长跨越式组块

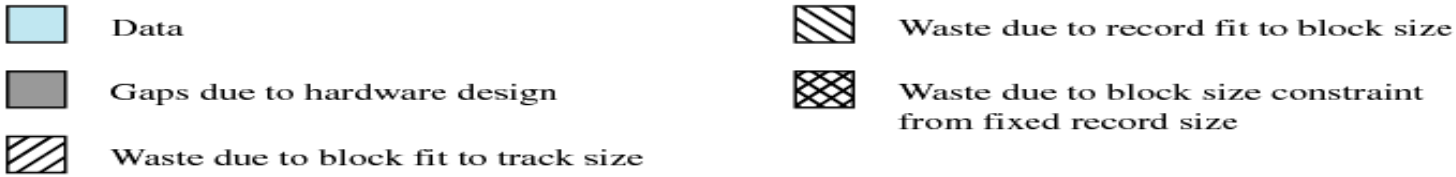


Variable Blocking: Spanned

变长非跨越式组块



Variable Blocking: Unspanned



Agenda

- 12.1 Overview
- 12.2 File Organization and Access
- 12.4 File Directories
- 12.5 File Sharing
- 12.6 Record Blocking
- 12.7 Secondary Storage Management
- 12.8 Summary

12.7 Secondary Storage Management (辅助存储管理)

- 12.7.1 File Allocation
- 拓展： File System in OpenEuler
- 12.7.2 Free Space Management
- 12.7.3 Reliability

12.7.1 File Allocation(1/13)

- portions (文件) 分区：
  - Space is allocated to a file as one or more contiguous units, which we shall refer to as portions. That is, a **portion** is a contiguous set of allocated blocks.
  - Size of portion ： variable size or fixed size (block)
- to keep track of the portions assigned to a file
  - Using FAT (File Allocation Table)

12.7.1 File Allocation(2/13)

- Preallocation VS Dynamic
  - Preallocation(预分配)
    - Need the maximum size for the file at the time of creation
    - Difficult to reliably estimate the maximum potential size of the file
  - Dynamic allocationallocation (动态分配)
    - Allocates space to a file in portions as needed.

12.7.1 File Allocation(3/13)

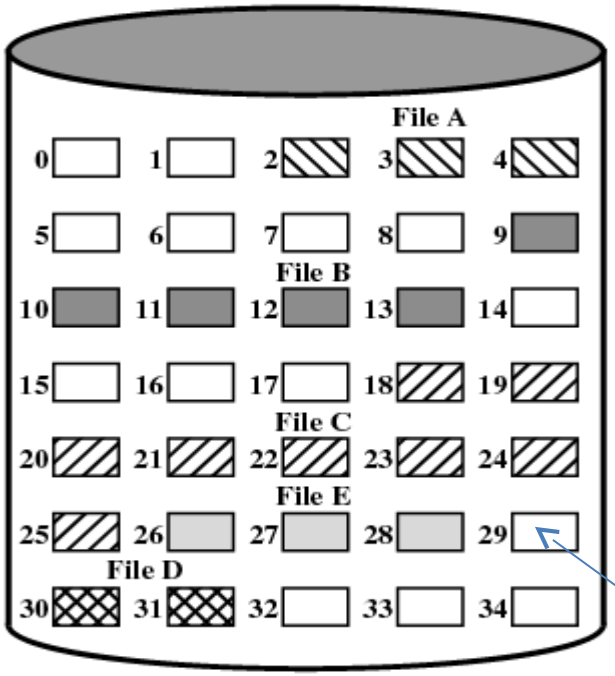
- Different allocation strategies (分配策略)
  1. Contiguous allocation (连续分配)
  2. Chained allocation (链式分配)
  3. Indexed allocation (索引分配)

12.7.1 File Allocation(4/13)

- Contiguous allocation (连续分配)
  - Single set of blocks is allocated to a file at the time of creation (文件创建时分配一组连续的块)
  - Only a single entry in the file allocation table
    - Starting block and length of the file
  - External fragmentation will occur
    - Need to perform compaction

12.7.1 File Allocation(5/13)

Contiguous allocation



File Allocation Table

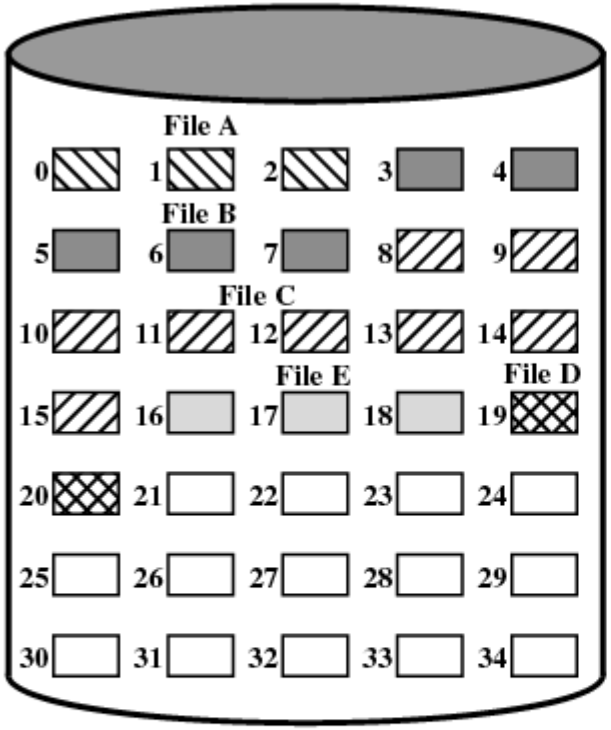
File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

外部碎片

Figure 12.7 Contiguous File Allocation

12.7.1 File Allocation(6/13)

Contiguous allocation



File Allocation Table

File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

Figure 12.8 Contiguous File Allocation (After Compaction)

12.7.1 File Allocation(7/13)

- Chained allocation (链式分配)
  - Allocation on basis of individual block(基于单个块进行分配)
  - Each block contains a pointer to the next block in the chain
  - Only single entry in the file allocation table
    - Starting block and length of file
  - No external fragmentation
  - Best for sequential files (适合顺序文件)
  - No accommodation of the principle of locality (局部性原理不再适用) ， so consolidation(迁移、集结) is need to move blocks adjacent each other

12.7.1 File Allocation(8/13)

Chained allocation

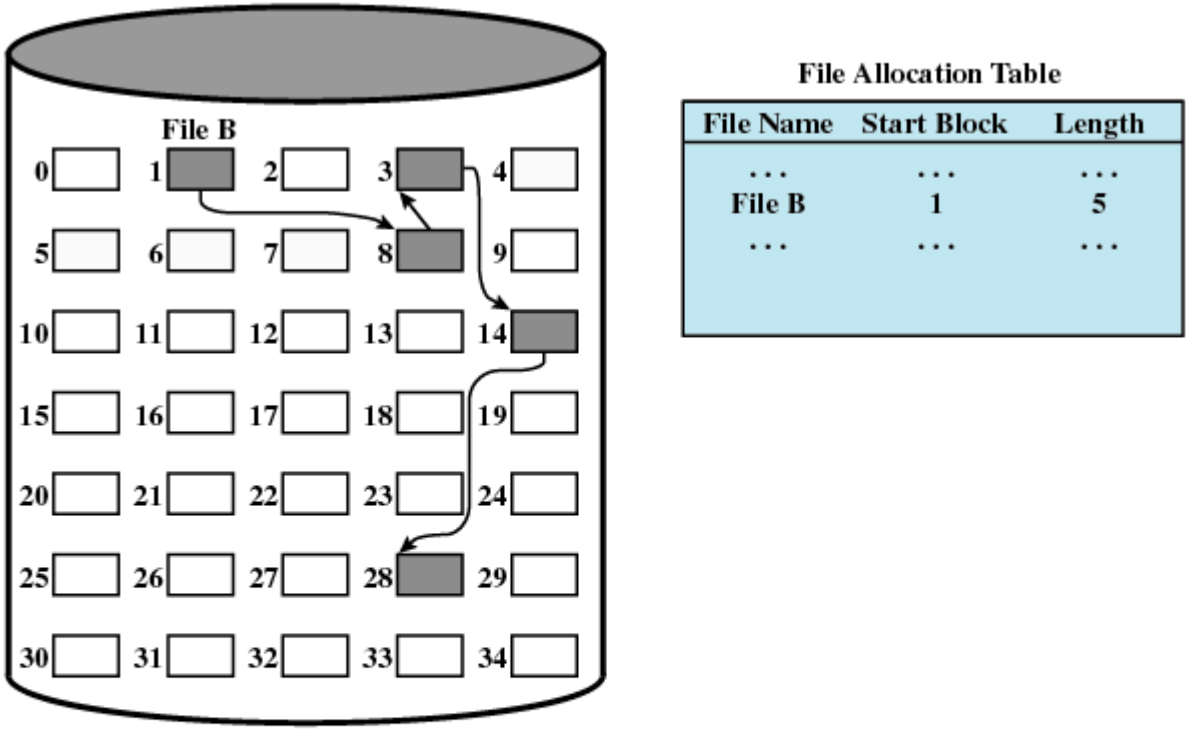
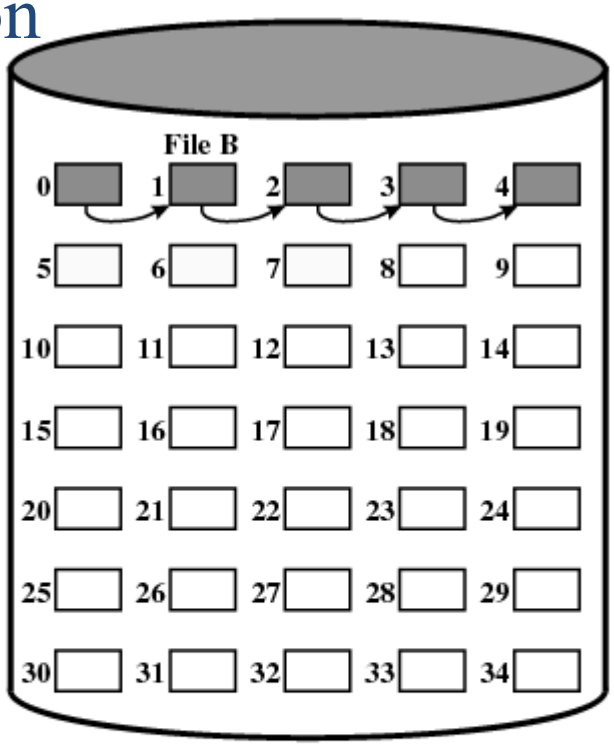


Figure 12.9 Chained Allocation

12.7.1 File Allocation(9/13)

Chained allocation



File Allocation Table

File Name	Start Block	Length
...	...	...
File B	0	5
...	...	...

合并后

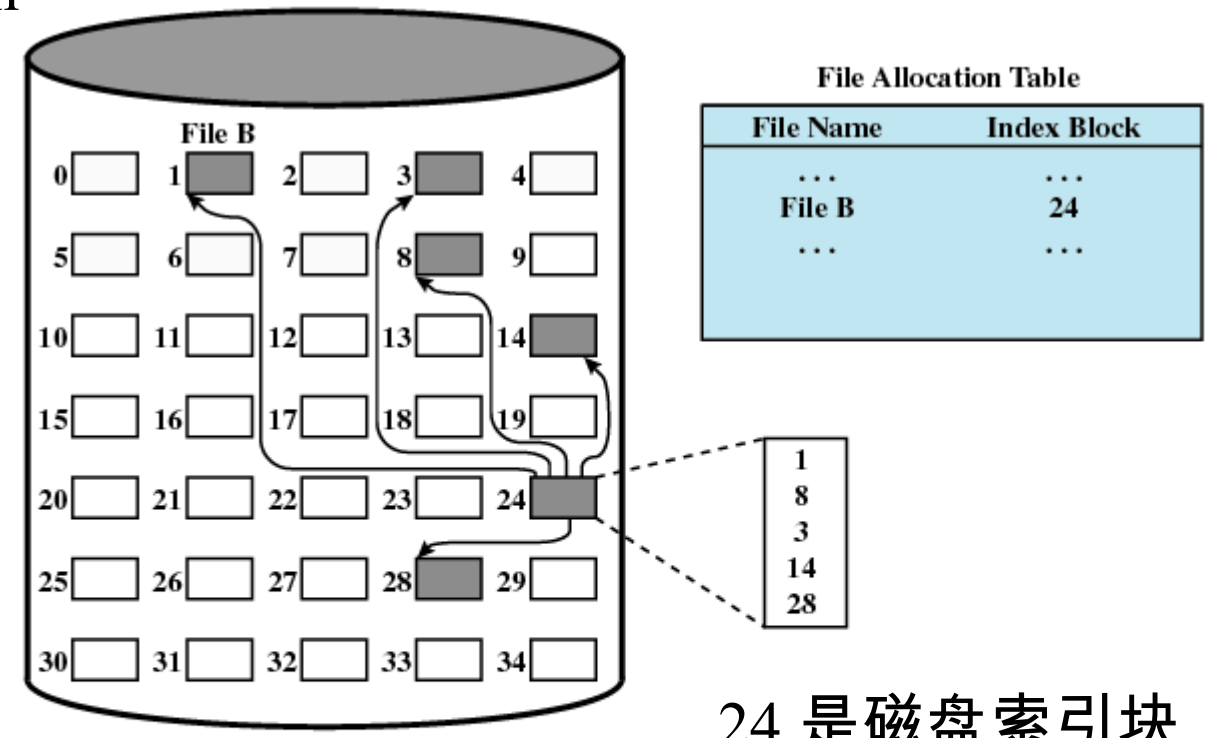
Figure 12.10 Chained Allocation (After Consolidation)

12.7.1 File Allocation(10/13)

- Indexed allocation (索引分配)
  - File allocation table contains a separate one-level index for each file (每个文件在文件分配表中有一个一级索引)
  - The file allocation table contains block number for the index (文件分配表指向该文件在磁盘上的索引块)
  - The index has one entry for each portion allocated to the file (分配给文件的每个分区都在索引中都有一个表项)
  - Indexed allocation supports both sequential and direct access to the file and thus is the most popular form of file allocation.

12.7.1 File Allocation(11/13)

Indexed allocation



24 是磁盘索引块和数据块一样大

Figure 12.11 Indexed Allocation with Block Portions

12.7.1 File Allocation(12/13)

变长 索引分配

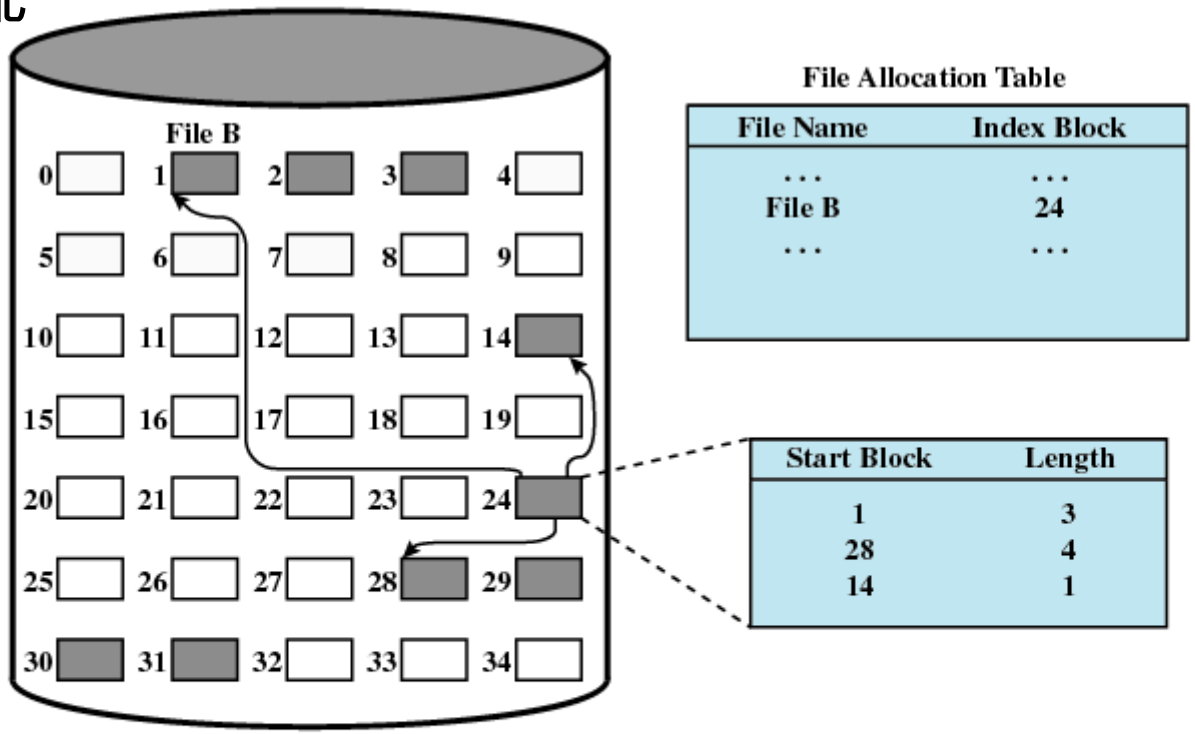
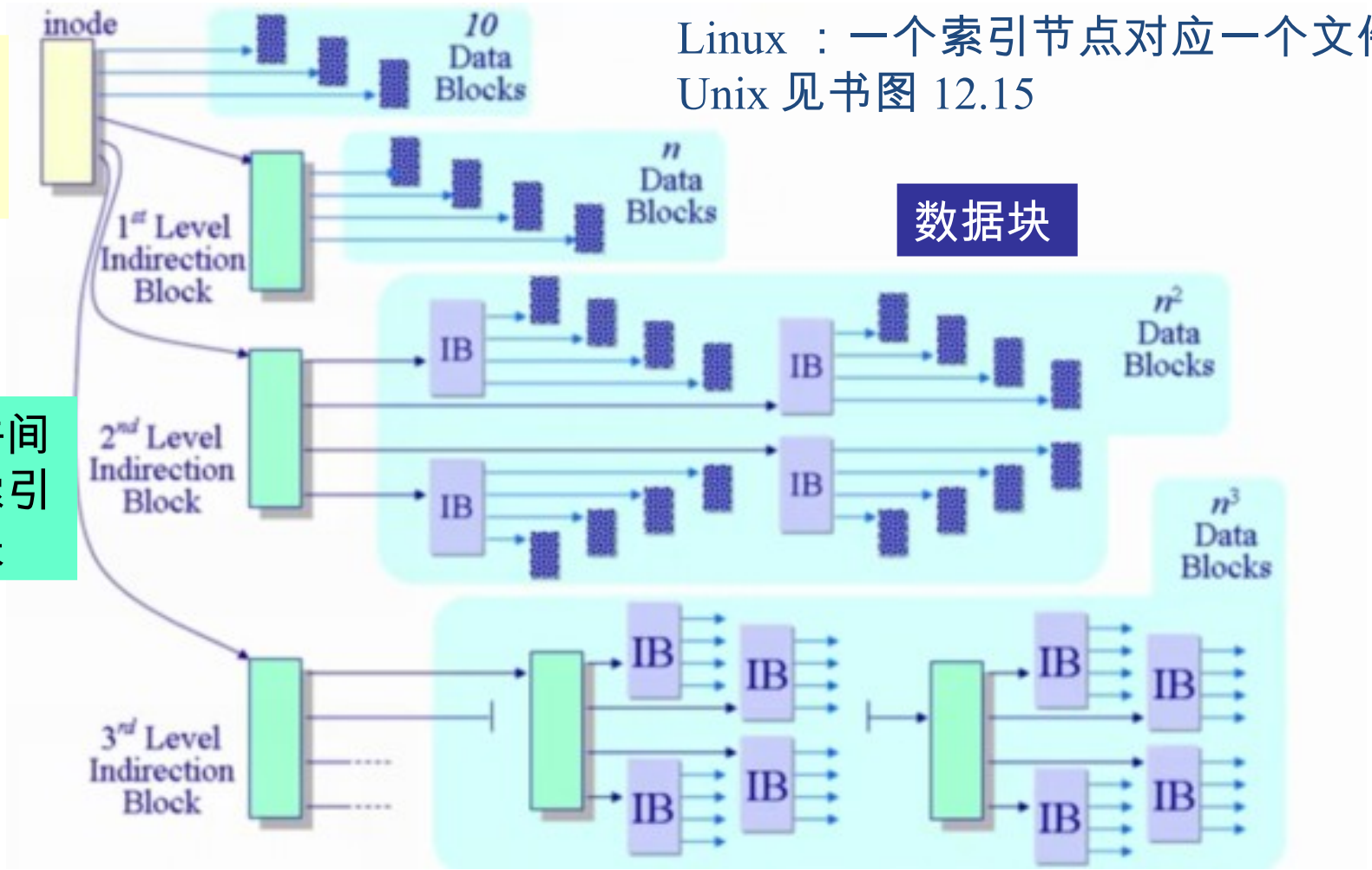


Figure 12.12 Indexed Allocation with Variable-Length Portions

12.7.1 File Allocation(13/13)

文件索引节点

文件间接索引节块



Linux ：一个索引节点对应一个文件
Unix 见书图 12.15

数据块

拓展： File System in OpenEuler(1/4)

- OpenEuler 文件系统架构
  - 进程位于架构上方，仅与 VFS(Virtual File System) 交互
  - VFS 抽象不同文件系统的行为，提供统一通用 API ，进行打开读取，写入等操作。是用户可见的目录树。
  - 现实层默认选用 Ext4 文件系统 Fourth Extended File System

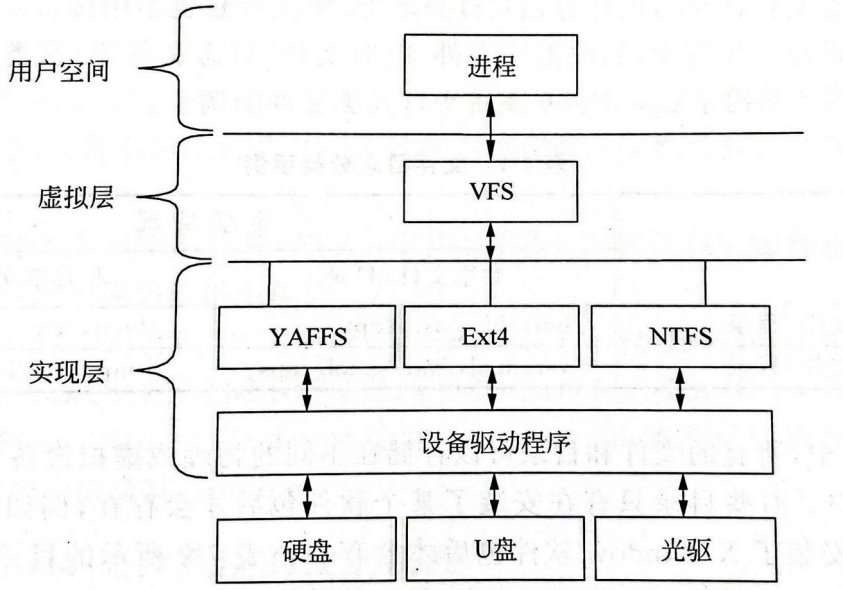


图 7-4　openEuler 中的文件系统架构

拓展： File System in OpenEuler (2/4)

- Ext4 文件系统布局
  - 每个分区可以安装不同文件系统
  - 安装操作系统的分区中的组块 0 中含有引导块，其它组块不含。
  -

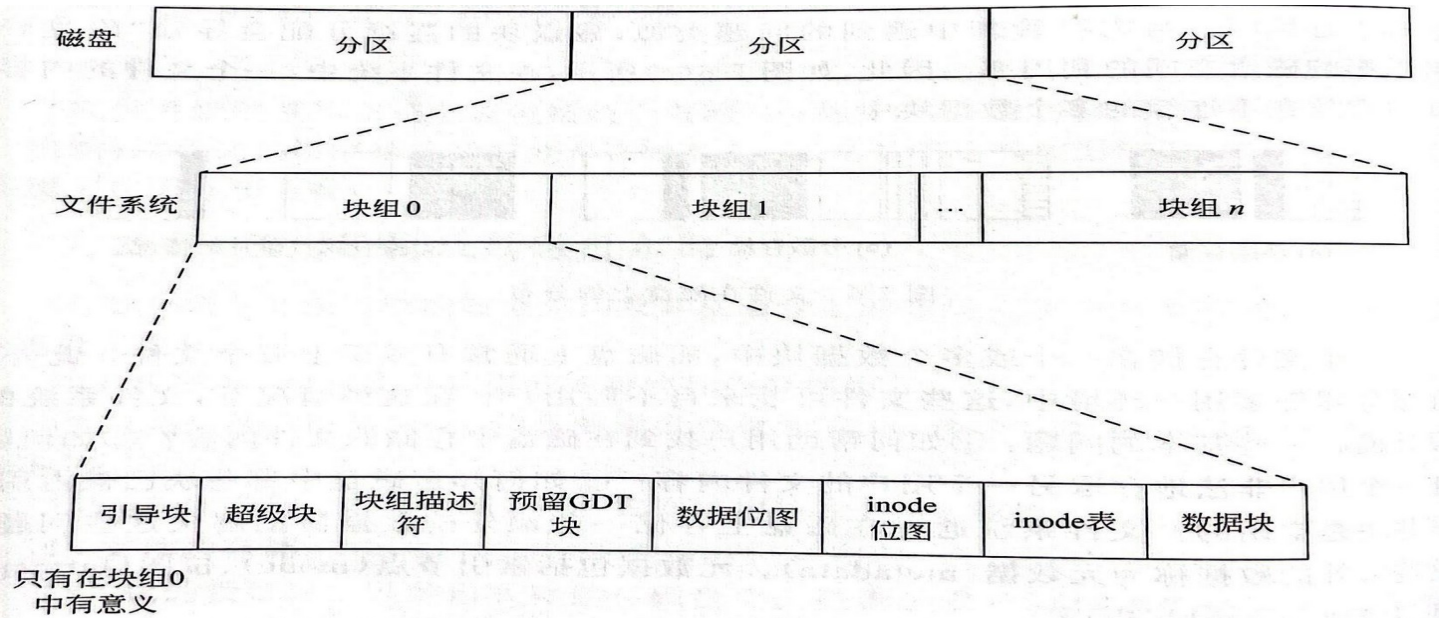


图 7-8　Ext4 文件系统的整体布局

拓展： File System in OpenEuler (3/4)

- Ext4 文件系统布局
  - 超级块大小 1KB, 含记录文件系统整体层面的数据结构

```
1.    //源文件：fs/ext4/ext4.h
2.    struct Ext4_super_block {
3.        __le32  s_inode_count;          /* inode 总数 */
4.        __le32  s_blocks_count_lo;      /* 以块为单位的文件系统的大小 */
5.        __le32  s_free_blocks_count_lo; /* 空闲块计数 */
6.        __le32  s_free_inode_count;     /* 空闲 inode 计数 */
7.        __le32  s_log_block_size;       /* 块的大小 */
8.        __le32  s_mtime;                /* 文件系统最后一次启动的时间 */
9.        __le32  s_wtime;                /* 上一次写操作的时间 */
10.       __le32  s_creator_os;           /* 创建文件系统的操作系统 */
11.       __le16  s_magic;                /* 文件系统魔术数,代表其类型 */
12.       __le16  s_state;                /* 文件系统的状态 */
13.       ...
14.   }
```

图 7-9　超级块的部分成员

拓展： File System in OpenEuler (4/4)

- Ext4 文件系统布局
  - Inode 索引节点的数据结构记录文件的信息

```
1.    //源文件：fs/ext4/ext4.h
2.    struct ext4_indoe{
3.        __le16  i_mode;                 /* 文件类型和访问权限 */
4.        __le16  i_uid;                  /* 文件所有者的标识符 */
5.        __le32  i_size_lo;              /* 以字节为单位的文件大小 */
6.        __le32  i_atime;                /* 上一次访问时间 */
7.        __le32  i_ctime;                /* 上一次 inode 改动时间 */
8.        __le32  i_mtime;                /* 上一次文件修改时间 */
9.        __le32  i_dtime;                /* 文件删除的时间 */
10.       __le16  i_links_count;          /* 链接数 */
11.       __le32  i_block[Ext4_N_BLOCKS]; /* 指向数据块 */
12.       ...
13.       __le32  i_size_high             /* 以字节为单位的文件大小 */
14.   }
```

图 7-10　Ext4 文件系统中 inode 的数据结构

12.7 Secondary Storage Management

- 12.7.1 File Allocation
- 拓展： File System in OpenEuler
- 12.7.2 Free Space Management
- 12.7.3 Reliability

12.7.2 Free Space Management(1/3)

- 1 、 Bit tables (位表)
  - Use a vector containing one bit for each block on the disk. Each entry of a 0 corresponds to a free block, and each 1 corresponds to a block in use.

$$\frac{\text{disk size in bytes}}{8 \times \text{file system block size}}$$

    - Thus, for a 16-Gbyte disk with 512-byte blocks, the bit table occupies about 4 Mbytes.
- Improve Efficiency :Auxiliary data structures that summarize the contents of subranges of the bit table

12.7.2 Free Space Management(2/3)

- 2、 Chained free portions (链式空闲区)
  - The free portions may be chained together by using a pointer and length value in each free portion
  - negligible space overhead: a pointer to the beginning of the chain and the length of the first portion
- 3、 Indexing (空闲索引表)
  - The indexing approach treats free space as a file and uses an index table as described under file allocation.
- 4、 Free block list (空闲列表)
  - Assign a number to each blocks
  - Maintain the list of the numbers of all free blocks
  - Cache part of the list in memory

12.7.2 Free Space Management(3/3)

- Allocate strategies:
  - **First fit:** Choose the first unused contiguous group of blocks of sufficient size from a free block list.
  - **Best fit:** Choose the smallest unused group that is of sufficient size.
  - **Nearest fit:** Choose the unused group of sufficient size that is closest to the previous allocation for the file to increase locality.

12.7 Secondary Storage Management

- 12.7.1 File Allocation
- 拓展： File System in OpenEuler
- 12.7.2 Free Space Management
- 12.7.3 Reliability

12.7.3 Reliability(1/1)

- Use a lock to prevent interfere among processes and make sure of consistent of space allocation