OPERATING SYSTEMS

File System Interface

Operating Systems

郑贵锋 博士
中山大学计算机学院
zhenggf@mail.sysu.edu.cn
https://gitee.com/code_sysu

中山大學 SUN YAT-SEN UNIVERSITY

- 目录
  - 基本概念
    - 文件概念
    - 文件属性
    - 文件类型
    - 文件操作
    - 访问方法
    - 文件系统
  - 文件目录
    - 元素
    - 操作
    - 单级目录
    - 二级目录
    - 树形目录
    - 无环图目录
    - 通用图目录
    - 文件系统挂载
  - 文件共享和保护

Basic Concepts

文件概念

- 文件是记录在辅助存储器上的相关信息的命名集合。
  - 通常，文件表示程序（源代码和对象形式）和数据。文件中的信息由其创建者和用户定义。
  - 操作系统将文件映射到具有连续逻辑空间的物理设备上，这些设备通常是非易失性的。
  - 从用户的角度来看，文件是逻辑辅助存储的最小分配。
- 文件结构由操作系统或应用程序决定。
  - 无结构
    - 文件是位、字节、字等的流。
  - 简单记录结构
    - 多行
    - 固定长度
    - 可变长度
  - 复杂结构
    - 格式化文件
    - 可重定位加载文件

Basic Concepts

文件属性

- 典型文件属性
  - 名称 – 一个符号文件名，是以人类可读形式保存的标识文件的唯一信息。
  - 标识符 – 一个唯一的标记，通常是一个数字，用于标识文件系统中的文件。
  - 类型 – 支持不同类型文件的系统所需的声明。
  - 位置 – 指向设备上文件位置的指针。
  - 尺寸 – 文件的当前大小（通常以字节为单位）。
  - 保护 – 访问控制信息，以确定谁能进行读、写、执行等操作。
  - 时间戳和用户标识 – 用于保护、安全和使用监控的数据。
- 有关文件的信息保存在目录结构中，目录结构维护在文件所在的设备上。
- 有许多变体，包括扩展文件属性，如文件校验和。

Basic Concepts

- **文件属性**

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file was last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Basic Concepts

- **文件类型扩展名**

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

Basic Concepts

- **文件类型扩展名**
  - 例子

Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
file.mpg	Movie encoded with the MPEG standard
file.o	Object file (compiler output, not yet linked)
file.pdf	Portable Document Format file
file.ps	PostScript file
file.tex	Input for the TEX formatting program
file.txt	General text file
file.zip	Compressed archive

Basic Concepts

- **通用文件操作**
  - 文件是一种抽象数据类型
    - 创建
    - 删除
    - 打开
    - 关闭
    - 截断
    - 设置属性
    - 获取属性
    - 读
    - 写
    - 追加
    - 在文件中重新定位

Basic Concepts

- **打开文件**
  - 与打开的文件关联的若干信息
    - 文件指针
      - 当前文件位置指针用于跟踪上次读写位置。此指针对于在文件上操作的每个进程都是唯一的。
    - 文件打开计数
      - 文件打开计数跟踪文件的打开和关闭次数，并在最后关闭时为0. 然后，操作系统可以从打开的文件表中删除该文件的条目。
    - 文件的磁盘位置
      - 定位文件所需的信息保存在内存中，这样系统就不必为每个操作从目录结构中读取它。
    - 访问权限
      - 每个进程以访问模式打开一个文件。此信息存储在每个进程表中，因此操作系统可以允许或拒绝后续I/O请求。

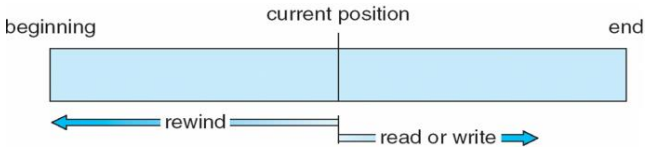
Basic Concepts

- **打开文件**
  - 锁定打开文件
    - 由某些操作系统和文件系统提供
      - 类似于读者写者锁
      - 共享锁与读者锁类似
        - 多个进程可以同时获取。
      - 独占锁类似于写者锁
    - 强制性或建议性
      - 强制性的
        - 操作系统根据持有和请求的锁（例如被独占时）拒绝访问。例如：Windows
      - 建议性的
        - 进程可以找到锁的状态并（由程序员）决定执行什么操作。例如：Unix/Linux

Basic Concepts

- **访问方法**
  - 顺序存取：类似磁带模式的操作

read next	读取指针下一记录，指针相应移动
write next	在结尾写入一记录，指针移至新结尾
reset	回到文件开头
no read after last write	写入后不能再读（因在结尾）
rewrite	可以重复写



Basic Concepts

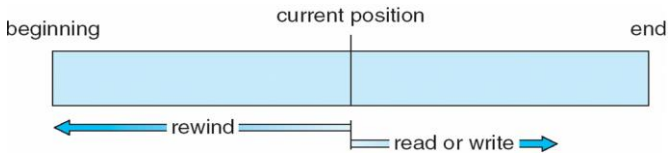
- **访问方法**
  - 直接访问
    - 假设文件具有固定长度的逻辑记录。

```
read n
write n
position to n
    read next
    write next
rewrite n
```

    - *n* = 相对块号（相对于文件开头的索引）
    - （逻辑）相对块号允许操作系统决定文件应该放在（物理）哪里。
      - 请参阅后面的课程中的分配问题。

Basic Concepts

- **访问方法**
  - 模拟对直接访问文件的顺序访问。



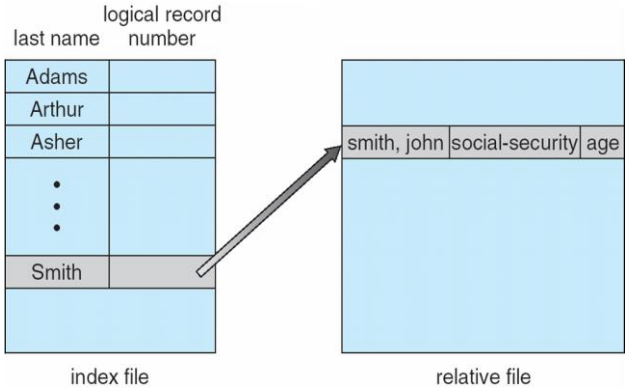
sequential access	implementation for direct access
reset	cp = 0;
read_next	read cp; cp = cp + 1;
write_next	write cp; cp = cp + 1;

Basic Concepts

- **访问方法**
  - 其他访问方法
    - 可以在基本方法之上构建。
    - 一般包括为文件创建索引。
    - 将索引保存在内存中，以便快速确定要操作的数据的位置。
    - 如果太大，索引（在磁盘上）的索引（在内存中）。
    - IBM索引顺序访问方法(ISAM)：
      - 小型主索引，指向二级索引的磁盘块
      - 文件按定义的键进行排序
      - 所有这些都是由操作系统完成的。
    - VMS操作系统提供索引和相关文件作为另一个示例（请参阅下一张幻灯片）。

Basic Concepts

- **访问方法**
  - 索引和相关文件的示例

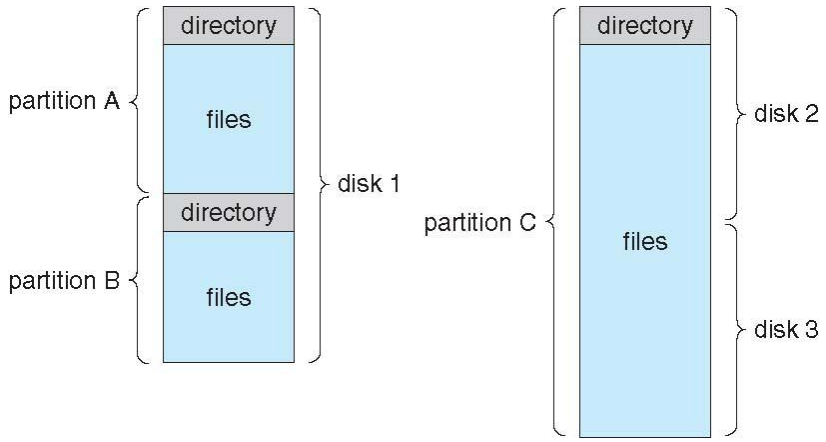


Basic Concepts

- **文件系统组织**
  - 磁盘结构
    - 磁盘可以细分为多个分区。
      - 磁盘或分区可以是RAID（独立磁盘的冗余阵列），以防出现故障。
      - 磁盘或分区可以使用原始格式。
        - 它可以在没有文件系统的情况下使用，也可以使用文件系统进行格式化。
      - 分区也称为小型磁盘、片。
    - 包含文件系统的实体也称为卷。
      - 包含文件系统的每个卷还跟踪设备目录或卷目录中该文件系统的信息。
    - 除了通用文件系统外，还有许多专用文件系统，通常都在同一操作系统或计算机中。

文件系统组织

- 典型的文件系统组织



文件系统的类型

- 我们主要讨论通用文件系统。
  - 操作系统通常有文件系统，有些是通用的，有些是专用的。
- 例如，Solaris具有：
  - tmpfs – 用于快速、临时I/O的基于内存的易失性FS
  - objfs – 进入内核内存的接口，用于获取内核符号以进行调试
  - ctfs – 用于管理守护进程的契约文件系统
  - lofs – 环回文件系统，允许在一个FS访问另一个FS，即创建新的虚拟FS，以使用替代的路径名访问文件
  - procfs – 进程结构的内核接口
  - ufs, zfs – 通用文件系统

File Directories

- **文件目录**
  - 文件目录包含目录中文件的信息，包括其属性、位置和所有权。
    - 这些信息中的大部分，特别是与存储有关的信息，都是由操作系统管理的。
  - 文件目录本身就是一个文件，可由各种文件管理例程访问。
    - 尽管目录中的一些信息可供用户和应用程序使用，但这些信息通常由系统例程间接提供。
  - 可以将文件目录视为符号表，表中将文件名转换为其文件控制块。
    - 表的每个条目对应一个文件。
    - 目录组织必须允许我们插入条目、删除条目、搜索命名条目，以及列出目录中的所有条目。

File Directories

- **文件目录**
  - 文件目录的信息元素
    - 基本信息
      - 文件名
      - 文件类型
      - 文件组织
    - 地址信息
      - 卷
      - 起始地址
      - 使用的大小
      - 分配的大小
    - 访问控制信息
      - 所有者
      - 访问信息
      - 允许的操作

File Directories

- **文件目录**
  - 文件目录的信息元素（续）
    - 使用信息
      - 创建日期
      - 创造者身份
      - 上次读取日期
      - 最后一位读者的身份
      - 上次修改日期
      - 最后一个修改者的身份
      - 上次备份的日期
      - 当前使用情况
        - 当前文件活动的信息，如打开文件的进程、是否被一个进程加锁、是否在内存中修改但修改未写入磁盘等

File Directories

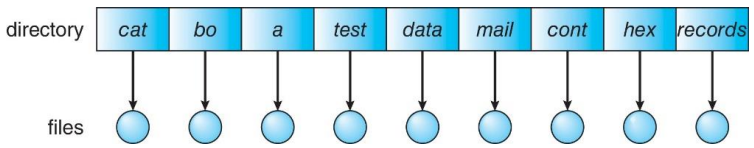
- **文件目录**
  - 对目录执行的操作
    - 创建文件
    - 删除文件
    - 重命名文件
    - 列出目录
    - 遍历文件系统

File Directories

- **文件目录**
  - 目录组织
    - 目录按逻辑组织，以获得：
      - 效率
        - 快速定位文件。
      - 命名
        - 两个用户可以为不同的文件使用相同的名称。
        - 同一个文件可以有几个不同的名称。
      - 分组
        - 按属性对文件进行逻辑分组
        - 例如，所有的Java程序，所有的游戏，……

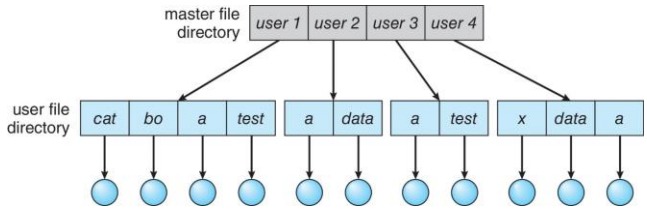
File Directories

- **文件目录**
  - 单级目录
    - 所有用户使用单一目录
    - 常见问题
      - 目录的最终长度
      - 为文件指定唯一的名称
      - 记住文件名
      - 文件分组（使用文件扩展名）



File Directories

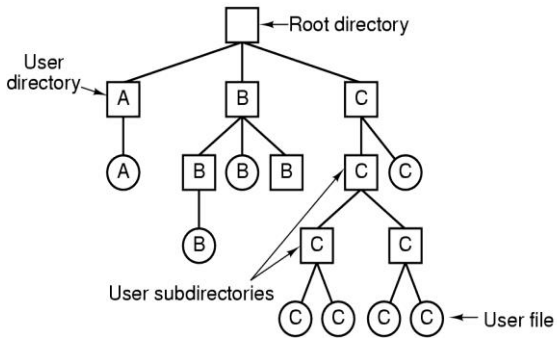
- **文件目录**
  - 二级目录
    - 为每个用户分隔目录
      - 使用路径名
      - 对于不同的用户可以具有相同的文件名
      - 提供高效的搜索
    - 没有分组功能



  - 主要问题：违反了零一无限原则。
    - 零一无限原则(Zero-One-Infinity, ZOI): “Allow none of foo, one of foo, or any number of foo.” - *Willem van der Poel*

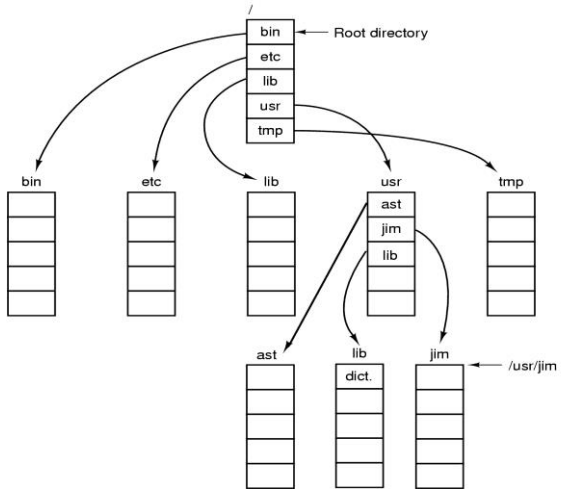
File Directories

- **文件目录**
  - 树状目录
    - 树状目录组件



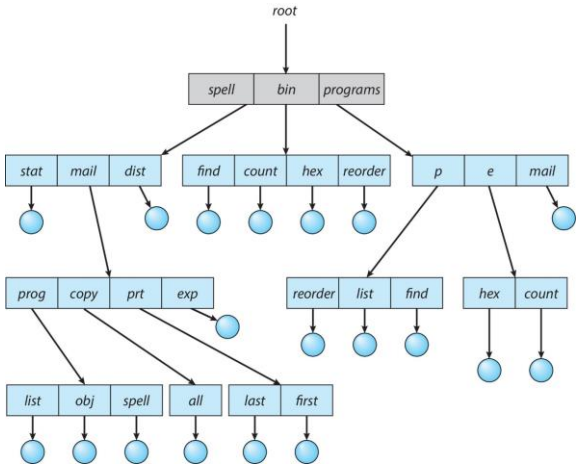
File Directories

- 文件目录
  - 树状目录
    - 目录路径名



File Directories

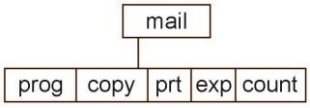
- 文件目录
  - 树状目录



File Directories

- **文件目录**
  - 树状目录
    - 提供高效的搜索
    - 具有分组能力
    - 当前目录（工作目录）

```
$cd /spell/mail/prog
$type list
```



    - 使用绝对或相对路径名
    - 在当前目录中创建新文件。

```
$touch <文件名>
```

    - 删除文件

```
$rm <文件名>
```

    - 在当前目录中创建新的子目录

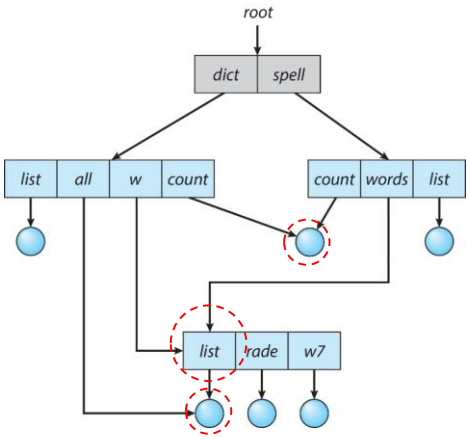
```
$mkdir <目录名>
```

    - 删除“mail” ⇒ 删除以“mail”为根的整个子树

```
$rmdir /spell/mail
```

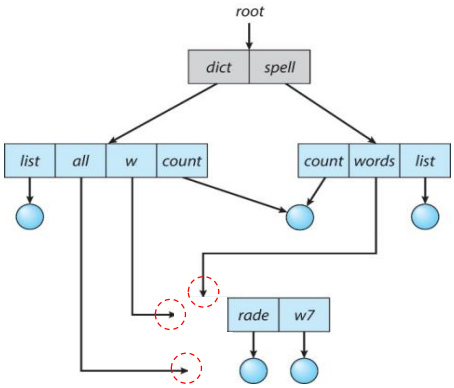
File Directories

- **文件目录**
  - 有向无环图(DAG)目录
    - 拥有共享的子目录和文件。
      - 一个条目可能有两个不同的名称和路径（别名）



File Directories

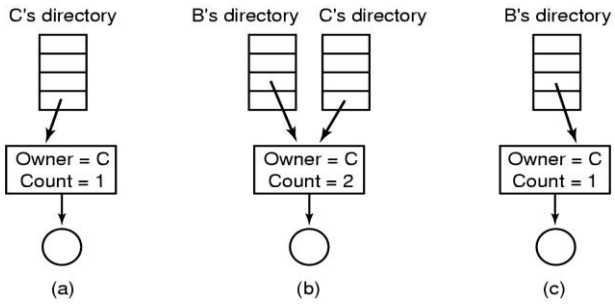
- **文件目录**
  - 有向无环图(DAG)目录
    - 如果/dict/w/list被删除 ⇒ 出现一些悬空指针



    - 目录项的新类型：
      - 链接文件Link – 指向现有文件的另一个名称（指针）
      - 解析链接 – 按照指针查找文件。

File Directories

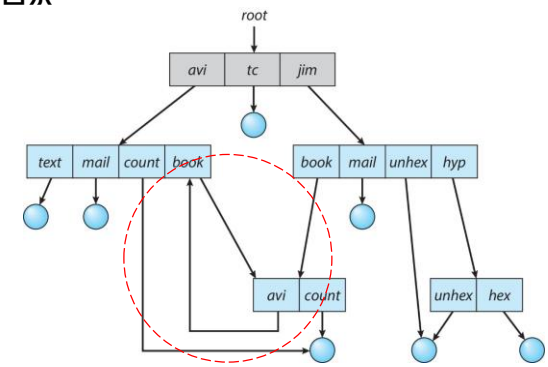
- **文件目录**
  - 有向无环图(DAG)目录
    - 使用共享计数器。



(a) 创建链接前的情况
(b) 在链接创建之后
(c) 原始所有者删除文件后

File Directories

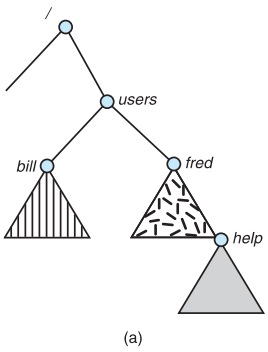
- **文件目录**
  - 通用图目录



  - 如何保证没有环？
    - 仅允许指向文件的链接，不允许指向子目录
    - 垃圾收集
    - 每次添加新链接时，使用循环检测算法确定其是否正常。

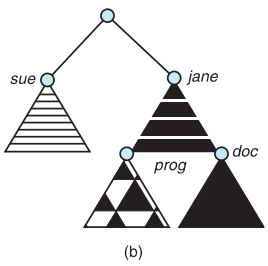
File Directories

- **文件系统挂载**
  - 在它被访问之前必须安装(mount, 挂载)文件系统
  - 在挂载点挂载未挂载的文件系统



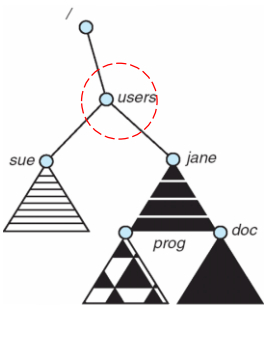
(a)

文件系统原状态



(b)

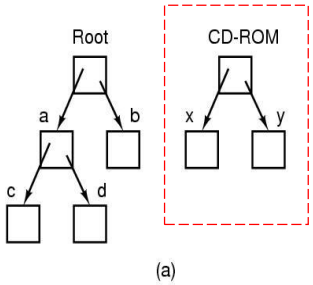
未挂载的文件系统



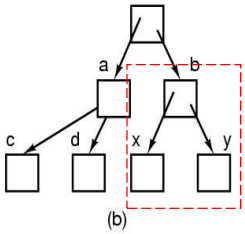
文件系统(a)挂载(b)后

File Directories

- **文件系统挂载**
  - 实例



(a) 挂载前



(b) 挂载后的系统

File Sharing and Protection

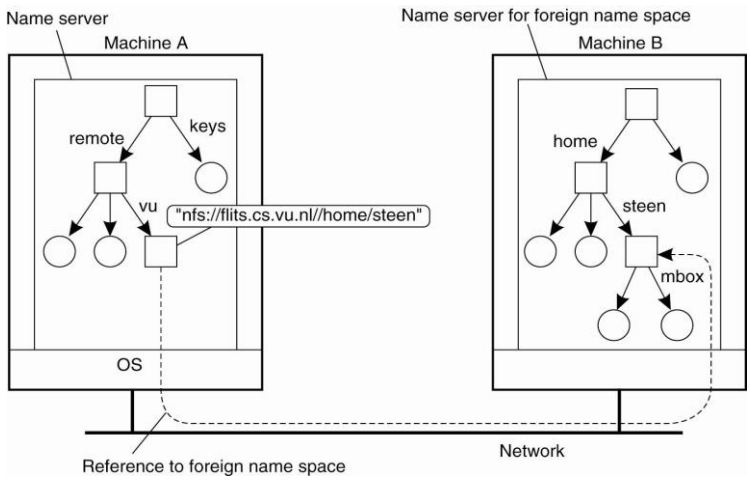
- **文件共享**
  - 在多用户系统上共享文件是可取的。
    - 用户ID标识用户，允许对每个用户进行权限和保护。
    - 组ID允许用户分组，允许组访问权限。
    - 所有者/组是文件/目录的属性。
  - 共享可以通过*保护方案*完成。
  - 在分布式系统上，文件可以通过网络共享。
    - 网络文件系统(NFS)是一种常见的分布式文件共享方法。

File Sharing and Protection

- **文件共享**
  - 远程文件系统
    - 使用网络允许系统之间的文件系统访问
      - 在Linux中通过命令行（如ftp）手动执行。
      - 自动地、无缝地使用分布式文件系统。
      - 通过Web半自动地访问。
    - 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。
      - 服务器可以服务于多个客户端。
      - 客户端和客户端用户的标识不安全/复杂。
      - NFS（网络文件系统）是一种标准的UNIX客户机-服务器文件共享协议。
      - CIFS（通用Internet文件系统）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。
      - 分布式信息系统（分布式命名服务），如LDAP、DNS、NIS、Active Directory，实现了对远程计算所需信息的统一访问

File Sharing and Protection

- **文件共享**
  - 挂载远程命名空间



文件共享

- 远程文件系统中的故障模式
  - 所有文件系统都有故障模式
    - 例如，目录结构或元数据（非用户数据）损坏。
  - 由于网络故障、服务器故障，远程文件系统添加了新的故障模式。
  - 从故障中恢复可能涉及有关每个远程请求状态的状态信息。
  - 无状态协议（如NFS v3）在每个请求中包含所有信息，允许轻松恢复，但安全性较低。

文件保护

- 文件所有者/创建者应能够控制：
  - 可以做什么，以及
  - 由谁来做
- 文件访问类型
  - 读
  - 写
  - 执行
  - 追加
  - 删除
  - 列出（文件名称、属性）
  - 修改属性
- 访问控制
  - 最常见方法是，根据用户身份控制访问

文件保护

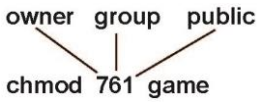
- 访问模式：读Read、写Write、执行eXecute
- Unix/Linux上的三类用户：所有者owner、组group和公共public
- 实例
  - 请管理员创建一个组（唯一名称），例如G，并向该组添加一些用户。
  - 对于特定的文件（比如game）或子目录，定义适当访问的模式

```
$chmod 761 game
```

  - 将文件附加到组G

```
$chgrp G game
```

	模式	R	W	X
所有者访问	7	1	1	1
组访问	6	1	1	0
公共访问	1	0	0	1



文件保护

- Linux目录列表示例

```
root@ubuntu:/etc# ls -l
total 1120
drwxr-xr-x  3 root root    4096 Aug  6  2019 acpi
-rw-r--r--  1 root root    3028 Aug  6  2019 adduser.conf
drwxr-xr-x  2 root root    4096 Jun 12 12:33 alternatives
-rw-r--r--  1 root root     401 May 30  2017 anacrontab
-rw-r--r--  1 root root     433 Oct  2  2017 apg.conf
drwxr-xr-x  6 root root    4096 Aug  6  2019 apm
drwxr-xr-x  3 root root    4096 Aug  6  2019 apparmor
drwxr-xr-x  8 root root    4096 Jul  6 17:50 apparmor.d
drwxr-xr-x  4 root root    4096 Apr 10 12:19 apport
-rw-r--r--  1 root root     769 Apr  4  2018 appstream.conf
drwxr-xr-x  7 root root    4096 Feb 21 14:10 apt
drwxr-xr-x  3 root root    4096 Aug  6  2019 avahi
-rw-r--r--  1 root root    2319 Apr  5  2018 bash.bashrc
-rw-r--r--  1 root root      45 Apr  2  2018 bash_completion
drwxr-xr-x  2 root root    4096 May 15 23:45 bash_completion.d
-rw-r--r--  1 root root     367 Jan 27  2016 bindresvport.blacklist
drwxr-xr-x  2 root root    4096 Apr 21  2018 binfmt.d
drwxr-xr-x  2 root root    4096 Apr 10 12:18 bluetooth
-rw-r-----  1 root root      33 Aug  6  2019 brlapi.key
drwxr-xr-x  7 root root    4096 Aug  6  2019 brltty
-rw-r--r--  1 root root   25341 Aug 29  2018 brltty.conf
```

  - 第一个字段描述文件或目录的保护。d作为第一个字符表示子目录。还显示了指向文件的链接数、所有者名称、组名称、文件大小（以字节为单位）、上次修改日期以及文件名。

文件保护

- Windows 10访问控制列表管理