File System

Instructor: Dr Tarunpreet Bhatia

Assistant Professor

CSED, TIET

Disclaimer

This is NOT A COPYRIGHT MATERIAL

***<u>Content has been taken mainly from the following books</u>*:**

Operating Systems Concepts By Silberschatz & Galvin,
Operating Systems: Internals and Design Principles By William Stallings

File Concept

- Contiguous *Logical Address Space*
- Types:
  - Data
    - numeric
    - character
    - Binary
  - Program

File Attributes

- Name – only information kept in human-readable form
- Identifier – Unique tag (number) identifies file within file system
- Type – Needed for systems that support different types
- Location – Pointer to file location on device
- Size – Current file size
- Protection – Controls who can do reading, writing, executing
- Time, Date, and user Identification – Data for protection, security, and usage monitoring
- Information about files are kept in the directory structure, which is maintained on the disk

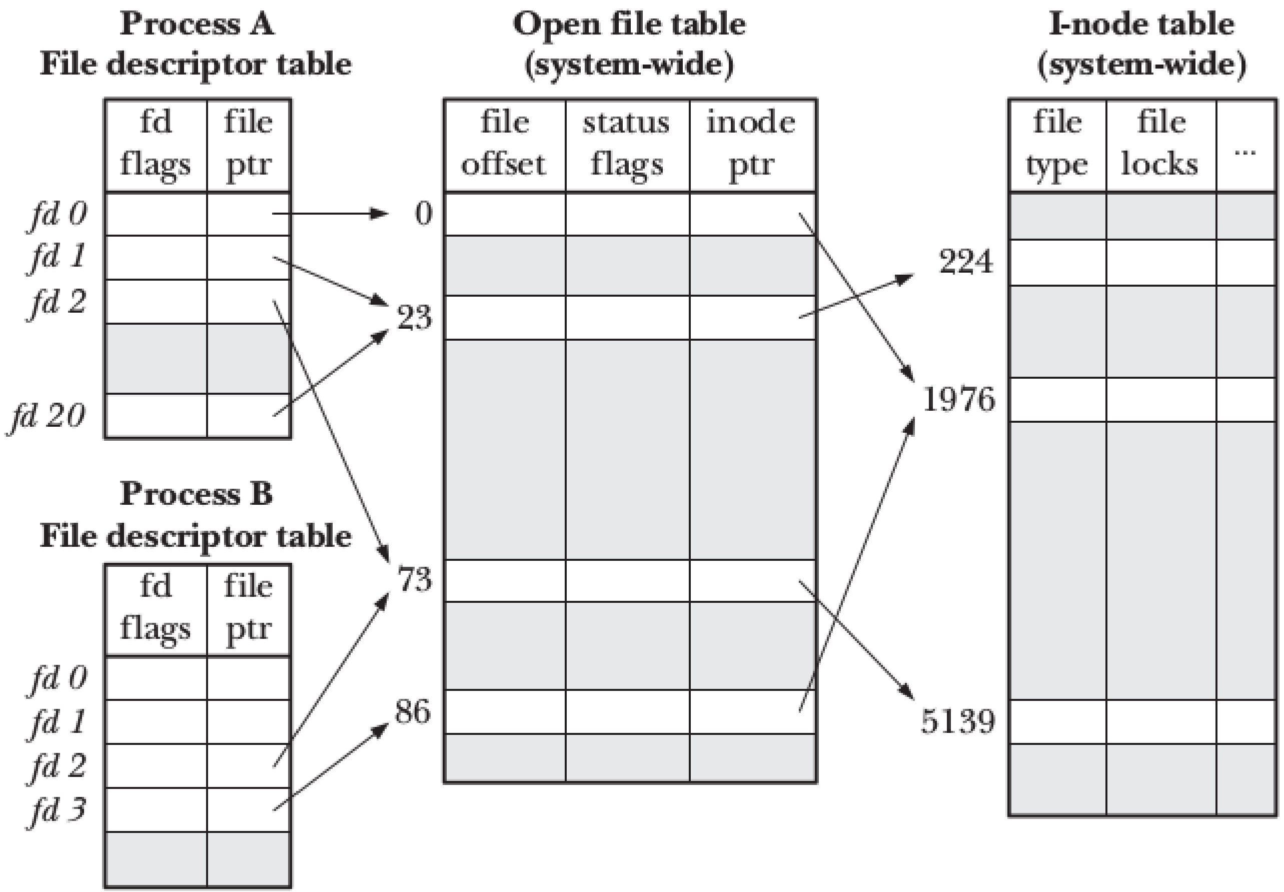
File Operations

- Create
- Write
- Read
- Reposition within File
- Delete
- Truncate

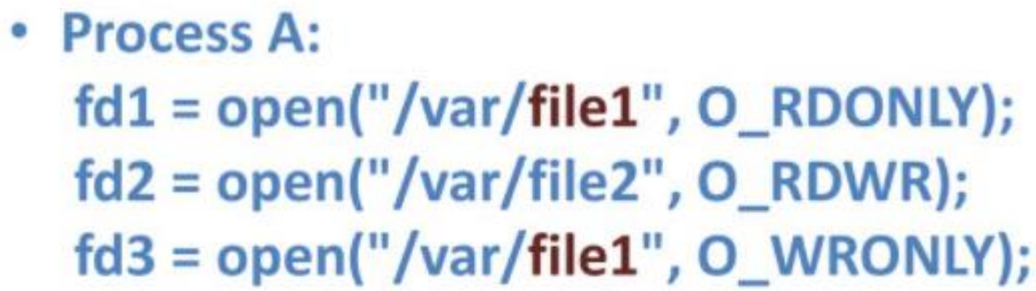
- Open (F_i) – Search the directory structure on disk for entry F_i, and move the content of entry to memory

- Close (F_i) – Move the content of entry F_i in memory to directory structure on disk

File Descriptor, File Table Entry and i-node Entry



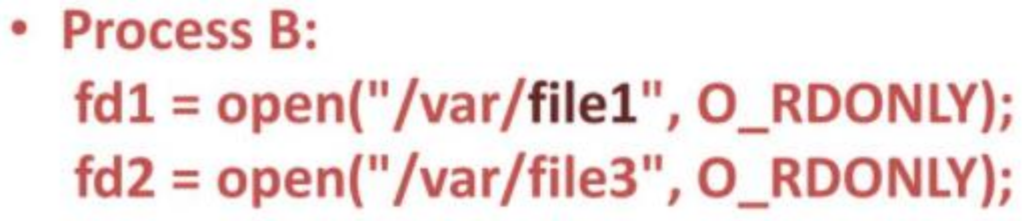
Interaction between these tables



- Process A:

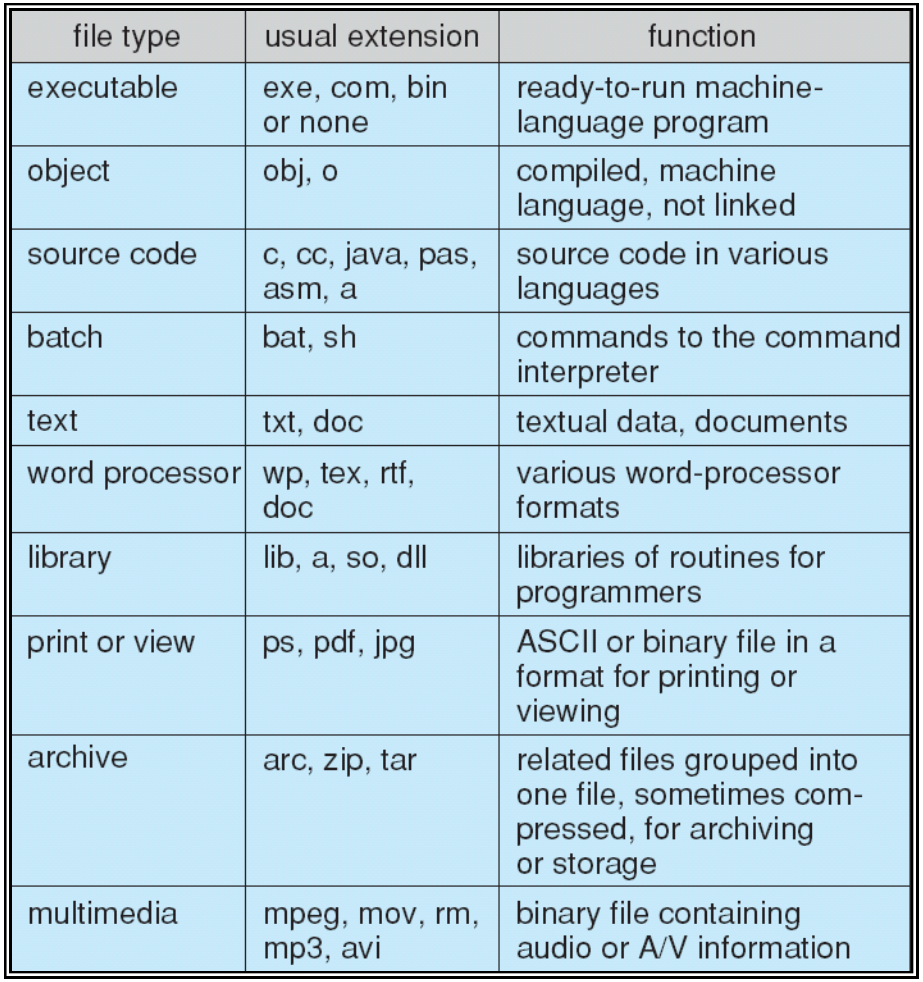
```
fd1 = open("/var/file1", O_RDONLY);
fd2 = open("/var/file2", O_RDWR);
fd3 = open("/var/file1", O_WRONLY);
```

•



- Process B:

```
fd1 = open("/var/file1", O_RDONLY);
fd2 = open("/var/file3", O_RDONLY);
```



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, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

Access Methods

- *Sequential Access*

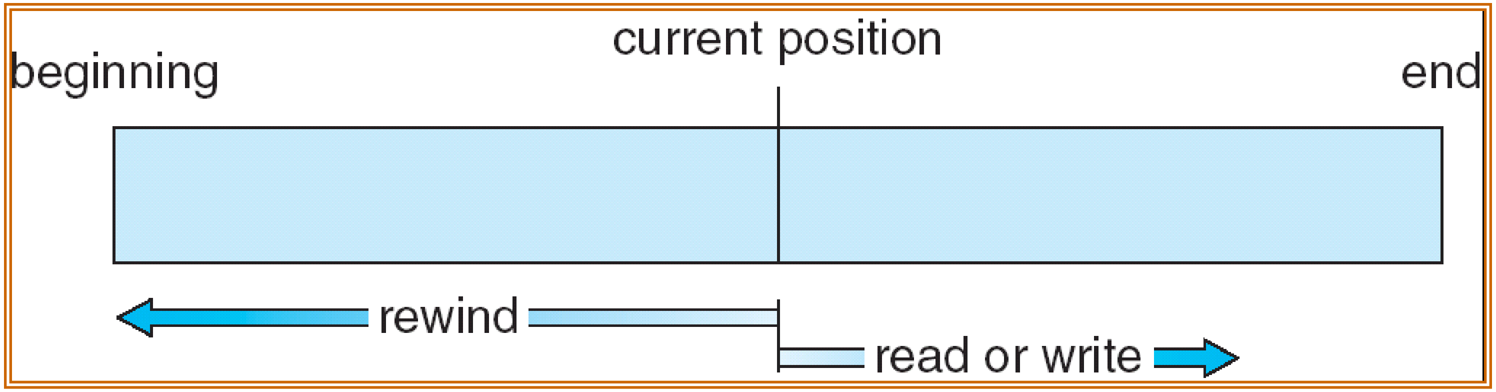
  read next
  write next
  reset

- *Direct Access or Relative Access*

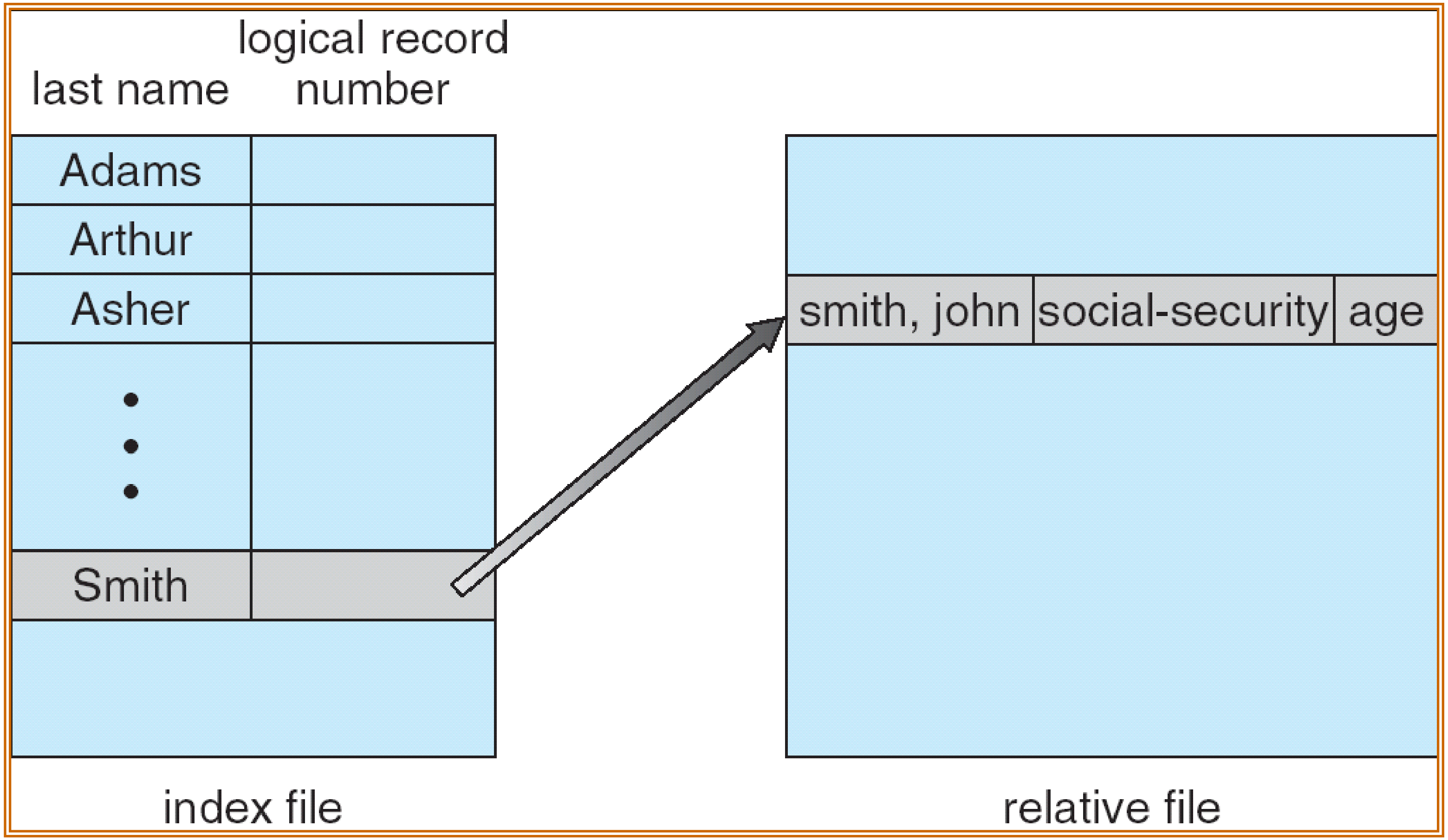
  read n
  write n
  position to n
    read next
    write next
  rewrite n

n = relative block number

Sequential Access

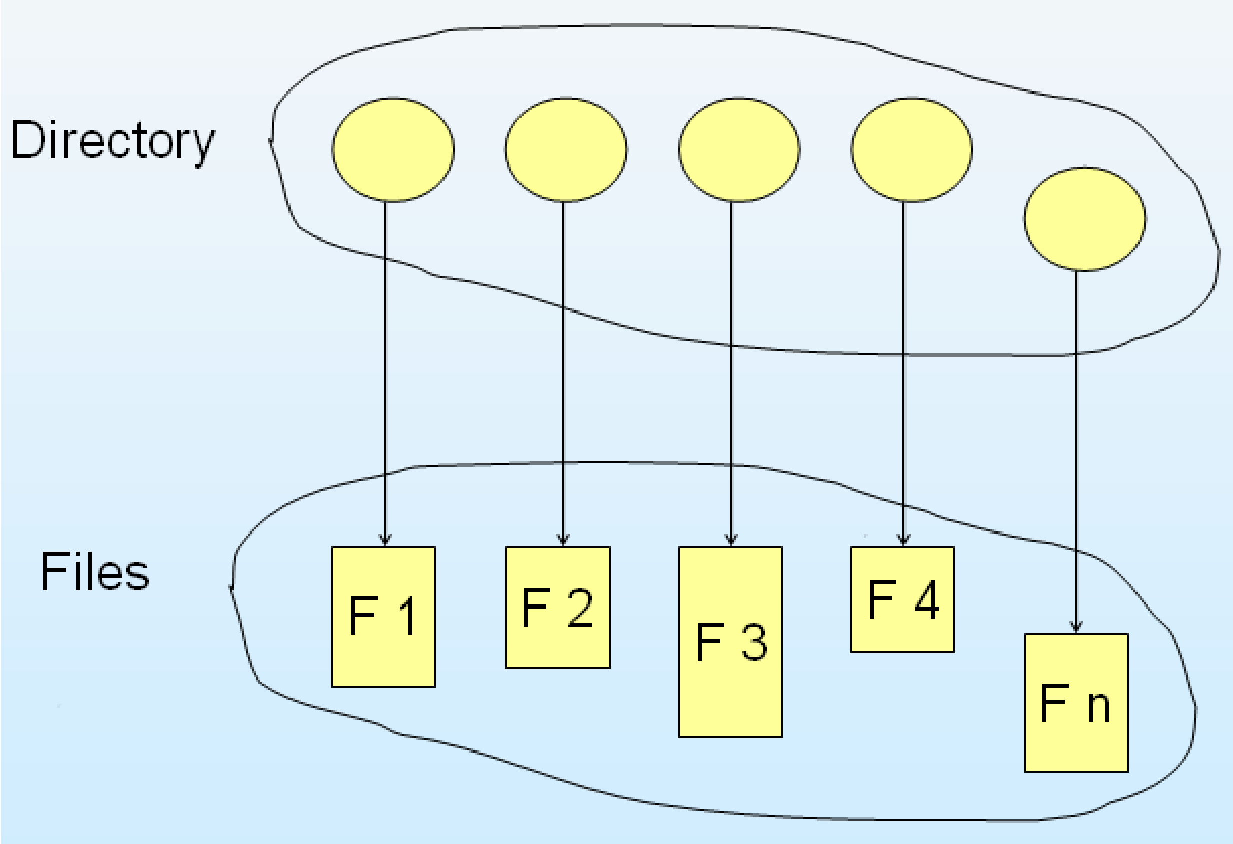


Index & Relative Files

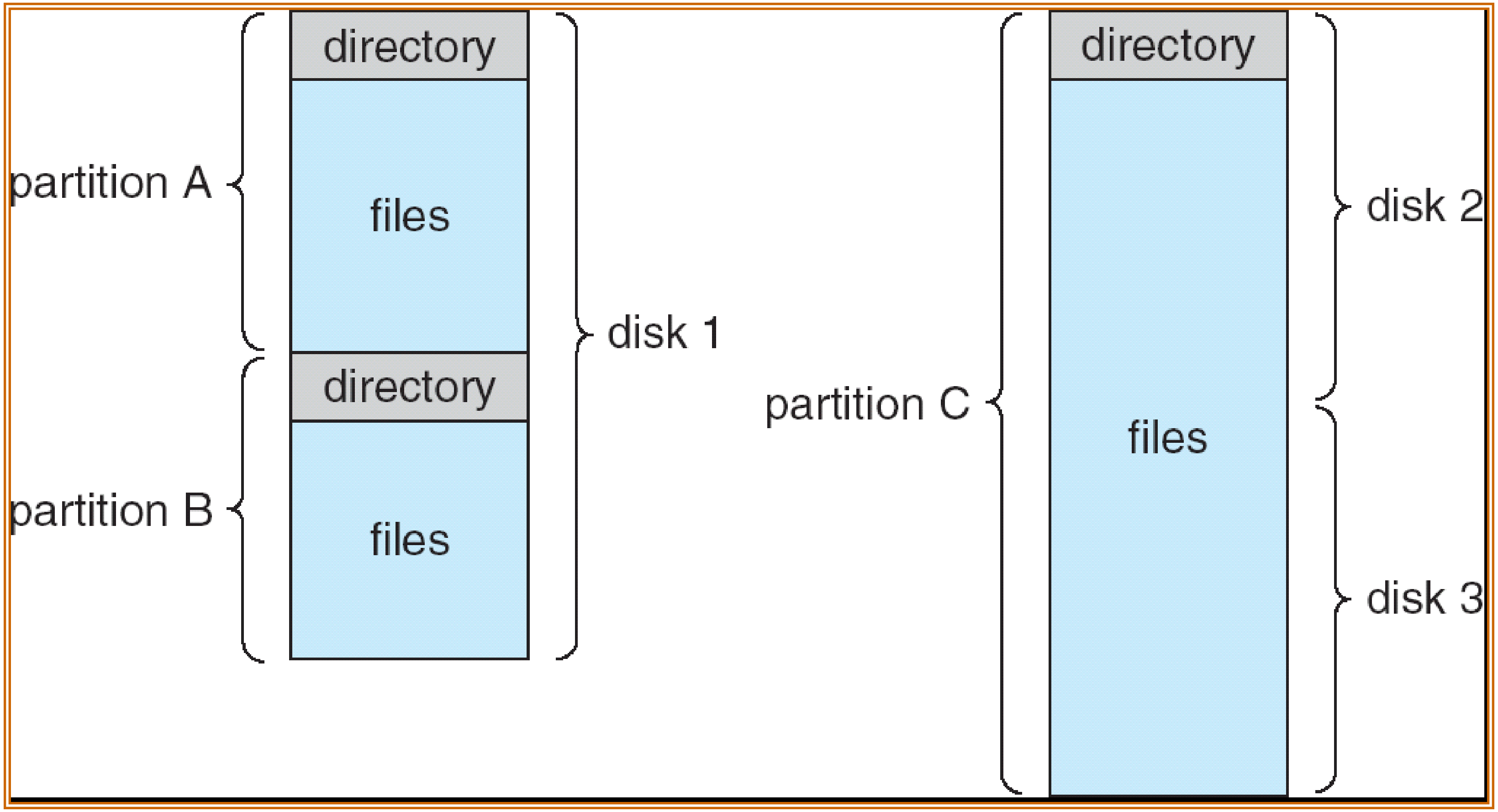


Directory Structure

- A collection of nodes containing information about all files



File System Organization



Operations performed on Directory

- Search for a File
- Create a File
- Delete a File
- List a Directory
- Rename a File
- Traverse the File System

Organize directory to obtain

- Efficiency – Locating a file quickly
- Naming – Convenient to users
  - Two users can have same name for different files
  - The same file can have several different names
- Grouping – Logical grouping of files by properties, (e.g., all Java programs, all games, …)

Single-level Directory

- In Single Level Directory all files are in the same directory.

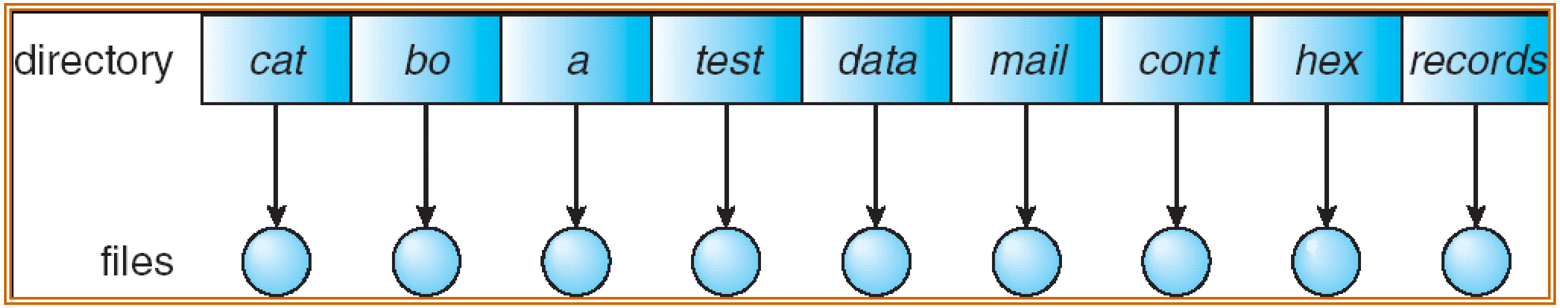
Advantages

- Since it is a single directory, so its implementation is very easy.
- The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

Disadvantages

- There may chance of name collision because two files can not have the same name.
- Even a single user may find it difficult to remember the names of all files as the number of file increases.
- Searching will be time consuming if the directory is large.
- This can not group the same type of files together.

Single-level Directory



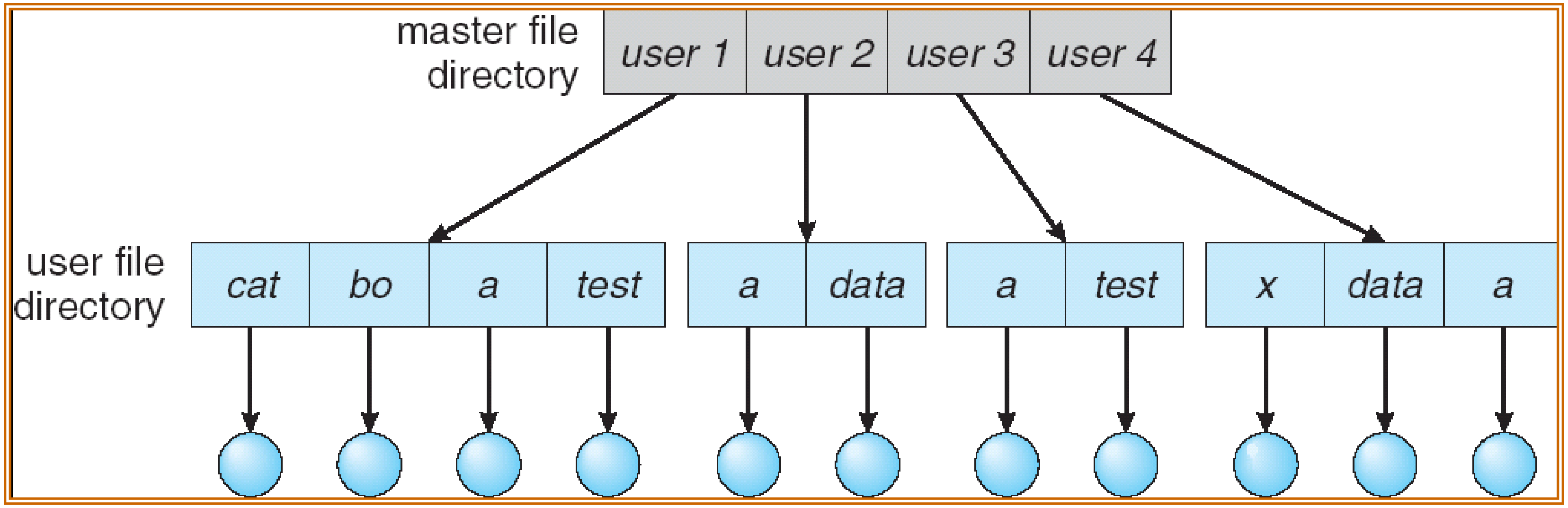
Two-level Directory

i) Each user has Its own User File Directory (UFD).
ii) When the user job start or user log in, the system Master File Directory (MFD) is searched. MFD is indexed by user name or Account Number.
iii) When user refers to a particular file, only his own UFD is searched.

A two level directory can be a tree or an inverted tree of height 2

The root of a tree is Master File Directory (MFD).
Its direct descendents are User File Directory (UFD). The descendents of UFD's are file themselves. The files are the leaves of the tree.

Two-level Directory



Two-level Directory

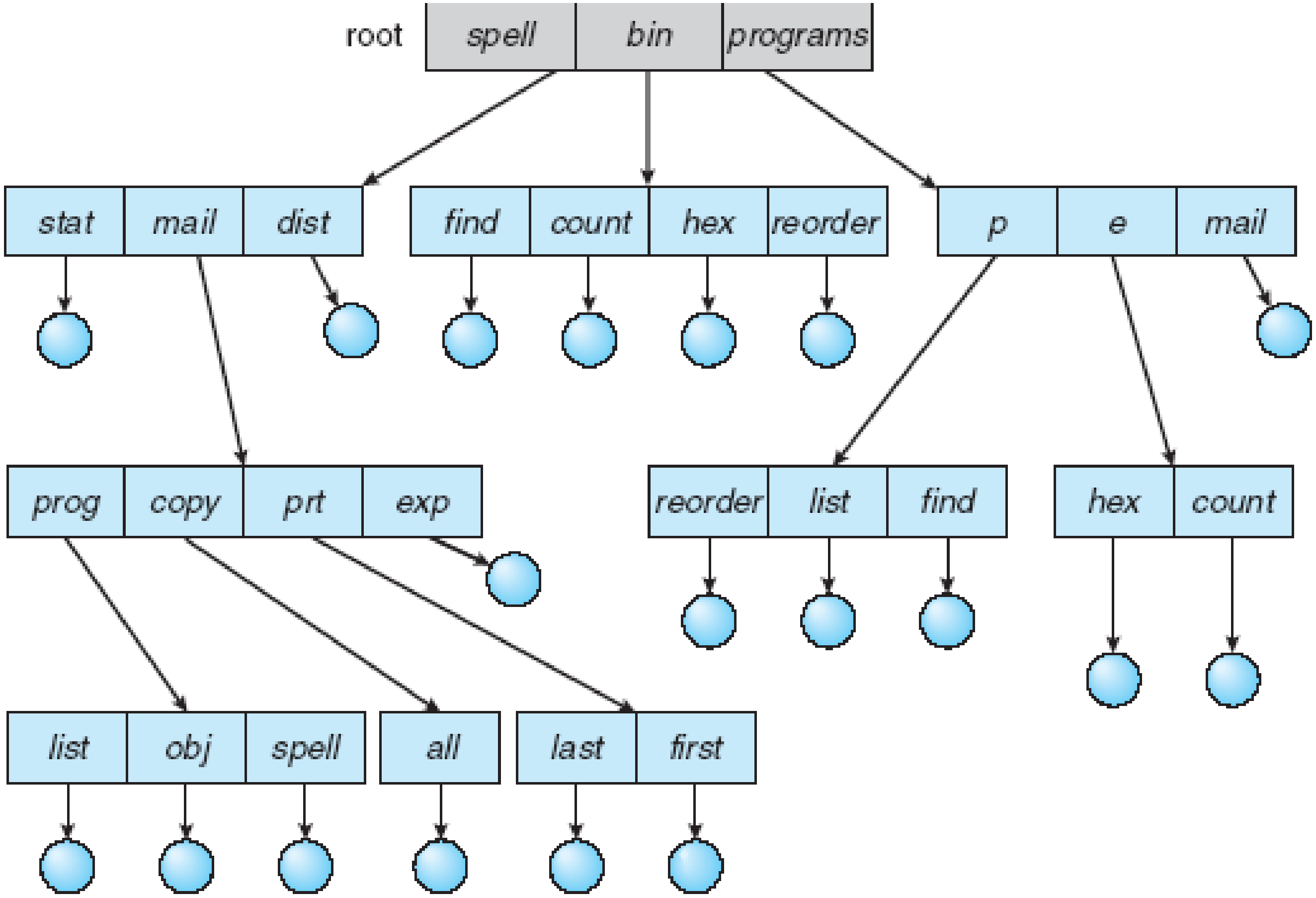
Advantages

- We can give full path like /User-name/directory-name/.
- Different users can have the same directory as well as the file name.
- Searching of files becomes easier due to pathname and user-grouping.

Disadvantages

- The structure effectively isolates one user from another. A user is not allowed to share files with other users.
- Still, it not very scalable, two files of the same type cannot be grouped together in the same user.

Tree Structured Directory



Tree Structured Directory

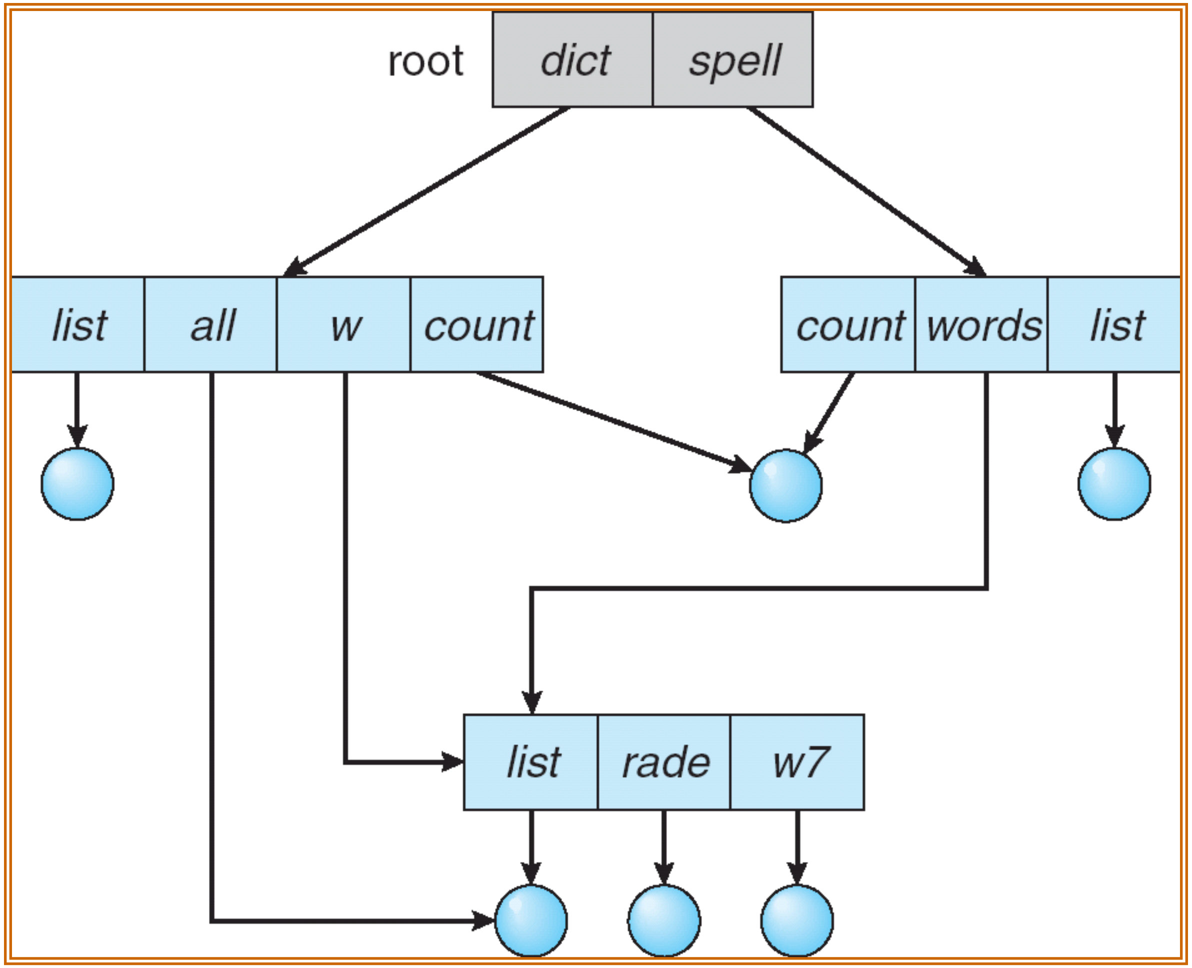
Advantages

- Very general, since full pathname can be given.
- Very scalable, the probability of name collision is less.
- Searching becomes very easy, we can use both absolute paths as well as relative.

Disadvantages

- Every file does not fit into the hierarchical model, files may be saved into multiple directories.
- We can not share files.
- It is inefficient, because accessing a file may go under multiple directories.

Acyclic Graph Directory



Acyclic Graph is the graph with no cycles. It allows directories to share sub directories and files. With a shared file, only one actual file exists, so any changes made by one person are immediately visible to the another.

Implementation of Shared files and directories

i) *To create a link*

A link is effectively a pointer to another file or sub directory.

Duplicate all the information about them in both sharing directories.

ii) *Deleting a link*

Deletion of the link will not effect the original file, only link is removed.

To preserve the file until all references to it are deleted.

Issues In Acyclic Graph Directories

- A file may have multiple path names
- Consequently distinct file names may refer to the same file
- When traversing the entire file system we may come across shared structures more than once

  Ex: when we copy all files to backup storage same file may get copied more than once

Approach 1

- Remove the file whenever anyone deletes the shared file
- Problems with Approach 1
  - Problem is dangling pointers which point to the now-nonexistent file
  - If the space is subsequently reused for other files, these dangling pointers may point into the middle of other files

How to handle problems with Approach 1?

- Easy to handle if sharing is implemented by symbolic links
- If file entry itself is deleted, then space is de allocated, leaving the links dangling. We can search for dangling links and remove them. The search can be expensive unless a list of the associated links is kept with each file.
- Alternatively, we can leave the links until an attempt is made to use them.
- At that time, we can determine the file name given by the symbolic link does not exist and can fail to resolve the link name

Approach 2

- Preserve the file until all references to it are deleted
- To implement this approach, we need a mechanism to determine that the last reference is deleted.
- A list of all references to a file can be kept.

Problems with Approach 2

- Variable and potentially large size of the file-reference list

How to handle problems with Approach 2?

- We need not keep the entire list
- Keep only a count of number of references
- A new link or directory entry increments the reference count
- Deleting a link or entry decrements the count
- When count is zero, the file is physically deleted

Acyclic Graph Directory

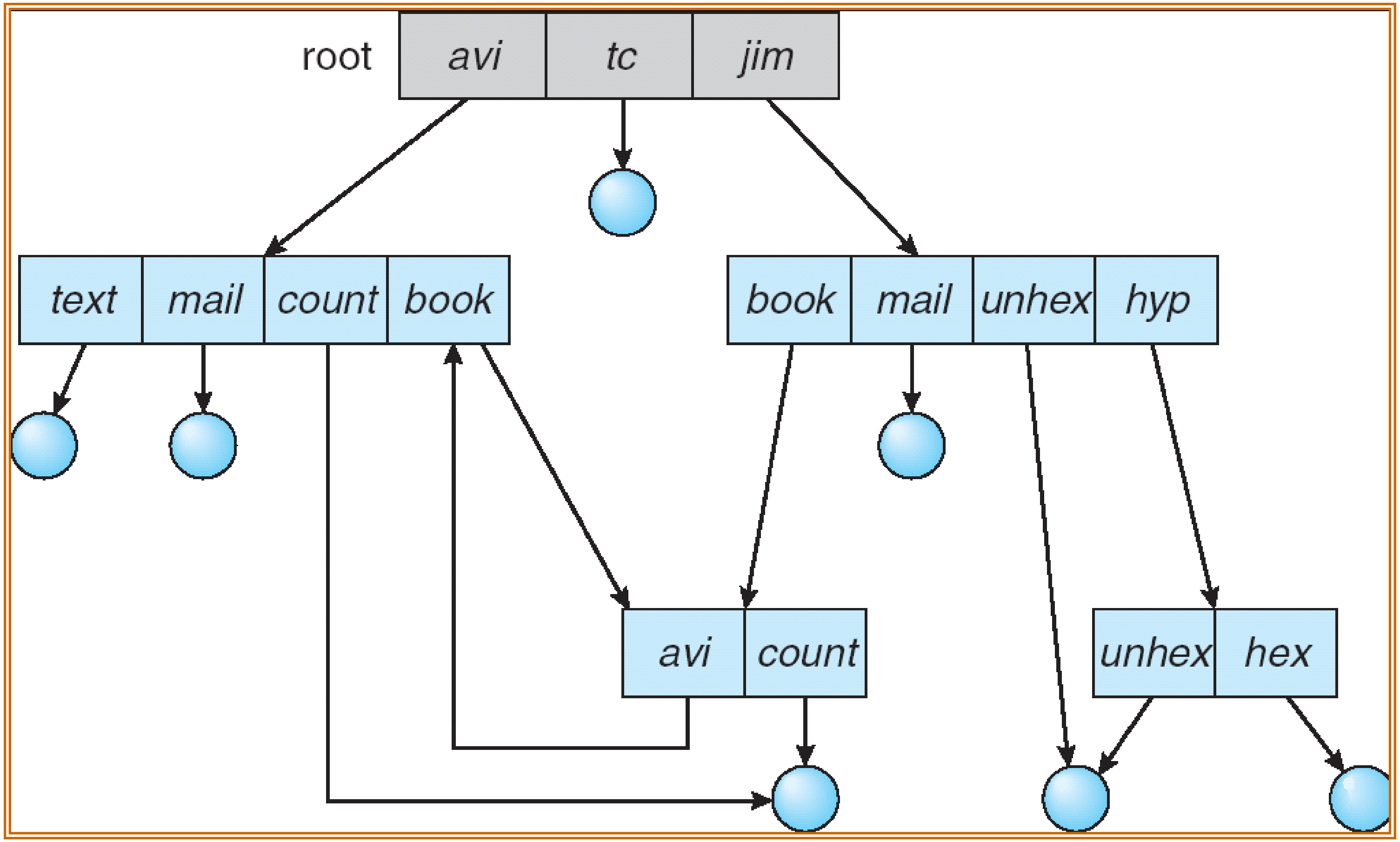
Advantages

- We can share files.
- Searching is easy due to different-different paths.

Disadvantages

- We share the files via linking, in case deleting it may create the problem,
- If the link is a soft link then after deleting the file we left with a dangling pointer.
- In the case of a hard link, to delete a file we have to delete all the references associated with it.

General Graph Directory



General Graph Directory

Advantages

- It allows cycles.
- It is more flexible than other directories structure.

Disadvantages

- It is more costly than others.
- It needs garbage collection.

Protection

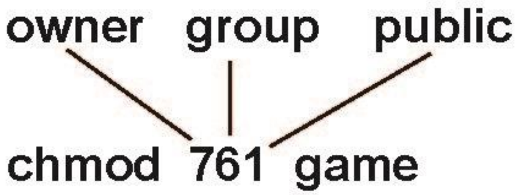
- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

Access Lists and Groups

- Mode of access: read, write, execute
- Three classes of users on Unix / Linux

			RWX
a) **owner access**	7	⇒	1 1 1
			RWX
b) **group access**	6	⇒	1 1 0
			RWX
c) **public access**	1	⇒	0 0 1

- 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 appropriate access.



owner group public

```
chmod 761 game
```

Attach a group to a file

```
chgrp G game
```