File System Interface

Operating Systems

School of Data & Computer Science
Sun Yat-sen University

Lecture Notes: os_sysu@163.com
Instructor: Guoyang Cai
email: isscgy@mail.sysu.edu.cn

Contents

File Concepts

- A file is a named collection of related information that is recorded on secondary storage.
  - Commonly, files represent programs (both source and object forms) and data. The information in a file is defined by its creator and user.
  - Files are mapped by the operating system onto physical devices, usually nonvolatile, with contiguous logical space.
  - From a user's perspective, a file is the smallest allotment of logical secondary storage.
- File Structures are decided by the operating system or applications.
  - None structure
    - The file is a stream of bits, bytes, words, etc.
  - Simple record structure
    - Lines
    - Fixed length
    - Variable length.
  - Complex Structures
    - Formatted document
    - Relocatable load file.

File Attributes

- Typical File Attributes
  - *Name* – a symbolic file name, the only information kept in human-readable form to identify the file.
  - *Identifier* – a unique tag, usually a number, identifying the file within file system.
  - *Type* – declaration needed for systems that support different types of files.
  - *Location* – a pointer to the location of the file on device.
  - *Size* – the current size of the file (in bytes normally).
  - *Protection* – access-control information to determine who can do reading, writing, executing, and so on.
  - *Timestamps and user identification* – data for protection, security, and usage monitoring.
- Information about files are kept in the directory structure, which is maintained on the device where the files reside.
- There are many variations, including extended file attributes such as file checksum.

File Attributes

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

File Type Extensions

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

File Type Extensions

- Examples.

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

Common File Operations

- File is an abstract data type.
  - Create
  - Delete
  - Open
  - Close
  - Truncate
  - Set Attributes
  - Get Attributes
  - Read
  - Write
  - Append
  - Reposition within file

Open Files

- Several pieces of information associated with an open file
  - File pointer
    - A current-file-position pointer is used to track the last read–write location. This pointer is unique to each process operating on the file.
  - File-open count
    - The file-open count tracks the number of opens and closes of the file and reaches zero on the last close. The OS can then remove the entry of the file from the open-file table.
  - Disk location of the file
    - The information needed to locate the file is kept in memory so that the system does not have to read it from the directory structure for each operation.
  - Access rights
    - Each process opens a file in an access mode. This information is stored on the per-process table so the operating system can allow or deny subsequent I/O requests.

Open Files

- Open File Locking
  - Provided by some operating systems and file systems
    - Similar to reader-writer locks
    - *Shared lock* similar to reader lock
      - Several processes can acquire concurrently.
    - *Exclusive lock* similar to writer lock
  - Mandatory or advisory
    - *Mandatory*
      - Access is denied depending on locks held and requested.
    - *Advisory*
      - Processes can find status of locks and decide what to do.

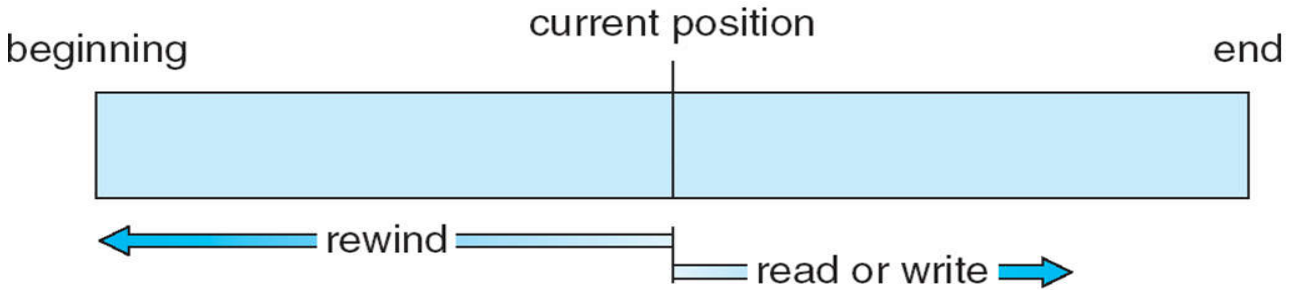
Access Methods

- Sequential Access

```
read next
write next
reset
```

no read after last write

rewrite



Access Methods

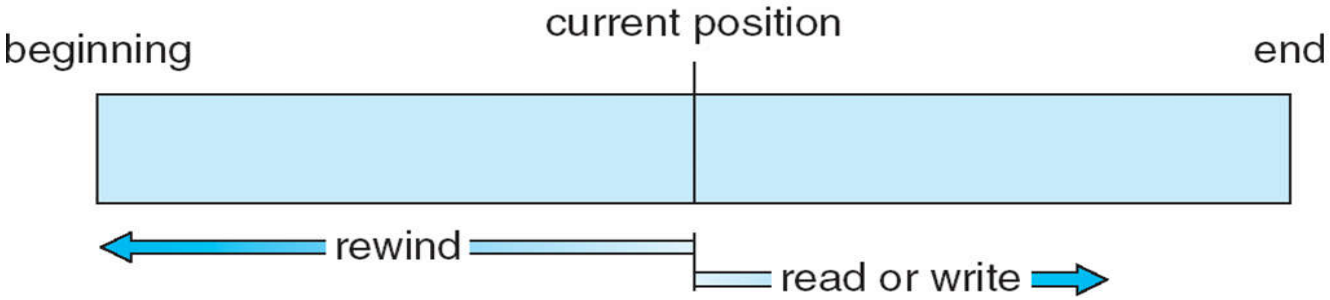
- Direct Access
  - Suppose that a file has fixed length logical records.

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

  - *n* = relative block number.
  - Relative block numbers allow OS to decide where the file should be placed.
    - See allocation problem in later lecture.

Access Methods

- Simulation of Sequential Access on Direct-access File.



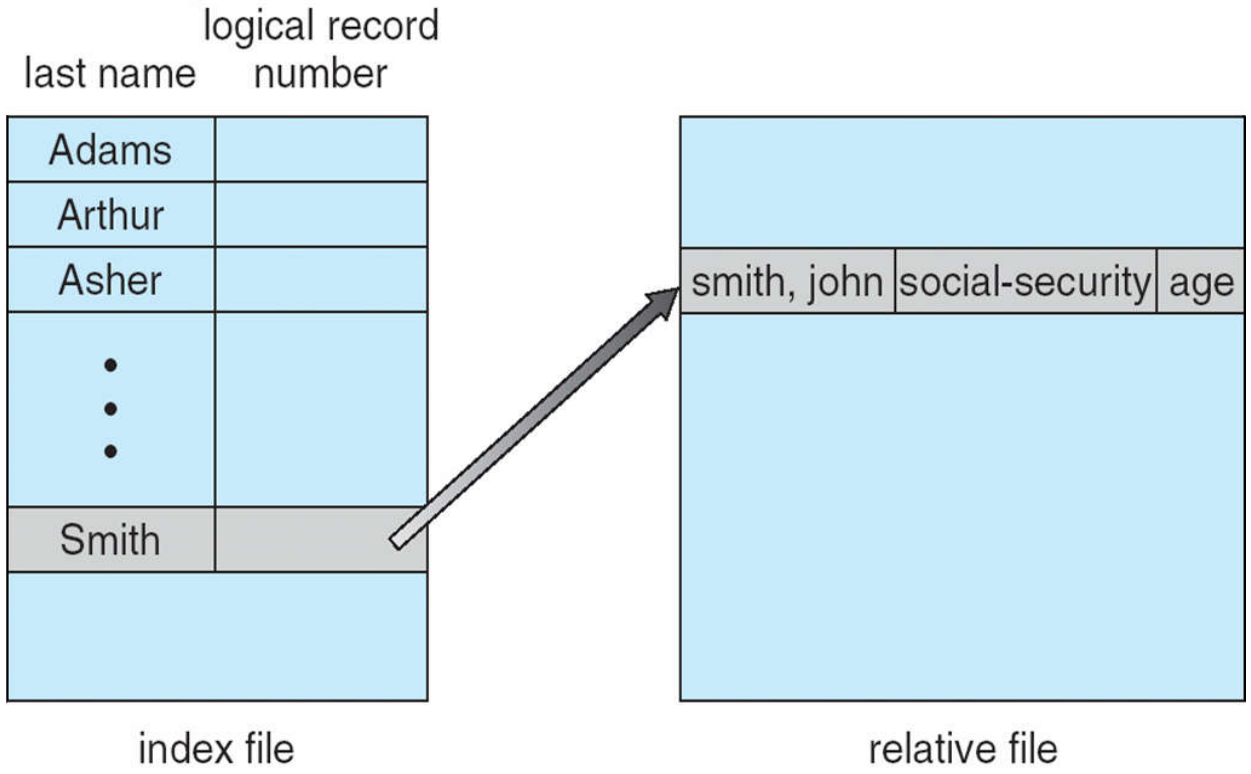
sequential access	implementation for direct access
reset	cp = 0;
read_next	read cp; cp = cp + 1;
write_next	write cp; cp = cp + 1;

Access Methods

- Other Access Methods
  - Can be built on top of base methods.
  - General involve creation of an *index* for the file.
  - Keep index in memory for fast determination of location of data to be operated on.
  - If too large, index (in memory) of the index (on disk).
  - IBM indexed sequential-access method (ISAM):
    - Small master index, points to disk blocks of secondary index
    - File kept sorted on a defined key
    - All done by the OS.
  - VMS operating system provides index and relative files as another example (see next slide).

Access Methods

- Example of Index and Relative Files.

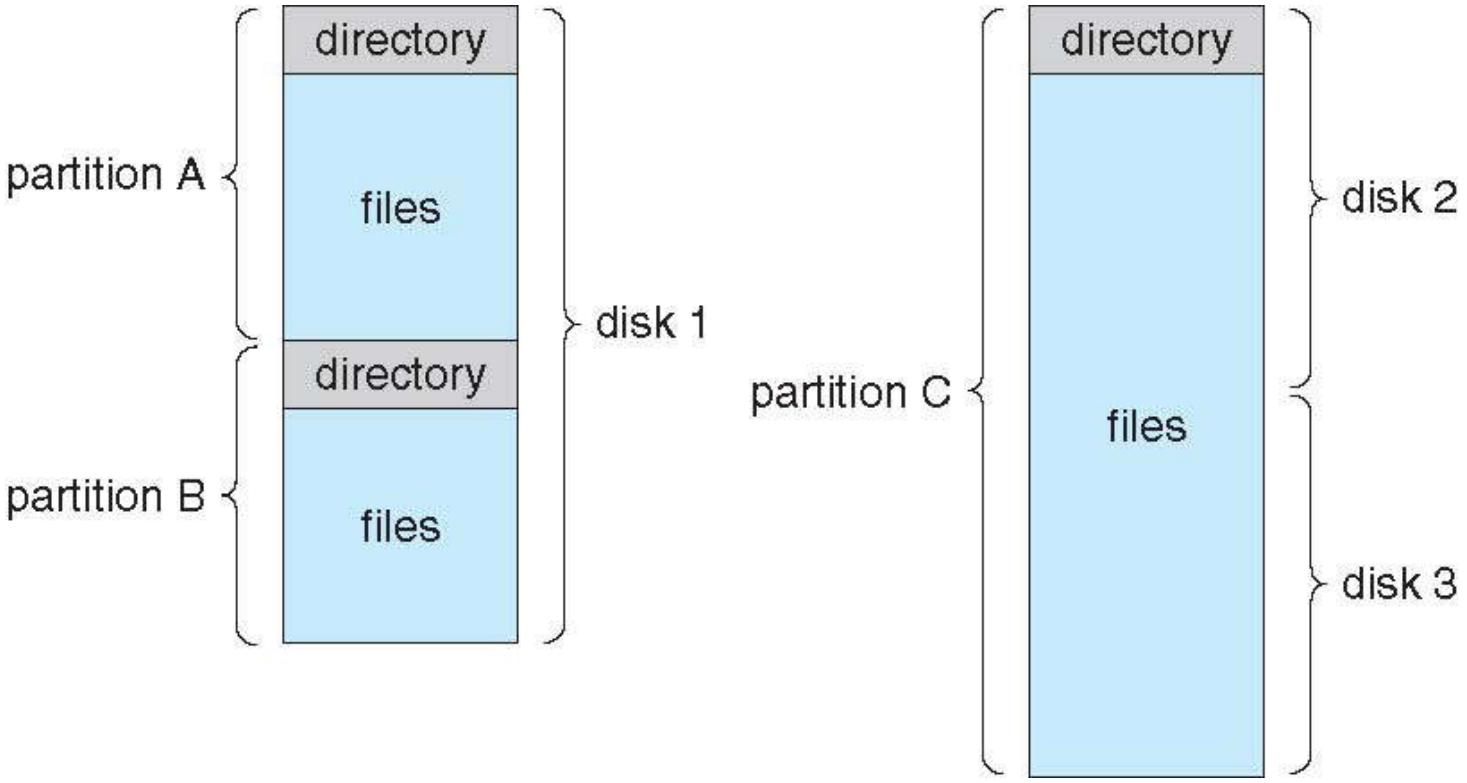


File-System Organization

- Disk Structures
  - Disk can be subdivided into partitions.
    - Disks or partitions can be RAID (Redundant Arrays of Independent Disks) protected against failure.
    - Disk or partition can be used raw.
      - It can be used without a file system, or be formatted with a file system.
    - Partitions also known as minidisks, slices.
  - Entity containing file system known as a volume.
    - Each volume containing file system also tracks that file system's info in device directory or volume table of contents.
  - As well as general-purpose file systems, there are many special-purpose file systems, frequently all within the same operating system or computer.

File-System Organization

- A Typical File-system Organization.



Types of File Systems

- We mostly talk of general-purpose file systems.
  - Operating systems frequently have may file systems, some general- and some special-purpose.
- For example, Solaris has:
  - tmpfs – memory-based volatile FS for fast, temporary I/O
  - objfs – interface into kernel memory to get kernel symbols for debugging
  - ctfs – contract file system for managing daemons
  - lofs – loopback file system allows one FS to be accessed in place of another
  - procfs – kernel interface to process structures
  - ufs, zfs – general purpose file systems

File Directories

- A file directory contains information about files in the directory, including their attributes, location, and ownership.
  - Much of this information, especially that concerned with storage, is managed by the operating system.
- A file directory is itself a file, accessible by various file management routines.
  - Although some of the information in directories is available to users and applications, this information is generally provided indirectly by system routines.
- A file directory can be viewed as a symbol table that translates file names into their file control blocks.
  - Each entry of the table corresponds to a file.
  - The directory organization must allow us to insert entries, to delete entries, to search for a named entry, and to list all the entries in the directory.

File Directories

- Information Elements of a File Directory
  - Basic Information
    - File Name
    - File Type
    - File Organization
  - Address Information
    - Volume
    - Starting Address
    - Size Used
    - Size Allocated
  - Access Control Information
    - Owner
    - Access Information
    - Permitted Actions

File Directories

- Information Elements of a File Directory (cont.)
  - Usage Information
    - Date Created
    - Identity of Creator
    - Date Last Read
    - Identity of Last Reader
    - Date Last Modified
    - Identity of Last Modifier
    - Date of Last Backup
    - Current Usage

File Directories

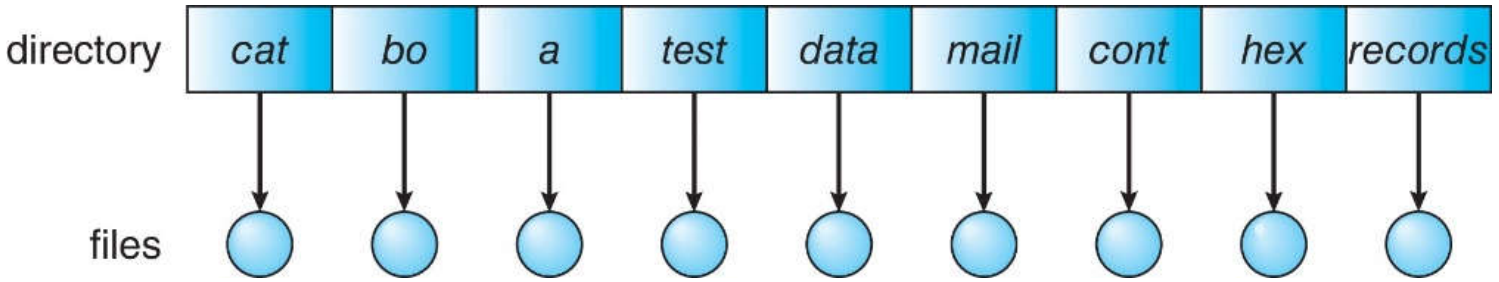
- Operations performed on a directory
  - Create a file
  - Delete a file
  - Rename a file
  - List a directory
  - Traverse the file system.

File Directories

- Directory Organization
  - The directory is organized logically to obtain:
    - Efficiency (效率)
      - locating a file quickly.
    - Naming (命名)
      - Two users can have same name for different files.
      - The same file can have several different names.
    - Grouping (分组)
      - logical grouping of files by properties
      - e.g., all Java programs, all games, …

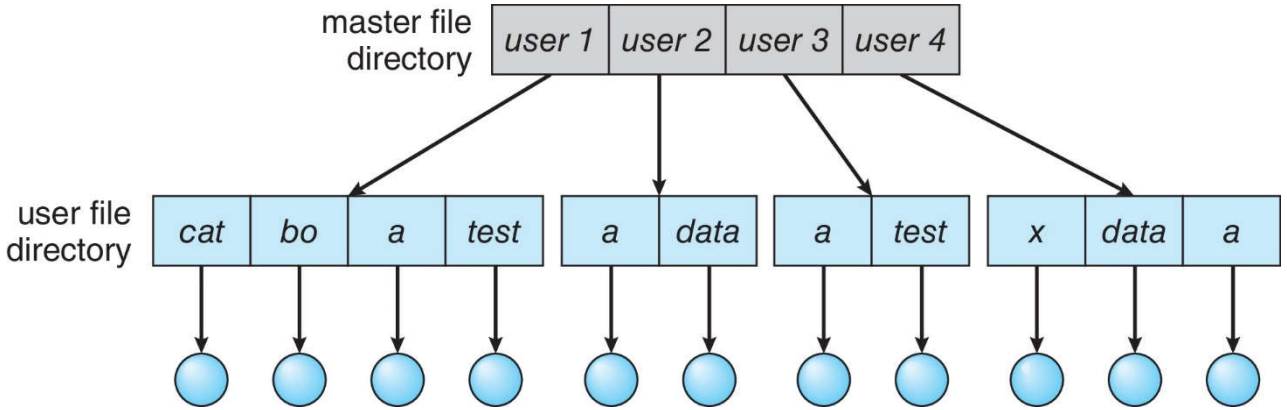
File Directories

- Single-Level Directory
  - A single directory for all users.
  - Common problems
    - Eventual length of directory
    - Giving unique names to files
    - Remembering names of files
    - Grouping of files (use file extensions).



File Directories

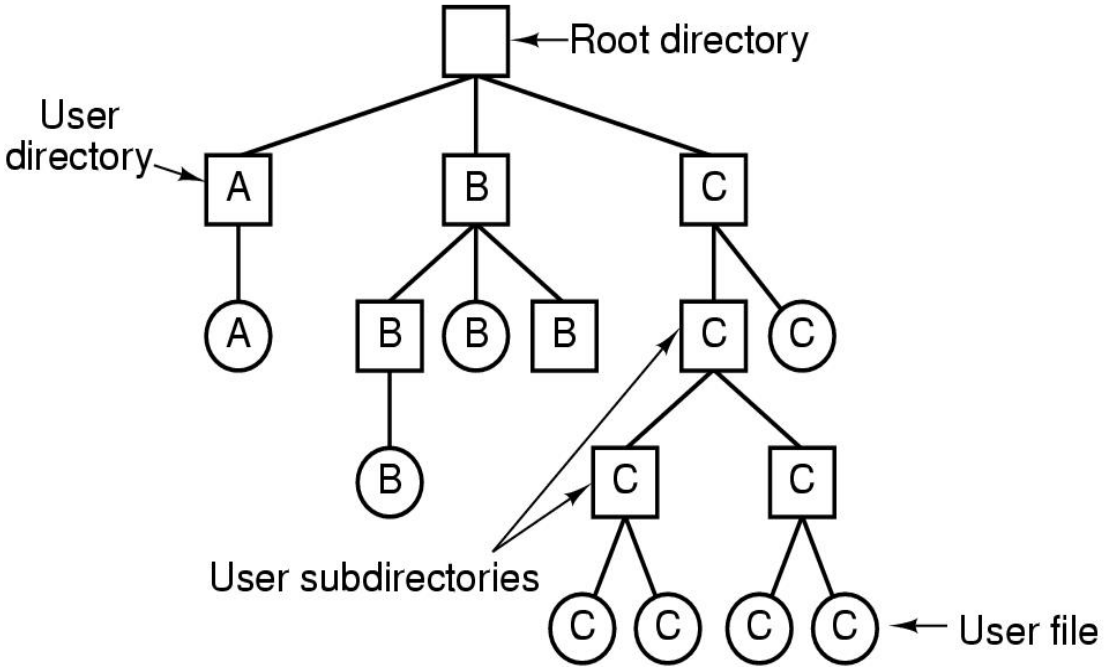
- Two-Level Directory
  - Separates directory for each user
    - Makes use of path name
    - Can have the same file name for different users
    - Provides efficient searching
  - No grouping capability



  - Main problem: violates zero-one-infinity principle.
    - Zero-One-Infinity principle (ZOI): "Allow none of foo, one of foo, or any number of foo.“ - by *Willem van der Poel*

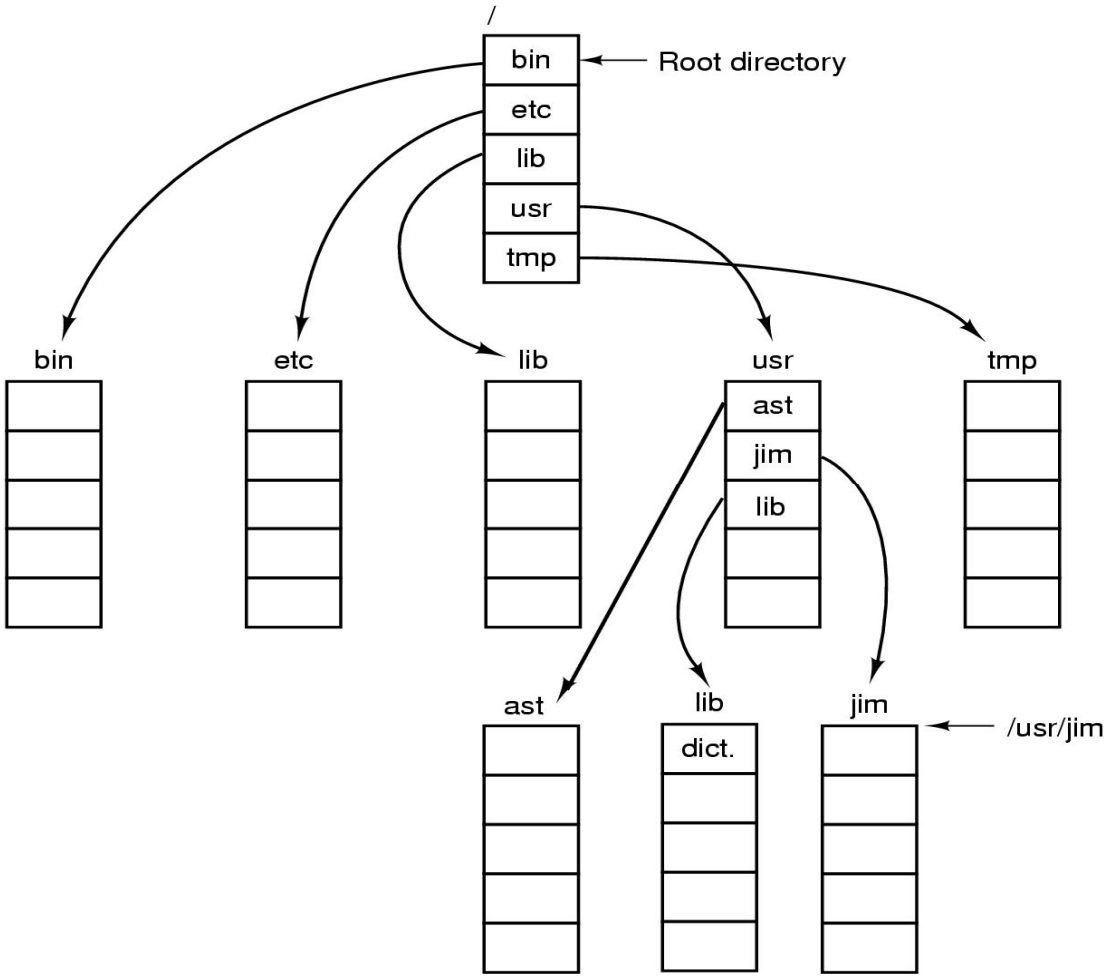
File Directories

- Tree-Structured Directories
  - Tree-Structured Directory Components.



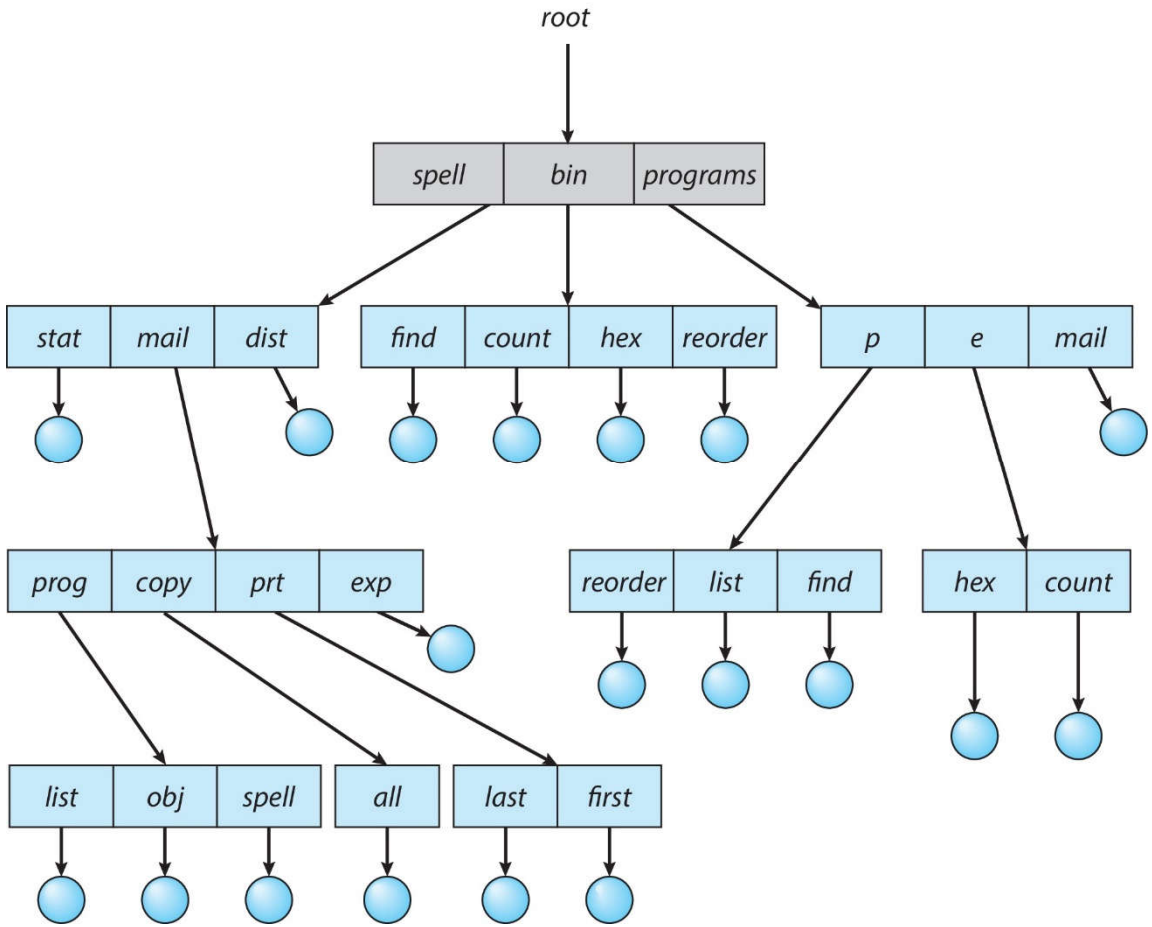
File Directories

- Tree-Structured Directories
  - Directory Path Names.



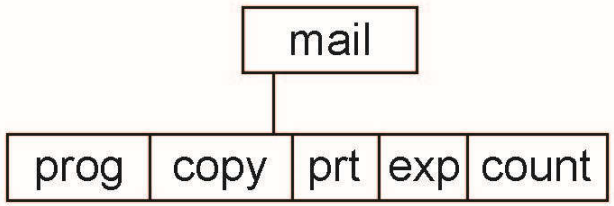
File Directories

- Tree-Structured Directories.



File Directories

- Tree-Structured Directories
  - Provides efficient searching.
  - Has grouping capability.
  - Current directory (working directory)




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

  - Makes use of *absolute* or *relative* path name
  - Creating a new file is done in current directory.

```
$touch <file-name>
```

  - Deleting a file

```
$rm <file-name>
```

  - Creating a new subdirectory is done in current directory

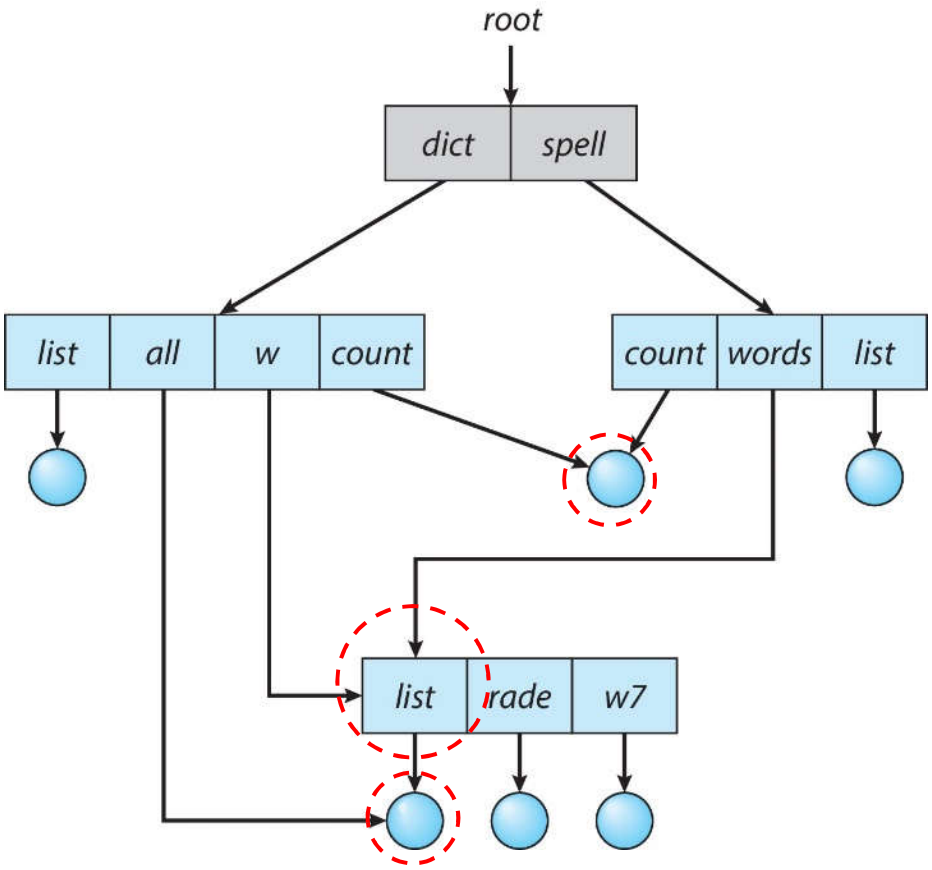
```
$mkdir <dir-name>
```

  - Deleting “mail” ⇒ deleting the entire subtree rooted by “mail”

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

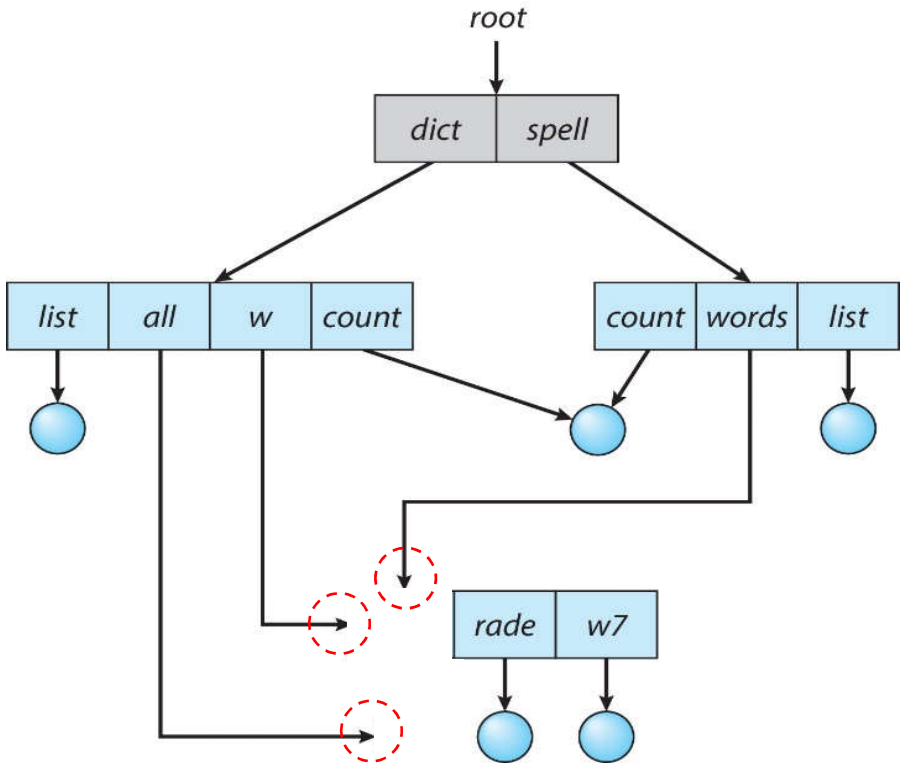
File Directories

- Directed Acyclic Graph (DAG) Directories
  - Have *shared* subdirectories and files.
    - An entry may have two different names with path (aliasing).



File Directories

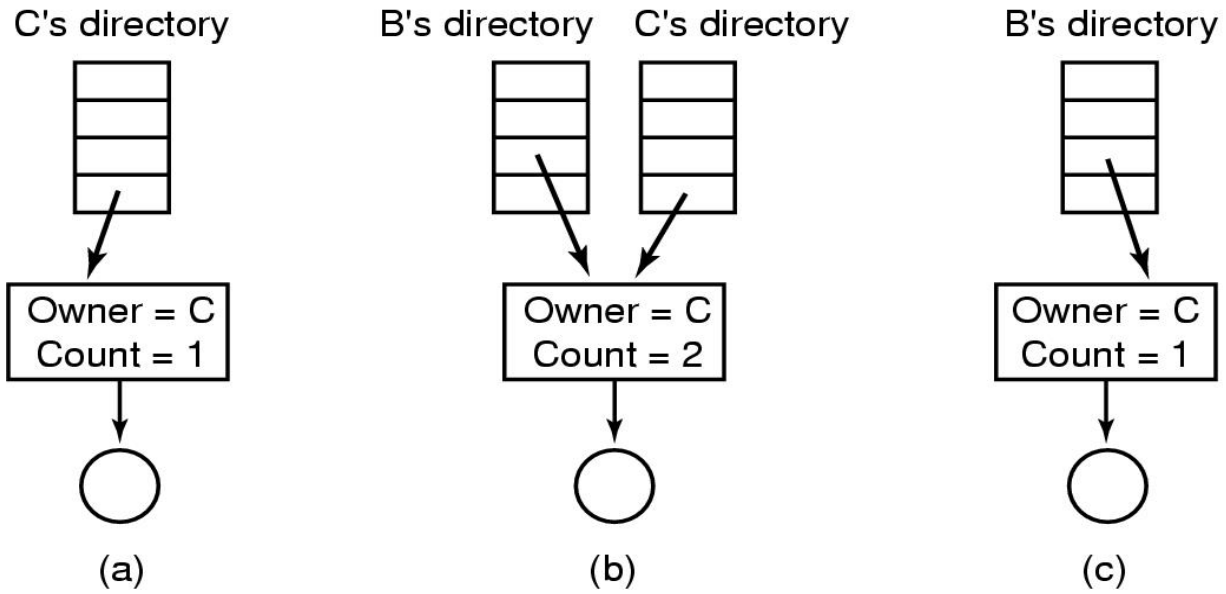
- Directed Acyclic Graph (DAG) Directories
  - If */dict/w/list* is deleted ⇒ some dangling pointers occur.



  - Newer type of directory entry:
    - *Link* – another name (pointer) to an existing file.
    - *Resolve the link* – follow the pointer to locate the file.

File Directories

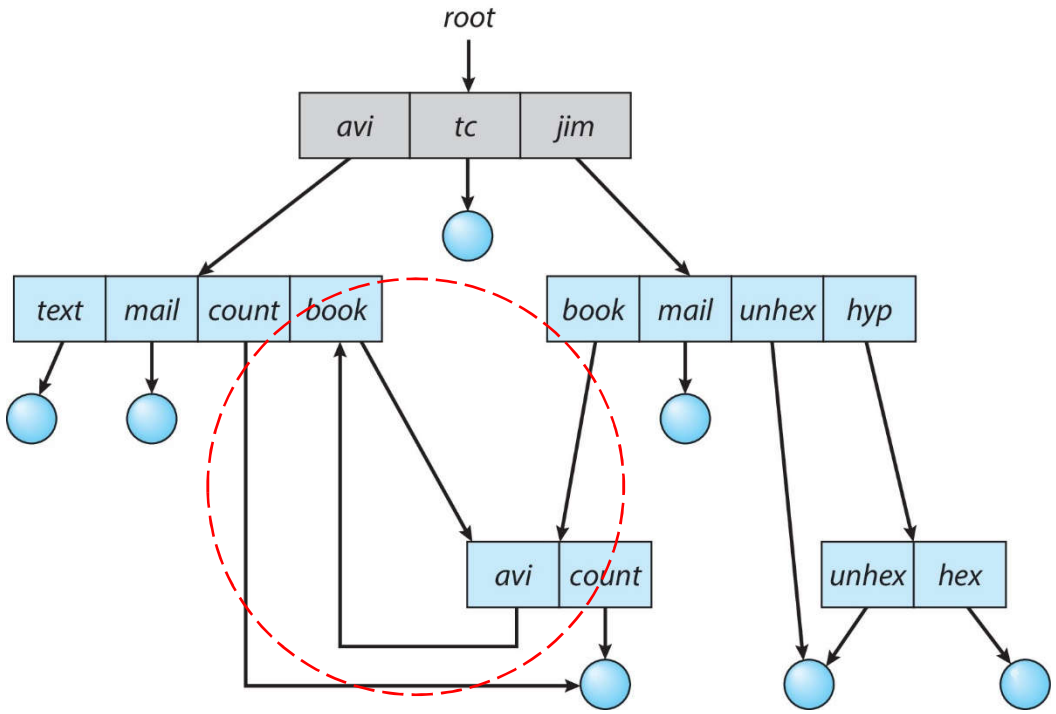
- Directed Acyclic Graph (DAG) Directories
  - Use a shared counter.



(a) Situation prior to linking.
(b) After the link is created.
(c) After the original owner removes the file.

File Directories

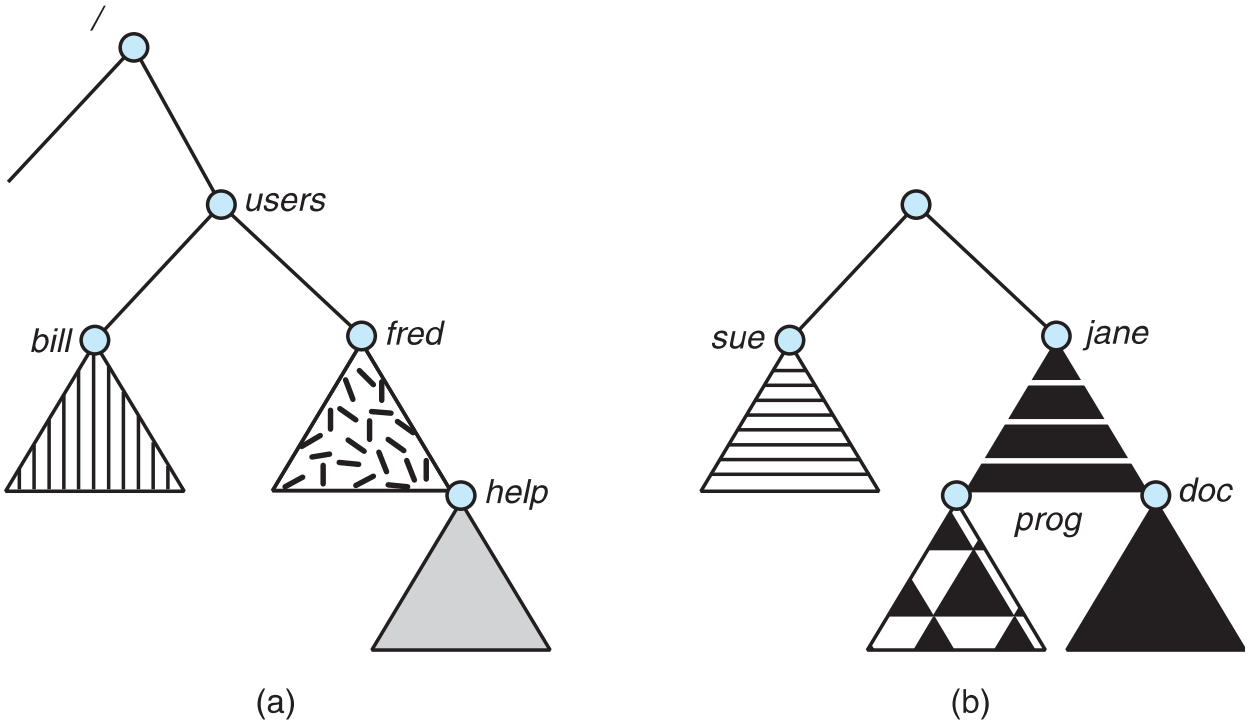
- General Graph Directory.



- How do we guarantee no cycles?
  - Allow only links to file but not subdirectories
  - Garbage collection
  - Every time a new link is added, use a cycle detection algorithm to determine whether it is OK.

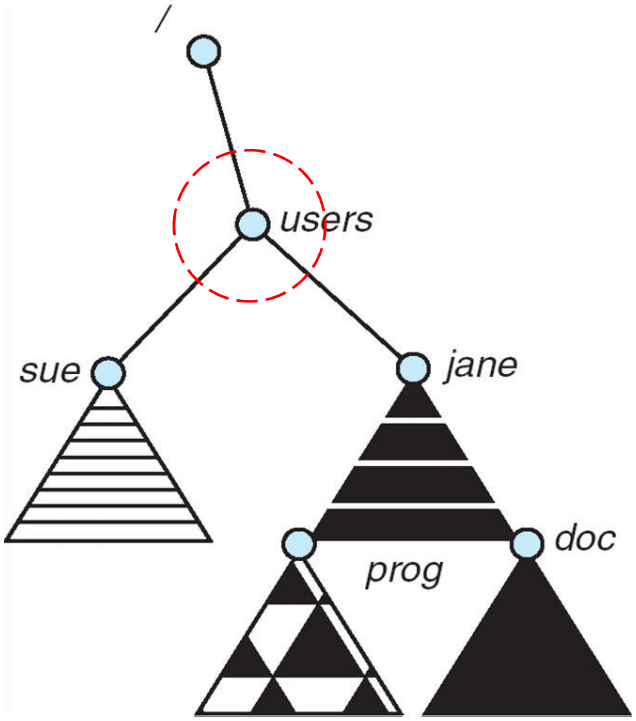
File System Mounting

- A file system must be mounted (挂载) before it can be accessed.
- An unmounted file system (i.e., Fig. b) is mounted at a mount point.



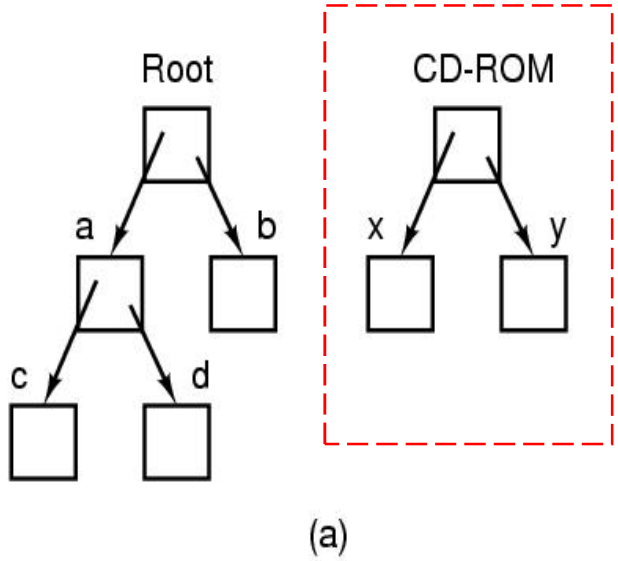
File System Mounting

- An unmounted file system (i.e., Fig. b) is mounted at a mount point.

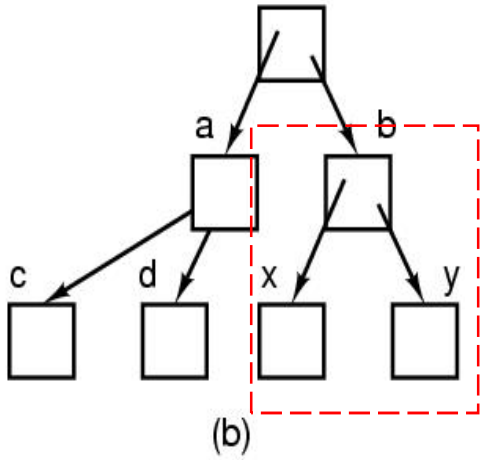


File System Mounting

- Example.



(a) Before mounting



(b) Mounted system

File Sharing

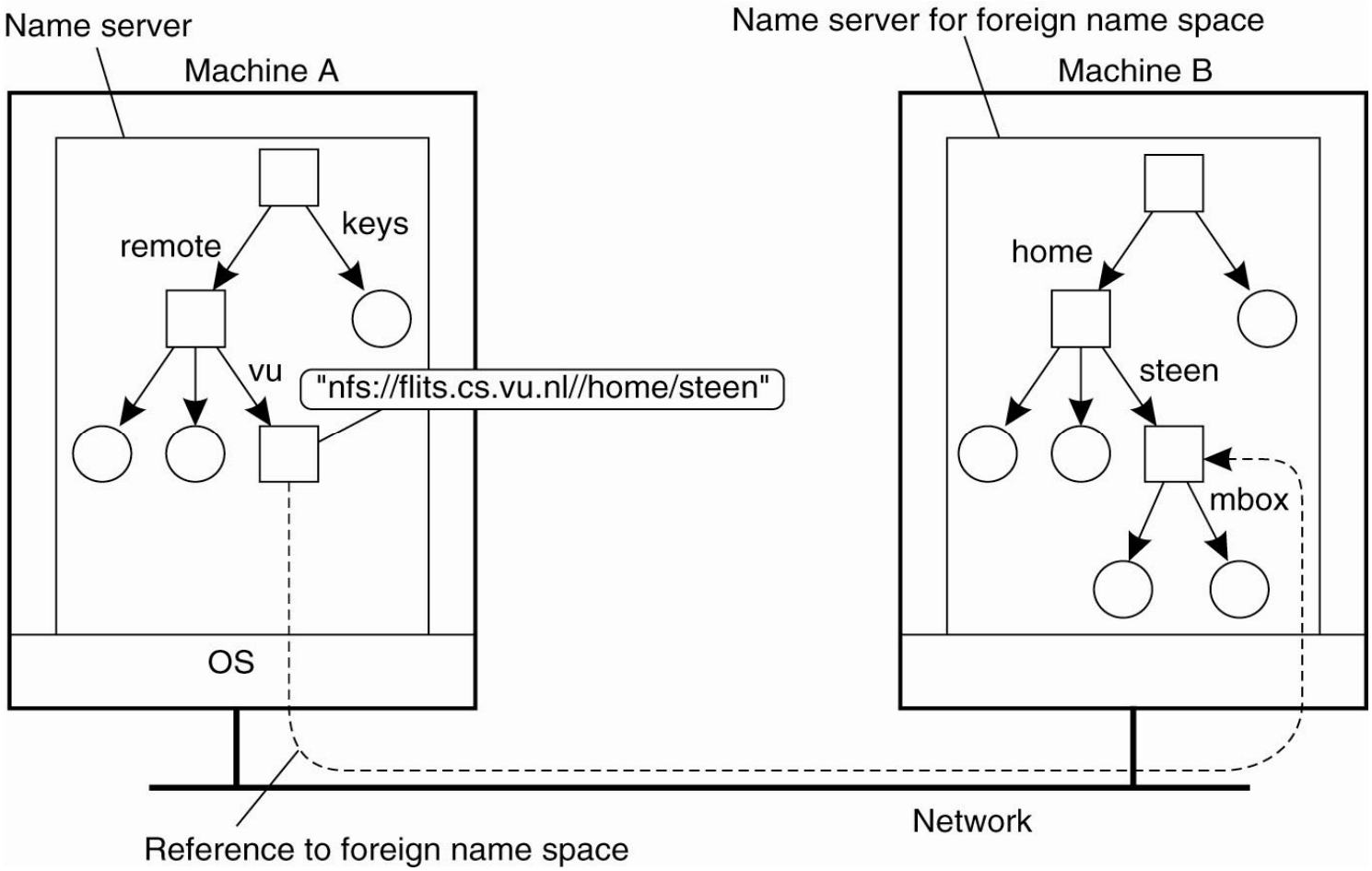
- Sharing of files on multi-user systems is desirable.
  - *User IDs* identify users, allowing permissions and protections to be per-user.
  - *Group IDs* allow users to be in groups, permitting group access rights.
  - Owner/Group are attributes of a file/directory.
- Sharing may be done through a *protection* scheme.
- On distributed systems, files may be shared across a network.
  - Network File System (NFS) is a common distributed file-sharing method.

File Sharing

- Remote File Systems
  - Uses networking to allow file system access between systems
    - Manually via command line like `$ftp` in Linux.
    - Automatically, seamlessly (无缝) using *distributed file systems*.
    - Semi automatically via the *Web*.
  - Client-Server model allows clients to mount remote file systems from servers.
    - Server can serve multiple clients.
    - Client & user-on-client identification is insecure/complicated.
    - *NFS* (Network File System) is a standard UNIX client-server file sharing protocol.
    - *CIFS* (Common Internet File System) is a standard Windows protocol.
    - Standard operating system file calls are translated into remote calls.
    - Distributed Information Systems (distributed naming services) such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing.

File Sharing

- Mounting Remote Name Spaces.



File Sharing

- Failure Modes in Remote File Systems
  - All file systems have failure modes.
    - For example, corruption of directory structures or *metadata* (non-user data).
  - Remote file systems add new failure modes, due to network failure, server failure.
  - Recovery from failure can involve *state information* about status of each remote request.
  - *Stateless* protocols such as NFS v3 include all information in each request, allowing easy recovery but less security.

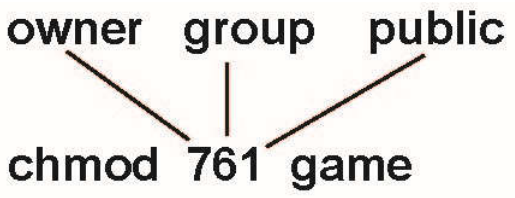
File Protection

- File owner/creator should be able to control:
  - what can be done, and
  - by whom.
- Types of file access
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List
  - Attribute change
- Access Control

File Protection

- Mode of access: Read, Write, eXecute.
- Three classes of users on Unix / Linux: Owner, Group and Public.

	Mode	R	W	X
Owner access	7	1	1	1
Group access	6	1	1	0
Public access	1	0	0	1



- Example
  - Ask manager to create a group (unique name), say G, and add some users to the group.
  - For a particular file (say *game*) or subdirectory, define an mode for appropriate access.

```
$chmod 761 game
```

  - Attach a group to the file.

```
$chgrp G game
```

File Protection

- A Sample Linux Directory Listing.

```
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
```

- The first field describes the protection of the file or directory. A d as the first character indicates a subdirectory. Also shown are the number of links to the file, the owner's name, the group's name, the size of the file in bytes, the date of last modification, and finally the file's name.

File Protection

- Windows 10 Access-Control List Management.