File Systems

Daniel Bosk[1]

Department of Information and Communication Systems (ICS),
Mid Sweden University, Sundsvall.

introfs.tex 235 2018-02-28 07:01:18Z jimahl

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

Literature

This lecture gives an overview of the technicalities of the file system. This is covered by Chapter 10 “File System” [SGG13a].

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

File Meta-Data and Operations

- A file is the smallest logical unit in the file system.
- Its contents is defined by its creator, hence the data inside can have any structure.
- From the perspective of the OS there is no difference between two files other than what can be seen in the meta-data, also known as attributes.

File Meta-Data and Operations

Attributes

Name The symbolic file name for humans.

Identifier A unique number identifying the file within the file system (FS).

Location The file must be stored on some device.

Size The file has a size which depends on how much data is in it, and how much is needed for storage.

Protection Access control information must also be available.

Time Date and time, for audit recording and other useful things.

File Meta-Data and Operations

Operations

Creating One must be able to create a new file, i.e. allocate space and entry for the file.

Writing We need some support from the OS for writing to a file, a system call.

Reading We also need a system call for reading from a file. This must keep track of where in the file we currently are, current-file-position pointer.

Repositioning We must be able to move within the file.

Delete And we must be able to remove the file if it's no longer needed.

File Meta-Data and Operations

- We need some other system calls and structures too.
- The OS uses an open(2) and a close(2) system call.
- OS keeps track of open files using the open-file table. Can be seen using lsof(1).
- The OS also has a per-process open-file table, this is where stdin and stdout resides.
- Keeps a file pointer, file-open count, disk location of the file.

File Meta-Data and Operations

```
\$ lsof .
COMMAND    PID   USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
bash      6729 danbos  cwd    DIR    8,5     4096 8667228 .
vim      13268 danbos  cwd    DIR    8,5     4096 8667228 .
bash     32564 danbos  cwd    DIR    8,5     4096 8667228 .
lsof     32565 danbos  cwd    DIR    8,5     4096 8667228 .
lsof     32566 danbos  cwd    DIR    8,5     4096 8667228 .
\$
```

Locks on Files

- What happens if two processes want to open the same file?
- We have shared and exclusive locks.
- OS also provides either mandatory or advisory file-locking mechanisms.
- Windows has mandatory (kind of), UNIX has advisory.

File Types and Internal Structure

- The file name is for the user. It can have an extension as a hint to the user and OS about its type.
- To see what type a file really is, try file(1).
- MacOS X keeps track of which program created the file, to use that for later opening again.
- Windows is entirely extension based.
- UNIX-like systems depends. It can use a magic number inside, a hint from the extension, or the file(1) utility.

File Types and Internal Structure

- From the operating system point of view, a file is simply a stream of bytes.
- The OS packs a certain number of logical records into a physical block on disk. Usually a logical record is a byte.
- Disk space is allocated in physical blocks on the device, usually somewhere between 512 to 4096 bytes.
- If a file is less than the block size we have internal fragmentation.

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

Mittuniversitetet
MID SWEDEN UNIVERSITY

Sequential Access

- What would you say was the biggest difference between the casette tape and the CD?
- The casette tape is a good example of sequential access, the CD is more close to direct access.
- You start reading from the beginning and can read forward or rewind.
- This simple model of access also works on random-access devices.

Direct Access

- In the direct-access model the file is made up of fixed-length logical records.
- These records can be accessed in no particular order.
- The system doesn't need to search for the record by winding, it can just compute the location and go there directly.

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

Mittuniversitetet
MID SWEDEN UNIVERSITY

Partitions and Volumes

- A storage device can be partitioned into several partitions.
- Then one can fit several file systems in one device.
- One can even leave space for swap space or raw disk.
- Partitions can also be called slices or minidisks.
- A file system can be created in any of these parts.
- A partition containing a file system is called a volume.
- The volume may be part of a disk, a whole disk or several disks in a RAID formation.
- Each volume has a device directory or volume table of contents to keep track of all files.

Partitions and Volumes

- There are many types of file systems!
- Memory-based file systems, various types of virtual file systems.

Directory Overview

- The directory can be seen as a symbol table translating names into directory entries.
- In a directory we can search for files, create and delete files, rename a file, traverse to subdirectories or parent directories.

Single-Level and Two-Level Directory

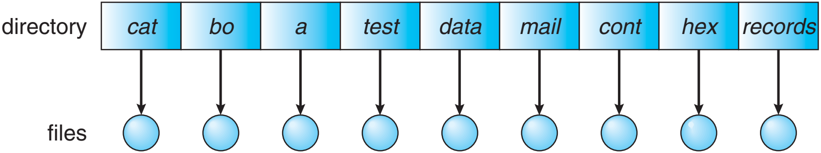

Figure: A one-level directory structure. Image: [SGG13b].

Single-Level and Two-Level Directory

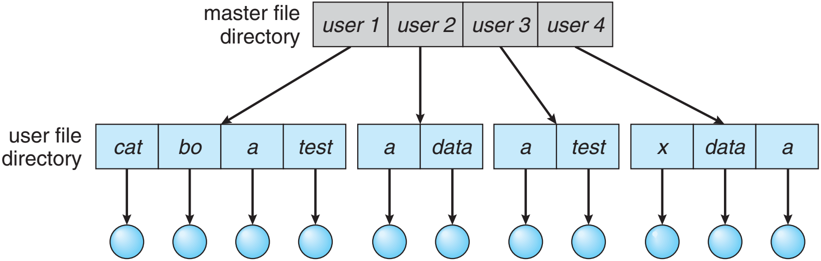

Figure: A two-level directory structure. Image: [SGG13b].

Tree-Structured Directories

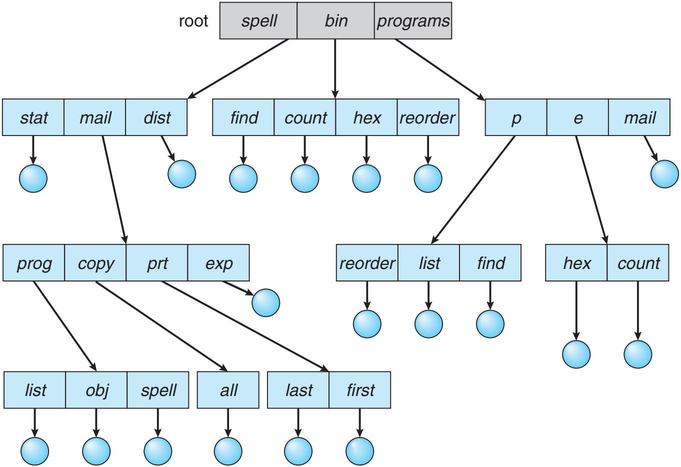

Figure: A tree-structured file system. Image: [SGG13b].

Graph Directories

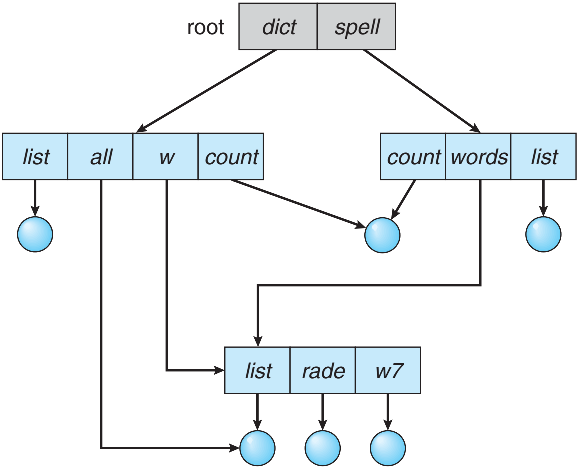

Figure: An acyclic graph directory structure. Image: [SGG13b].

Graph Directories

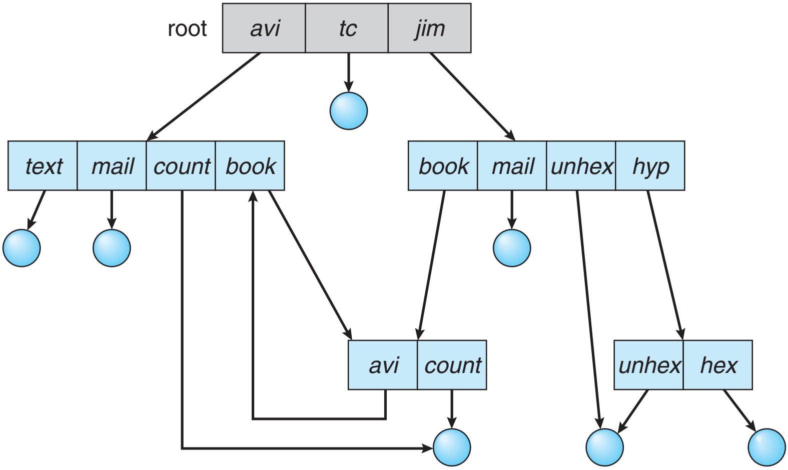

Figure: A general graph directory structure. Image: [SGG13b].

Graph Directories

- Directories can share subdirectories.
- Can be implemented using links.

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

Mount Points

- FS must be mounted, just as a file must be opened.
- An FS can be mounted in any directory using a special directory entry pointing to the root directory of the mounted file system.
- Windows uses an extended two-level directory structure with devices and volumes assigned drive letters. Then volumes have a general graph directory structure.

Mount Points

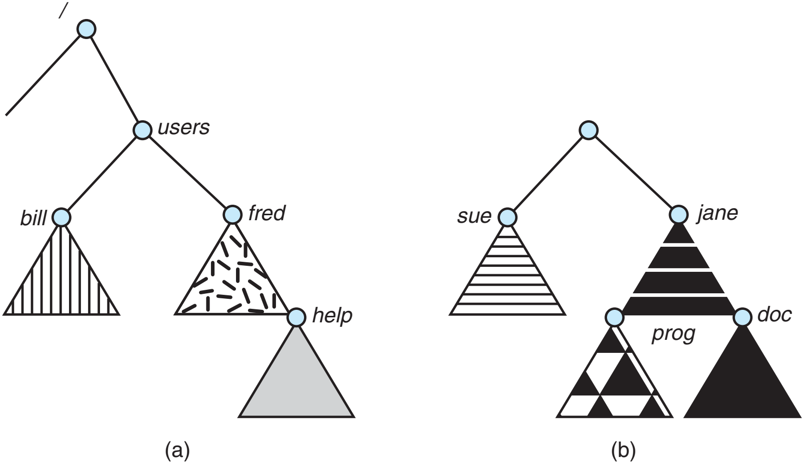

Figure: Two file systems, one is unmounted. Image: [SGG13b].

Mount Points

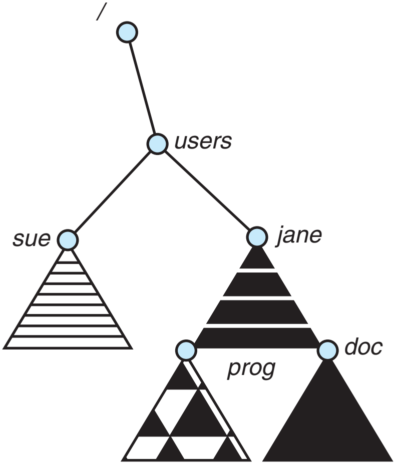

Figure: The previously unmounted file system is now mounted. Note that the subdirectories are invisible when the directory is used as a mount point. Image: [SGG13b].

Mount Points

- In UNIX-like systems the command mount(8) is used to work with mounting volumes.
- Without arguments mount(8) will show information on all currently mounted volumes in the system.

Remote File Systems

- These FSs are mounted as other FSs, although they need some special care in the kernel.

Overview

1. File Concept
   - File Meta-Data and Operations
   - Locks on Files
   - File Types and Internal Structure
2. Access Methods
   - Sequential Access
   - Direct Access
3. Directory and Disk structure
   - Partitions and Volumes
   - Directory Overview
   - Single-Level and Two-Level Directory
   - Tree-Structured Directories
   - Graph Directories
4. File System Mounting
   - Mount Points
   - Remote File Systems
5. Protection
   - Access Control

Access Control

```
\$ ls -lha
total 60K
drwxr-x--- 2 danbos danbos 4.0K Dec  3 13:16 .
drwxr-x--- 6 danbos danbos 4.0K Dec  3 10:07 ..
-rw-r----- 1 danbos danbos  11K Dec  3 13:16 fs.tex
-rw-r----- 1 danbos danbos  28K Dec  3 13:16 .fs.tex.swp
-rw-r----- 1 danbos danbos  100 Dec  3 10:12 literature.tex
-rw-r----- 1 danbos danbos  551 Dec  3 10:10 Makefile
\$
```

Referenser I

Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. 9th ed. International Student Version. Hoboken, N.J.: John Wiley & Sons Inc, 2013.

Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. 9th ed. Hoboken, N.J.: John Wiley & Sons Inc, 2013.