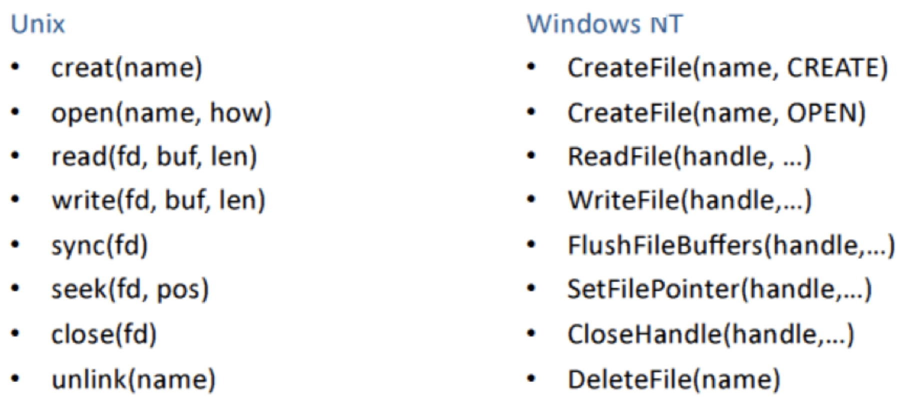
11 File System

- File Concept
  - What Are File Systems
    - Everything is stored as files in a computer system. The files can be data files or application files.
      - 一切都以文件的形式存储在计算机系统中。文件可以是数据文件或应用程序文件。
    - A file is a named, linear region of bytes that can grow and shrink.
      - 文件是一个命名的、线性的字节区域，它可以增长和缩小。
    - The operating system performs this management with the help of a program called File System.
      - 操作系统在一个叫做文件系统的程序的帮助下执行这种管理。
    - Different operating systems use different file systems.
      - 不同的操作系统使用不同的文件系统。
  - File System
    - FILE SYSTEM INTERFACE
      - The user level (more visible) of the file system.
        - 文件系统的用户级别(更可见)。
        - • Access methods
        - • Directory Structure
        - • Protection
        - • File-System Mounting
        - • File Sharing
          - •访问方法
          - •目录结构
          - •保护
          - •文件系统挂载
          - •文件共享
      - FILE SYSTEM IMPLEMENTATION
        - 文件系统实现
        - The OS level (less visible) of the file system.
        - • Allocation and Free Space Management
        - • Directory Implementation
          - 文件系统的操作系统级别(不太可见)。

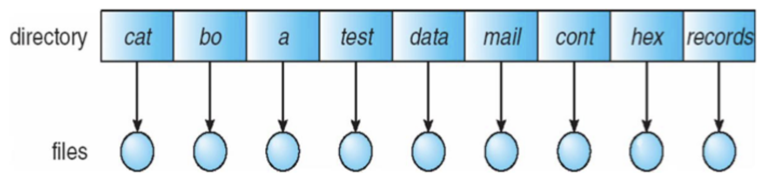
        - •分配和空闲空间管理
        - •目录实现
    - Basic File Operations
        - 

Unix	Windows NT
• creat(name)	• CreateFile(name, CREATE)
• open(name, how)	• CreateFile(name, OPEN)
• read(fd, buf, len)	• ReadFile(handle, …)
• write(fd, buf, len)	• WriteFile(handle,…)
• sync(fd)	• FlushFileBuffers(handle,…)
• seek(fd, pos)	• SetFilePointer(handle,…)
• close(fd)	• CloseHandle(handle,…)
• unlink(name)	• DeleteFile(name)

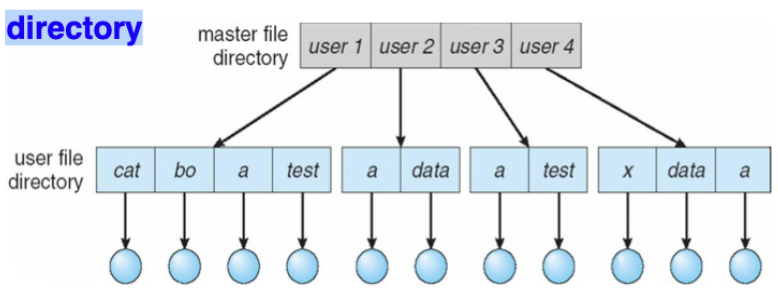
        - File systems break files down into two logical categories:
            - 文件系统将文件分为两个逻辑类别:
            - ☐ Shareable vs. Unsharable files
            - ☐ Variable vs. Static files
                - ☐可共享与不可共享文件
                - ☐变量文件与静态文件
        - ▪ Shareable files - can be accessed locally and by remote hosts;
            - ▪可共享的文件-可以访问本地和远程主机;
        - ▪ Unsharable files - only available locally.
            - ▪不可共享文件-仅在本地可用。
        - ▪ Variable files - can be changed at any time;
            - ▪可变文件-可以随时更改;
        - ▪ Static files - cannot be changed without an action from the system administrator (such as binaries).
            - ▪静态文件-没有系统管理员的操作(如二进制文件)不能被更改。
        - Open File Table - Since the open operation fetches the attributes of the file to be opened, the OS uses a data structure known as open file table (OFT),to keep the information of an opened file.
            - 打开文件表——由于打开操作获取要打开的文件的属性，操作系统使用一种称为打开文件表(OFT)的数据结构来保存打开文件的信息。
- Access Methods
    - When it is used, the information must be accessed and read into computer memory.
        - 当它被使用时，信息必须被存取并读入计算机存储器。
    - The information in the file can be accessed in several ways:

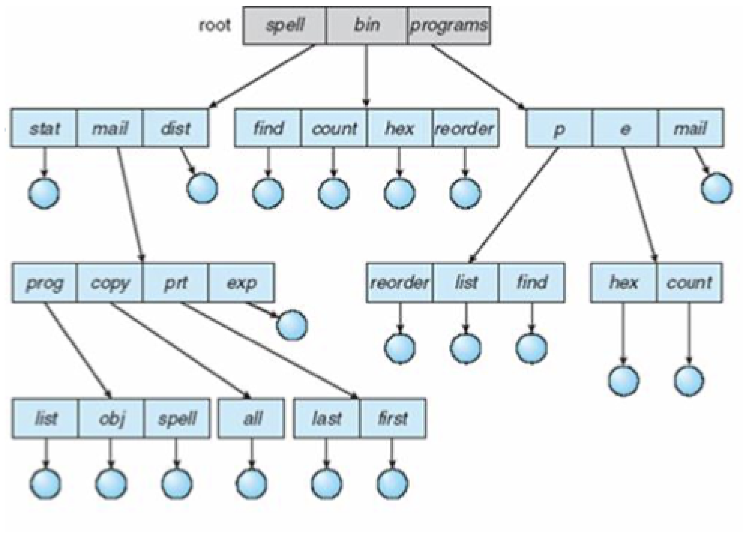
        - 文件中的信息可以通过几种方式访问
    - Sequential access
        - 顺序存取
        - Data is accessed one record right after the last
        - Reads cause a pointer to be moved ahead by one
        - Writes allocate space for the record and move the pointer to the new End Of File.
        - • Such a method is reasonable for tape.
            - 在最后一条记录之后访问数据
            - 读取导致指针向前移动1
            - 写操作为记录分配空间，并将指针移动到新的文件结束位置。
            - 这种方法对于磁带是合理的。
    - Direct access
        - 直接访问
        - Method useful for disks.
        - The file is viewed as a numbered sequence of blocks or records.
        - There are no restrictions on which blocks are read/written in any order.
            - 对磁盘有用的方法。
            - 文件被看作是块或记录的编号序列。
            - 对以任何顺序读取/写入的块没有限制。
    - Indexed access
        - ▪索引访问
        - Uses multiple indexes
        - An index block says what's in each remaining block or contains pointers to blocks containing particular items.
        - Suppose a file contains many blocks of data arranged by name alphabetically.
            - 使用多个索引
            - 索引块表示每个剩余块中的内容，或者包含指向包含特定项的块的指针。
            - 假设一个文件包含许多按名称字母顺序排列的数据块。
- Directory and Disk Structure
    - Storage Structure
        - A disk can be used in its entirety for a file system.
            - 一个磁盘可以完整地用于文件系统。
        - ▪ A disk can be broken up into multiple partitions, slices, or mini-disks, each can have its own filesystem.
            - 磁盘可以被分割成多个分区、片或迷你磁盘，每个磁盘都有自己的文件系统。

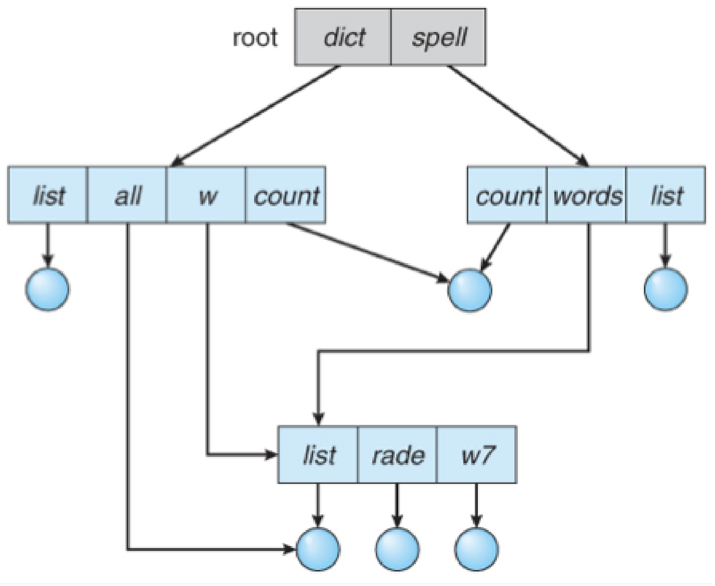
        - ▪ Disk/partition is partitioned into Blocks or Sectors
            - 磁盘/分区被划分为块或扇区
            - ▪ Modern disks have 512-byte or more sectors
            - ▪ File Systems usually work in block sizes of 4 KB
                - ▪现代磁盘有512字节或更多扇区
                - 文件系统通常以4 KB的块大小工作
- Directory
    - The directories are used to maintain the structure of a file system.
        - 目录用于维护文件系统的结构。
    - Directories serve two purposes:
        - 目录有两个用途:
    - - For User – they provide a structured way to organize files;
    - - For the File System – they provide an interface that allows the implementation to separate logical file organization from physical file placement on the disk.
        - -用户-他们提供了一个结构化的方式来组织文件;
        - -对于文件系统-它们提供了一个接口，允许实现将逻辑文件组织与磁盘上的物理文件放置分开。
- Schemes of logical structure of a directory
    - Single-Level Directory
        - root director, and all the files are stored only under it根目录下，所有文件都存储在该目录下



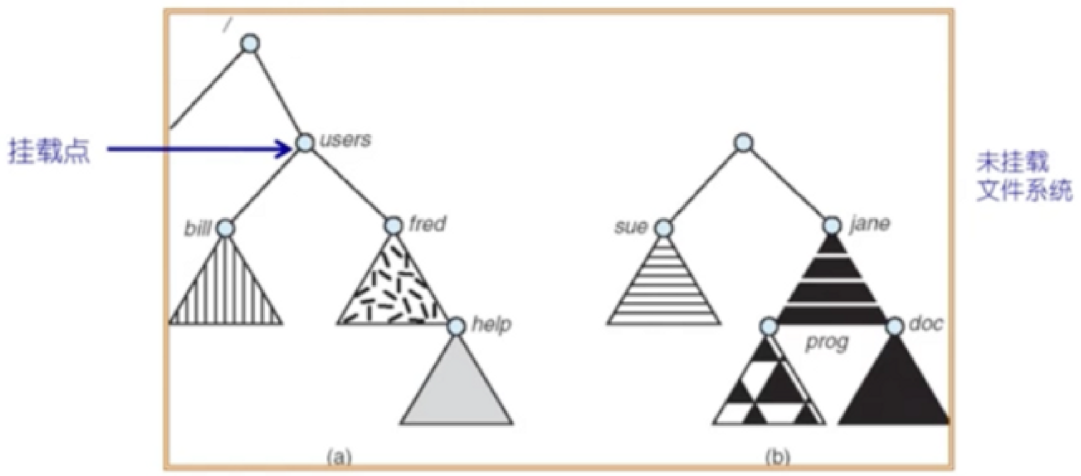
    - Two-Level Directory
        - separate directories for each user
            - 为每个用户单独设置目录
        - ▪ there are two levels: master directory and user directory
            - 有两个级别:主目录和用户目录
        -



- Hierarchical /Tree-Structured Directories
    - • absolute path - the root and followsa path down to the specified file
        - •绝对路径-根路径和后续路径到指定的文件
    - • relative path - defines a path from the current directory.
        - •相对路径-从当前目录定义一个路径。
    - 

- Acyclic-Graph Directories
    - the directory structure must allow sharing of files or sub-directories.目录结构必须允许共享文件或子目录。
    - 

- File-System Mounting
    - 每个进程都会指向一个文件目录用于解析文件名
    - 允许用户指定相对路径来代替绝对路径
    - Mounting = attaching portions of the file system into a directory structure.
        - 挂载=将文件系统的部分附加到目录结构中。

- The directory where the device is attached, is known as a mount point.
  - 连接设备的目录称为挂载点。
- Similarly, unmounting is done to remove the device from the mount point.
  - 类似地，卸载是为了将设备从挂载点移除。
- In Windows, the devices and/or partition can be accessed by opening MyComputer on the desktop.
  - 在Windows中，可以通过在桌面上打开“我的电脑”来访问这些设备和/或分区。
- 

- File Sharing
  - Sharing must be done through a protection scheme
    - 共享必须通过保护计划来实现
  - May use networking to allow file system access between systems
    - 可以使用网络来允许系统之间的文件系统访问吗
    - • Manually via programs like FTP or SSH
    - • Automatically, seamlessly using distributed file systems
    - • Semi automatically via the world wide web
      - •手动通过FTP或SSH等程序
      - •自动，无缝地使用分布式文件系统
      - •半自动通过万维网
  - Client-server model allows clients to mount remote file systems from servers
    - 客户机-服务器模型允许客户机从服务器挂载远程文件系统
    - • Server can serve multiple clients
    - • Client and user-on-client identification is insecure or complicated
    - • Network File System NFS is standard UNIX client-server file sharing protocol
    - • Common Internet File System CIFS is standard Windows protocol
    - • Standard operating system file calls are translated into remote calls

        - •服务器可以服务多个客户端
        - •客户端和用户对客户端的识别不安全或复杂
        - •网络文件系统NFS是标准的UNIX客户端-服务器文件共享协议
        - •公共Internet文件系统CIFS是标准的Windows协议
        - •标准操作系统文件调用被转换为远程调用
  - Distributed Information Systems implement unified access to information needed for remote computing (LDAP, DNS, NIS, Active Directory).
    - 分布式信息系统实现了对远程计算所需信息的统一访问(LDAP、DNS、NIS、Active Directory)。
- Protection
  - Protection mechanisms provide controlled access by limiting thetypes of file access that can be made.
    - 保护机制通过限制可以进行的文件访问类型来提供受控访问。
  - File owner/creator should be able to control:
    - 文件所有者/创建者应该能够控制:
    - ☐ what can be done
    - ☐ by whom
- Access-Control List and Groups
  - general scheme to implement identity dependent access is to associate with each file and directory an access-control list (ACL) specifying usernames and the types of access allowed for each user.
    - 实现身份依赖访问的一般方案是为每个文件和目录关联一个访问控制列表(ACL)，指定用户名和每个用户允许的访问类型。
  - Mode of access: read, write, execute (R, W, X)
  - The classifications:
    - a) owner access - the user who created the file is the owner.
    - b) group access - a set of users who are sharing the file and need similar access is a group, or work group.
    - c) public access / universe - all other users in the system constitute the universe.
      - A)所有者访问——创建文件的用户是文件的所有者
      - B)组访问——一组共享文件并需要类似访问权限的用户称为组，或工作组。
      - C)公共访问/宇宙——系统中的所有其他用户构成宇宙
  -

Alice can read and write to the file **X**, can read the file **Y**, and can execute the file **Z**.

Bob can read X, can read and write to **Y**, and cannot access **Z**.

Write a set of *Access Control Lists* for this situation. Which list is associated with which file?

Solution:

ACL for:

X = (Alice, read/write), (Bob, read)

Y = (Alice, read), (Bob, read/write)

Z = (Alice, execute), (Bob, -----)

Example 1

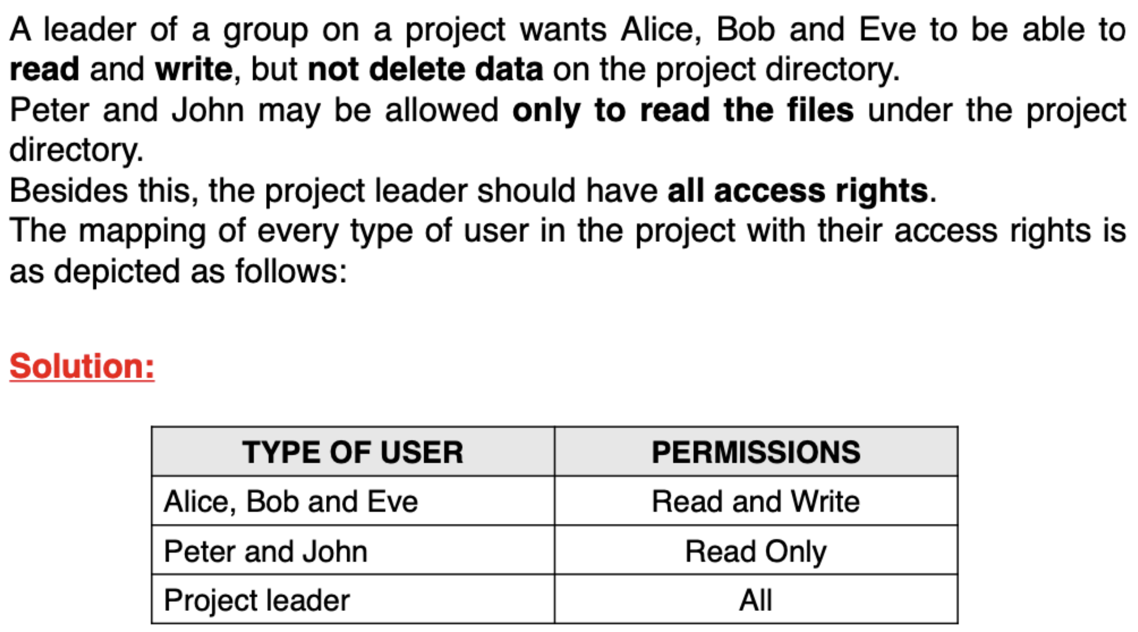
The following is an access verification technique, listing several files and the access allowed for a single user. Identify the control technique used here and for each, explain the type of access allowed.

a. File_1	RX
b. File_12	RWX
c. File_13	RW
d. File_14	X

Solution

This is an access control list.

a. File_1	This user can Read and Execute File 1.
b. File_12	This user can Read, Write, and Execute File 12 .
c. File_13	This user can Read and Write File 13.
d. File_14	This user can only Execute File 14 but cannot Read, Write the file.



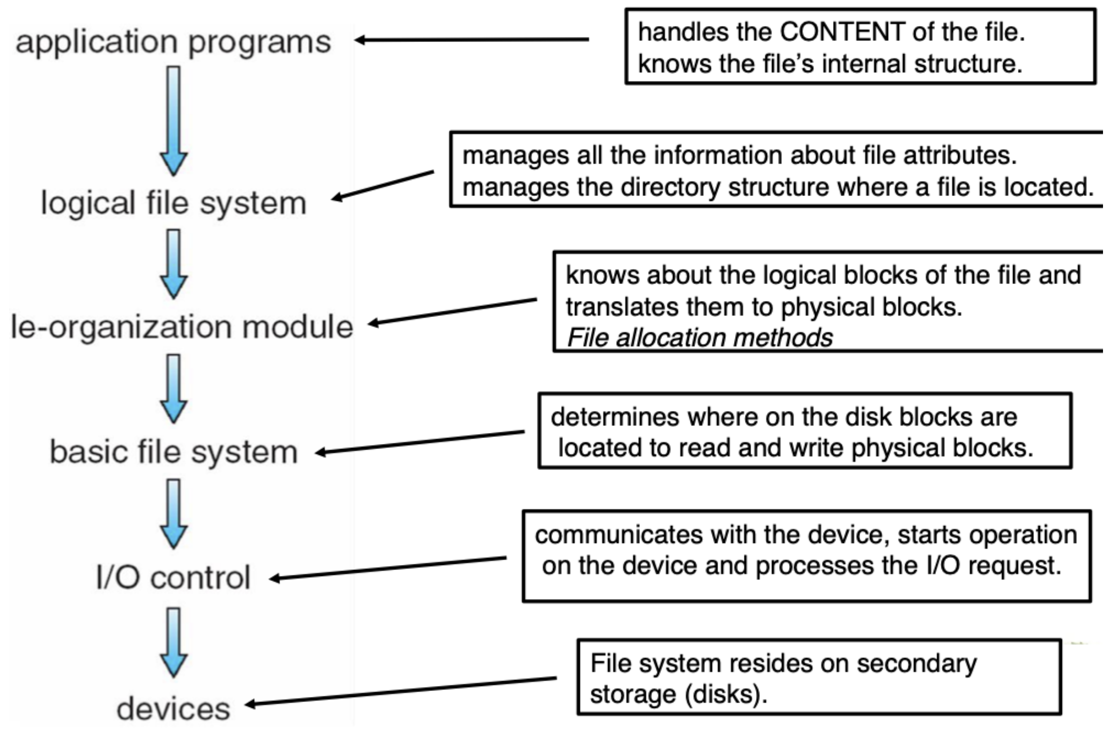
A leader of a group on a project wants Alice, Bob and Eve to be able to **read** and **write**, but **not delete data** on the project directory.
Peter and John may be allowed **only to read the files** under the project directory.
Besides this, the project leader should have **all access rights**.
The mapping of every type of user in the project with their access rights is as depicted as follows:

Solution:

TYPE OF USER	PERMISSIONS
Alice, Bob and Eve	Read and Write
Peter and John	Read Only
Project leader	All

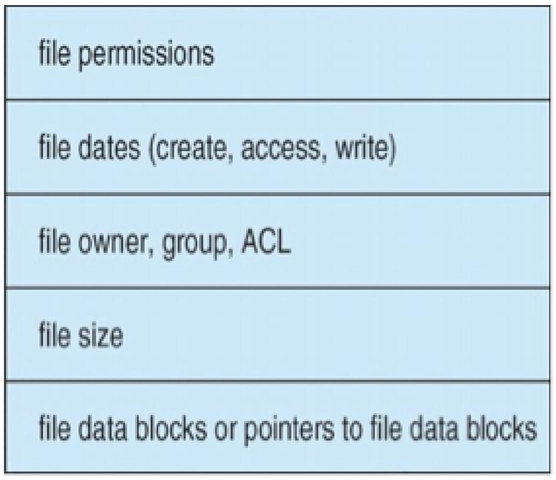
- 文件访问控制列表acl
- user=<文件实体,权限>

- File-System Structure
  - Layered File System
    - handles the CONTENT of the file. knows the file’s internal structure.
      - 处理文件的CONTENT。知道文件的内部结构。
    - manages all the information about file attributes. manages the directory structure where a file is located.
      - 管理文件属性的所有信息。管理文件所在的目录结构。
    - knows about the logical blocks of the file and translates them to physical blocks. File allocation methods
      - 了解文件的逻辑块并将其转换为物理块。文件分配方法
    - determines where on the disk blocks are located to read and write physical blocks.
      - 确定读取和写入物理块的磁盘块的位置。
    - communicates with the device, starts operation on the device and processes the I/O request.
      - 与设备通信，启动设备上的操作并处理I/O请求。
    - File system resides on secondary storage (disks).
      - 文件系统位于二级存储(磁盘)上。
    -

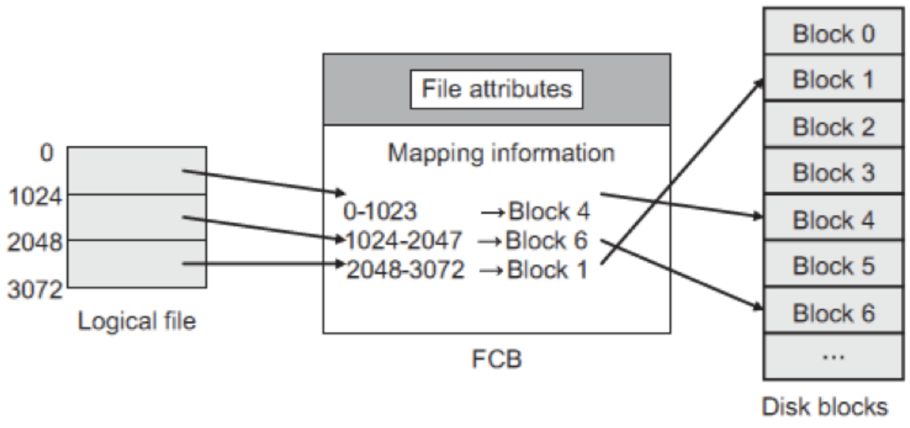

application programs
handles the CONTENT of the file.
knows the file's internal structure.
logical file system
manages all the information about file attributes.
manages the directory structure where a file is located.
le-organization module
knows about the logical blocks of the file and translates them to physical blocks.
File allocation methods
basic file system
determines where on the disk blocks are located to read and write physical blocks.
I/O control
communicates with the device, starts operation on the device and processes the I/O request.
devices
File system resides on secondary storage (disks).

- File System Implementation
  - File system needs to maintain on-disk or in-memory structures▪ On-disk for data storage.
    - 文件系统需要维护磁盘或内存结构——磁盘上的数据存储。
    - On-disk for data storage.
      - 磁盘上用于数据存储。
      - On disk, the file system may contain information about how to boot an operating system stored there, the total number of blocks, the number and location of free blocks, the directory structure, and individual files.
        - 在磁盘上，文件系统可能包含有关如何引导存储在那里的操作系统、块总数、空闲块的数量和位置、目录结构和单个文件的信息。
    - ▪ In-memory for data access.
      - 在内存中进行数据访问。
      - The in-memory information is used for both file-system management and performance improvement via caching.
        - 内存中的信息用于文件系统管理和通过缓存提高性能
- File System Implementation On-disk Structure
  - Boot control block is the first block of volume, and it contains information needed to boot an operating system.
    - 引导控制块是卷的第一个块，它包含引导操作系统所需的信息。
  - oFile Control Block (FCB per file) contains details about file, and it has a unique identifier number to allow association with directory entry. It is also known as Inode (Index node).

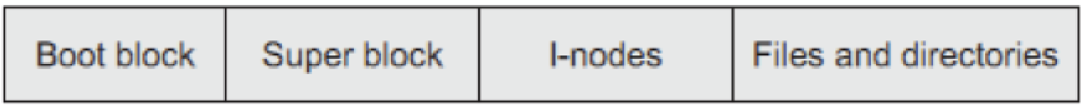
- 文件控制块(每个文件的FCB)包含文件的详细信息，它有一个唯一的标识符编号，允许与目录条目相关联。它也被称为Inode(索引节点)。
-

  
  file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

- File mapping through FCB通过FCB进行文件映射
  - The file system uses the logical position in a file stored by the FCB tomap it to a physical location on the disk.
    - 文件系统使用FCB存储的文件中的逻辑位置将其映射到磁盘上的物理位置。
  - The FCB contains a list of blocks of a file and their corresponding disk block addresses.
    - FCB包含一个文件的块列表和它们对应的磁盘块地址。
  - To retrieve the data at some position in a file, the file system first translates the logical position to a physical location in the disk.
    - 为了检索文件中某个位置的数据，文件系统首先将逻辑位置转换为磁盘中的物理位置。
  -

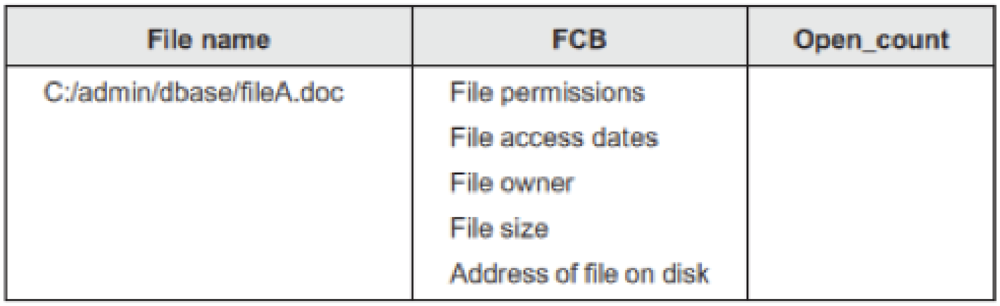
    

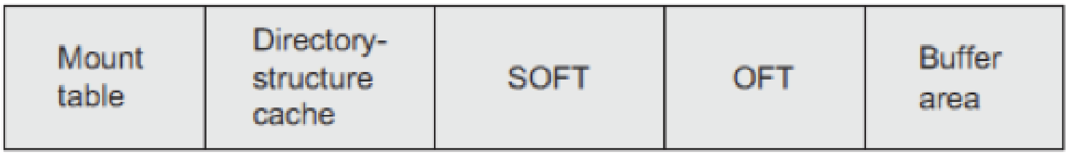
- o Volume control block / Superblock contains volume/partition details: no.of blocks in the partition, block size, free block count, block pointers, etc.
  - o卷控制块/超级块包含卷/分区的详细信息:no。分区中的块，块大小，空闲块计数，块指针等。

- oFiles and Directory Structure stores file names and associated filenames.
    - 文件和目录结构存储文件名和相关文件名。
- 

Boot block	Super block	I-nodes	Files and directories

  Disk layout implementing the file system
- File System Implementation In-memory Structure
    - Mount table stores file system mounts, mount points, file system types.
        - 挂载表存储文件系统挂载、挂载点、文件系统类型。
    - - Directory (structure cache) holds information of recently accessed directories.
        - —目录(结构缓存)保存最近访问过的目录信息。
    - - System-wide open-file table SOFT maintains the information about the open files in the system.
        - —系统范围的open-file table SOFT维护系统中打开的文件信息。
        - ***Structure of the SOFT***

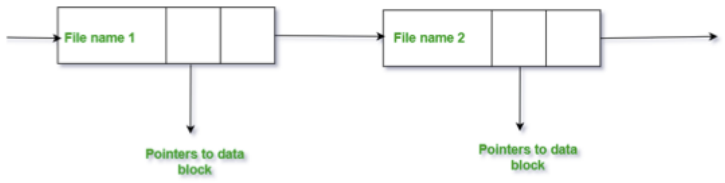
          

File name	FCB	Open_count
C:/admin/dbase/fileA.doc	File permissions File access dates File owner File size Address of file on disk	
    - - Per-process open-file table OFT maintains the detail of every file opened by a process and an entry in the OFT points to a SOFT.
        - -每进程打开文件表OFT维护了进程打开的每个文件的详细信息，并且OFT中的条目指向一个软文件。
    - - Buffer area is a temporary storage area in the memory for assisting in the reading/writing of information from/to disk.
        - —缓冲区是内存中的一个临时存储区域，用于帮助从磁盘读/写信息。
    - 

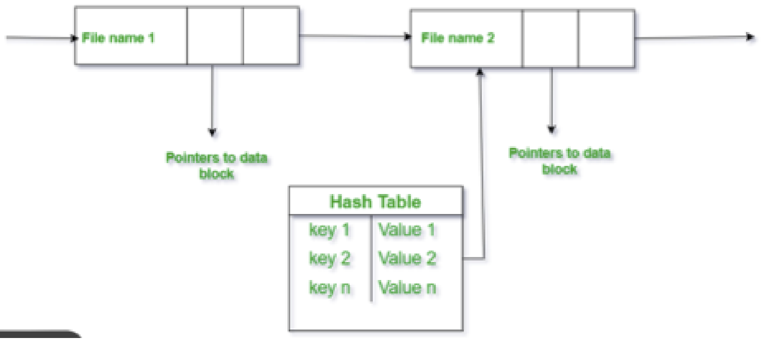
Mount table	Directory-structure cache	SOFT	OFT	Buffer area

      In-memory data structures implementing the file system
- Directory Implementation in O.S.

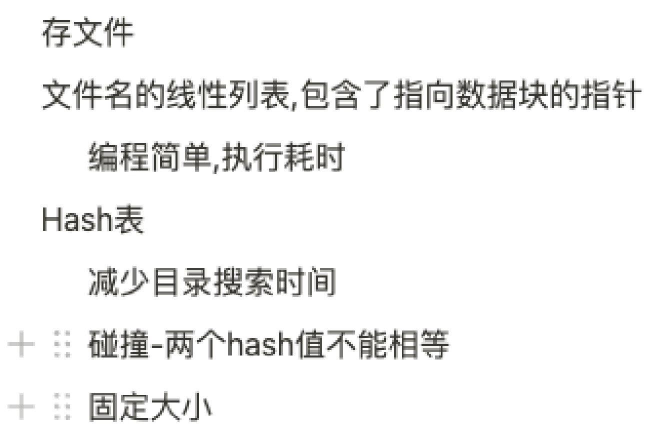
- A file system uses directory to provide a way to name and organize multiple files. Directory implementation in the operating system can be done as:
    - 文件系统使用目录来提供命名和组织多个文件的方法。目录在操作系统中的实现方法如下:
- Linear list - all the files in a directory are maintained as singly lined list.Each file contains the pointers to the data blocks which are assigned to itand the next file in the directory.
    - 线性列表-目录中的所有文件都以单行列表的形式维护。每个文件都包含指向分配给它的数据块的指针和目录中的下一个文件。
    - ➢ simple to program
        - 程序设计简单



- Hash Table – the hash table takes a value computed from the file name and returns a pointer to the file name in the linear list.
    - 哈希表-哈希表接受从文件名计算的值，并返回一个指向线性列表中的文件名的指针。
    - ➢ it can greatly decrease the directory search time.
        - (四)可大大减少目录查找时间。



        -



存文件

文件名的线性列表,包含了指向数据块的指针

编程简单,执行耗时

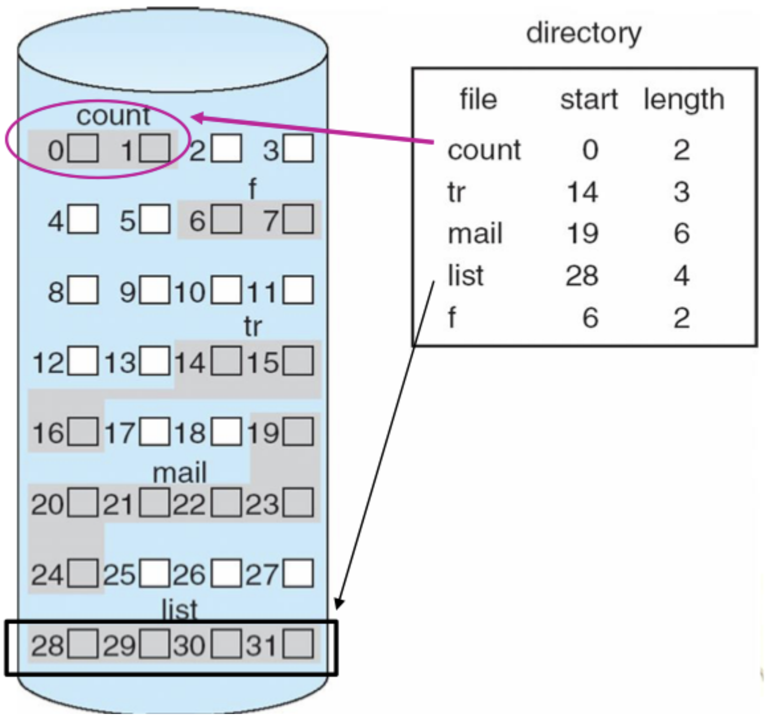
Hash表

减少目录搜索时间

碰撞-两个hash值不能相等

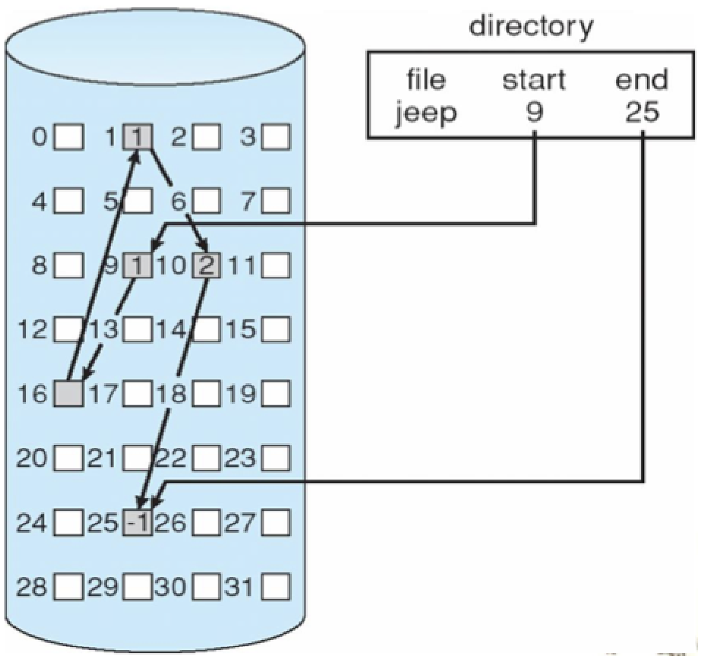
固定大小

- Allocation Methods

- Contiguous Allocation
  - Each file occupies a set of contiguous blocks on the disk.
    - 每个文件占用磁盘上一组连续的块。
  - Simple – only starting location (block #) and length (number of blocks) are required
    - 简单-只需要起始位置(块#)和长度(块数量)
  - Random access
    - 随机存取
  - Wasteful of space(dynamic storage-allocation problem)
    - 空间浪费(动态存储分配问题)
  - Files cannot grow
    - 文件不能增长
  - external fragmentation ,need for compaction off-line(downtime) or on-line
    - 外部碎片化，需要离线(停机)或在线压缩
    - 优势
      - 文件读取表现好
      - 高效的顺序和随机访问
    - 缺点
      - 碎片
      - 文件增长问题
      - 有一个文件要扩展,不得不吧正接着后面的那个文件娜一个位置,或者自己挪,‘
      - 可以放只读的文件,一次写完不会再修改扩容
  - 

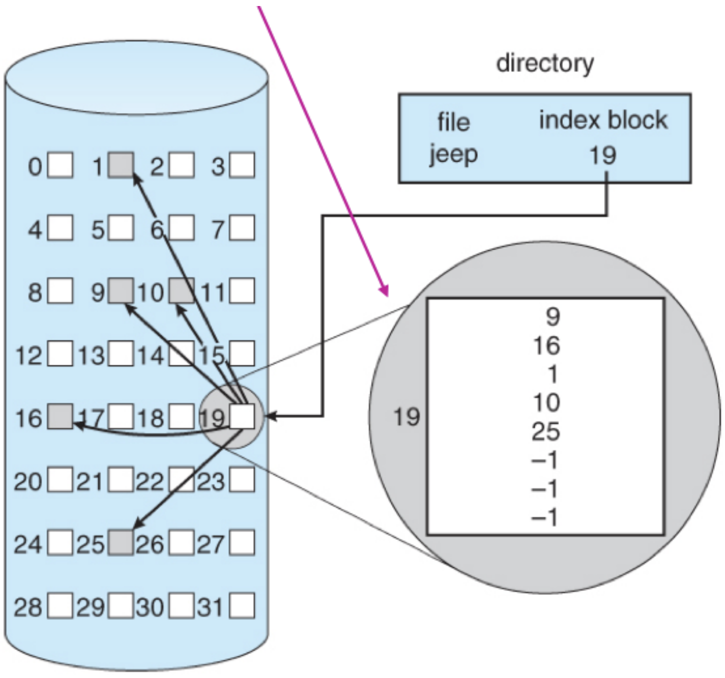
file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

- Linked Allocation
  - each file is a linked list of disk blocks-
    - 每个文件都是一个磁盘块链表
  - only efficient for sequential access files,
    - 仅对顺序访问文件有效,
  - - random access requires starting at the beginning of the list foreach new location access.
    - -随机访问要求从列表的开头开始每个新的位置访问。
  - 文件以数据块链表方式储存
  - 文件头包括了第一块和最后一块的指针
  - 优点
    - 创建,增大,缩小很方便
    - 基本没有碎片
  - 缺点
    - 不可能进行真正的随机访问,不高效,是串形
    - 可靠性,链接信息丢失
  -

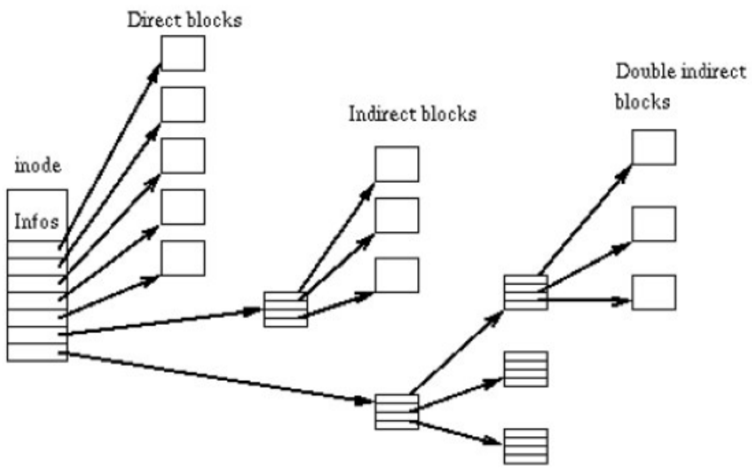


- Indexed Allocation
  - Each file has its own index block(s) of pointers to its data blocks
    - 每个文件都有自己的索引块，其中包含指向其数据块的指针
  - Brings all pointers together into the index block.
    - 将所有指针聚集到索引块中。
  - Need index table

        - 需要索引表
    - Random access
        - 随机存取
    - Dynamic access
        - 动态访问
    -



- Unix Inode
    - Unix uses an indexed allocation structure
        - Unix使用索引分配结构
    - - An inode (indexed node) stores both the attributes (infos) and the pointers to disk blocks.
        - —inode(索引节点)存储属性(信息)和指向磁盘块的指针。
    - Typically, an inode contains 15 pointers as follows:
        - 通常，一个inode包含15个指针，如下所示:
        - 12 direct pointers - that point directly to blocks
            - 12个直接指针-直接指向块
        - 1 indirect pointer - point to indirect blocks; each indirect block contains pointersthat point directly to blocks
            - 1间接指针-指向间接块;每个间接块包含直接指向块的指针
        - 1 double indirect pointer - that point to doubly indirect blocks, which are blocks that have pointers that point to additional indirect blocks
            - 1双间接指针-指向双间接块，这些块具有指向其他间接块的指针
        - 1 triple indirect pointer - etc
            - 1个三重间接指针-等等
    -



- How big the index block should be? and How it should be implemented?
  - 索引块应该有多大?它应该如何实施?
  - There are several approaches:
    - 有几种方法:
  - - Linked Scheme - An index block is one disk block, which can be read and written in a single disk operation. The first index block contains some header information, the first N block addresses, and if necessary, a pointer to additional linked index blocks.
    - —链接方案—索引块是一个磁盘块，可以在一次磁盘操作中读取和写入索引块。第一个索引块包含一些头信息、前N个块地址，如果需要，还包含一个指向其他链接索引块的指针。
  - - Multi-Level Index - The first index block contains a set of pointers to secondary index blocks, which in turn contain pointers to the actual data blocks.
    - —多级索引—第一个索引块包含一组指向二级索引块的指针，二级索引块又包含指向实际数据块的指针。
  - - Combined Scheme
    - -合并方案
- Free-Space Management
  - Disk management maintains free-space list to track available blocks / freespace.
    - 磁盘管理维护空闲空间列表以跟踪可用块/空闲空间。
    - Bit Vector - each bit represents a disk block, set to 1 if free or 0 if allocated.
      - 位向量-每个位代表一个磁盘块，如果空闲则设为1，如果已分配则设为0。
    - Linked List - link together all the free disk blocks, keeping a pointer to the first free block .
      - 链表-将所有空闲的磁盘块链接在一起，保持指向第一个空闲块的指针。
    - Grouping - stores the addresses of n free blocks in the first free block.
      - 分组——将n个空闲块的地址存储在第一个空闲块中。
    - Counting - the number of contiguous free blocks.
      - 计数-连续空闲块的数量。

> > - Space Maps - free-space list is implemented as a bit map, bit maps must be modified both when blocks are allocated and when they are freed.
> >   > - 空间映射-自由空间列表是作为位映射实现的，位映射必须在分配块和释放块时进行修改。