UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 537
Intro to Operating Systems

A. Arpaci-Dusseau
Spring 2000

Introduction to Files

Questions answered in these notes

- What is a file and what operations can be performed on them?
- What information (meta-data) must be associated with a file?
- How are directories organized?
- What is the difference between hard and soft links?

Reading for this topic: Chapter 10

What is a File?

User view

- Named collection of bytes
  - Type defined by creator
  - Example: Text, source, object, executables
- Permanently and conveniently available

Operating System view

- Map bytes as collection of blocks on physical storage
- Nonvolatile storage device
  - Magnetic disks
  - Persistent across reboots and power failures

Role of File System

Naming

- How do users select files?
- Convert “name” + offset --> logical disk block --> cylinder #, sec #

Protection

- All users are not equal

Reliability

- Information must last safely for long periods of time

Disk Management and Allocation

- Efficient use of disk space
- Fast access to files

File Operations

Create file

- Find space on disk
- Give name to file

Write or Read from named file

- Find current location in file (track per-file & per-process)
- Seek within file to desired location

Delete file

- Release disk space

Truncate

- Set size to 0, but keep other attributes

Rename

Change access permissions

Meta-Data

Additional system information on disk associated with each file

- Name of file, type of file
- Pointer to data on disk
- File size
- Most recent access and modification time
- Owner and group id
- Protection bits
- Special file?
  - Directory? Symbolic link?

UNIX Example: ls -lig midSol.ps

122945 -rw-r----- 1 dusseau cs140 2918 Nov 1 18:53 midSol.ps

Issue: How is meta-data stored on disk? (i-node in UNIX)

Opening Files

Expensive to access file by symbolic name

Open file before first access

- Search directories for file name and check permissions
- Copy relevant meta-data to open file table in memory
- Return index in open file table (“file descriptor”)
- Application passes index to OS when accessing file

Per-process open file table

- Current position pointer
- Index in system table

System open file table shared across processes

- File open count

Directories

Organization technique

- How to map file name to location of file on disk?
  - Index of meta-data in list

Single-Level directory

- Single directory for entire disk
  - Each file must have a unique name
- Special part of disk holds one directory
  - Contains <file name, index> pairs
  - Fixed length name
- Only appropriate for personal computers

Two-Level directory

- Directory for each user
- Specify file with user and file name

Tree-Structured Directories

Directory stored on disk just like regular file

- Data consists of <file name, index> pairs
  - File name can be another directory
- Special bit set in meta-data
  - User programs can read directories
  - Only special system programs can write directories
- Reference by separating names with slashes

Special directories

- Root: Fixed index for meta-data (2)
- . : This directory
- .. : Parent directory

Example: mkdir /a/b/c

2: Find <“a”, 5>

5: Find <“b”, 9>

9: Verify no <“c”, X> previously; add “c” to directory

Acyclic-Graph Directories

Create links from one file/directory to another

Hard link: “ln a b” (“a” must exist already)

- Multiple directory entries with same index number
- Remove one file --> Decrement reference count
- Deallocate space when reference count reaches 0
- Cannot refer to directories; Why?

Symbolic link: “ln -s a b”

- Special file, designated by bit in meta-data
- Contents of file contains name of another file
  - Remove original file --> links cannot be resolved when access
- Optimization: In directory entry, put filename instead of index

Path Names

Absolute path name (full path name)

- Start at root directory: /usr/spool/mail

Relative path name

- Cumbersome to specify full path
- Associate current working directory with every process
- Assume file is in current directory

Shortcuts

- Search path listed in environment variable
- Special variable: ~user/file and ~yourfile
- UNIX: Shell examines directories listed in environment variable
  - Drawback: Other applications must duplicate functionality

Directories as Files

Separate functionality into two levels

- Lowest-level: Storage management system
- Higher-level: Naming, directories

Advantages

- Simplifies design and implementation
- Optimizations in low-level file management -->
  Speeds up directory operations

Protection

Types of Access

- **Read, write, execute,** append, delete, list, ...

Access Control List

- Associate list of users with access rights for every file
- Advantage: Complete control
- Disadvantage:
  - Tedious to construct list (may not know in advance all users)
  - Requires variable-size information

Classify users

- Owner, Group, Universe
- Advantage: Easier to implement
- Disadvantage: Cannot exclude particular users