Computer Systems [CS 5600] | November 3, 2010 | Lecture Notes - 8:00 PM to 9:00 PM

Scribe Notes by Komal Sodha

Sequential & mutual access to a file is common and hence Kernel needs support for file locking

File locking – 2 types

Exclusive locks – writing purpose

Shared locks – sufficient for reading concurrently

File Structure

What is File? – It's a named collection of related data

Different structure of file

- None (array of bytes which is not much useful)
- Records (sort of lines)
- Fixed width format (each record have same size, to access ith record, locate at (record size * i))
- Hierarchical (e.g. XML file)
- Other formats (e.g. pdf)

- ✓ How to linearize your data?
- ✓ Need some data structure to have file structure in place
- ✓ Serialization
- ✓ OS should know certain formats like executable.
- ✓ Each file structure has some rules

Should OS "know about" (able to interpret) file structure?

Advantage

Reduces complexity if OS knows the data structure

Verify correct format

Provide services like .o & .c files comes together

Disadvantage

Less flexible, if OS knows format, you can't store invalid XML file

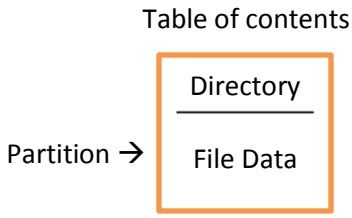
Add significant complexity

User level file structures changes often, if implemented in OS, need to update it e.g. XML different versions

File Organization

- ✓ Partition - division of physical storage device
- ✓ Each partition uses a specific file system format e.g. XFS ZFS,
- ✓ Mount a partition similar to open
- ✓ Must mount before use

Within partition



File Directory

Entries with human readable file names

How do we organize file entries?

What operations do we perform ?

1. Insert file entries
2. Remove file entries
3. Search for file name i.e. find file foo.txt in the directory
4. List files
5. Modify attributes
6. Rename file

Need to take care of above operations while designing directory structure

Different Design for Directory Structure

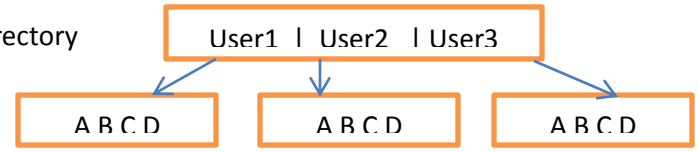
1. Single Level directory

List may be unacceptable coz accessing files is slow -> linear search

Files must be uniquely named - across users

Number for file identifies is not feasible to user

2. Two level directory



Solving naming problem but introduces sharing problems

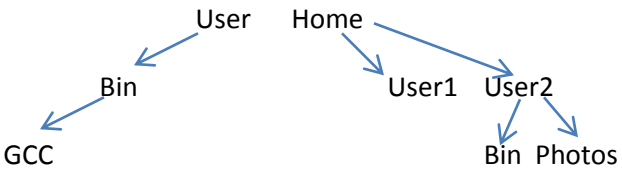
How to allow processes to share files

To solve, introduce Paths e.g. "user1/foo"

- ✓ Default search path
- ✓ Absolute & Relative Path

Common files put copy in every directory

3. Tree Directories



Searching for a file is not linear, hence increases performance

Subdirectories implemented as files

Current directory

Preprocess

Each process has not open of current directory, coz process need to refer other directories

System calls to change current directory i.e. pwd

Implement delete of a Directory

If not empty, delete everything recursively - easy solution

tradeoff b/t safety and convenience

OS keeps flags for decision

Difference : Disk file points to disk file Vs. Disk file points to data

Two files point to same link is sometimes useful

DAG - Directory Acyclic Graph

Soft links

- ✓ A pointer to the shared file's name
- ✓ Deleting link doesn't delete file data
- ✓ what happes if file linking to is deleted/renamed?
- ✓ link may be the same but data is changed

Hard links

- ✓ A pointer to same data as another FCB - file control block
- ✓ Just like a normal file
- ✓ What happens if one of the FCB is deleted, data may lost
- ✓ Each data has a reference count, delete file only if reference is 1 i.e. no other is having it