File System Topics

- File Concept
- Access Methods
- Directory Structure
- File System Mounting
- File Sharing
- Protection

File Concept

- A file is a named collection of related information recorded on secondary storage.
- "Contiguous" logical address space.
- File types:
 - Data:
 - numeric.
 - character.
 - binary.
 - Program (executable).

File Structure
None: just a sequence of words or bytes.
Simple record structure:
– Lines,
- Fixed length,
- variable length.
 Complex Structures. Formatted document.
 Relocatable load file.
Can simulate last two with first method by inserting
appropriate control characters.
Who decides:
 Operating system, Program

File Attributes

- Name only information kept in human-readable form.
- 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 is kept in the directory structure, which resides on the disk.

File Operations

- Create.
- Write.
- Read.
- Random access.
- Delete.
- Append.
- Truncate (reset size to 0, keep current attributes).
- 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 type	usual extension	function
kecutable	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
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 com- pressed, for archiving or storage
multimedia	mpeg, mov, rm,	binary file containing



Users and Passwords

 In Unix, globally readable file /etc/passwd maps each user's name to an integer number, to a home directory, and to a shell.

 /etc/shadow maps each user to a an encrypted (actually, *hashed*) password. Not readable to prevent "password guessing" attacks.





ACCESS CONTION (PROLECTION) Access control (RBAC): System policy: defines <u>"roles"</u> (generalization of the Unix idea of groups). Access can change roles (in a pre-defined set of possibilities) Applicy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	Access Control (Protection)	
Role-Based Access Control (RBAC): System policy: defines "roles" (generalization of the Unix idea of groups). Roles are associated with access rules to sets of files and devices. A process can change roles (in a pre-defined set of possibilities) during execution. A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	Access Control (Frotection)	
 Roles are associated with access rules to sets of files and devices. A process can change roles (in a pre-defined set of possibilities) during execution. A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	 Role-Based Access Control (RBAC): System policy: defines <u>"roles"</u> (generalization of the Unix idea of groups). 	
 A process can change roles (in a pre-defined set of possibilities) during execution. A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator 	 Roles are associated with access rules to sets of files and devices 	
A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	 A process can change roles (in a pre-defined set of possibilities) during execution. 	
A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the system. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator		
System. Users can be granted access to multiple roles. The assignment of users to roles can be changed more easily. The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	A policy determines information access rights for each role based on needs is determined in advance and is strictly enforced by the	
The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	system. Users can be granted access to multiple roles. The	
The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	assignment of users to roles can be changed more easily.	
The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator		
	The a change in the rights for each roles requires a change in the policy and actions carried out by an administrator	
	policy and actions carried out by all administration	

Open Files

Several pieces of data are needed to manage open files:

- Open-file table: tracks open files
- File pointer: pointer to last read/write location, per process that has the file open
- File-open count: counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
- Disk location of the file: cache of data access information
- Access rights: per-process access mode information

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
- Hediates access to a file
- 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

















File System Mounting

- A file system (partition) must be mounted before it can be accessed. Mounting allows one to attach the file system on one device to the file system on another device.
- A unmounted file system needs to be attached to a mount point before it can be accessed.



Operations on Directories	
Search for a file	
• Create a file	
Delete a file	
List a directory.	
Rename a file.	
Traverse the file system.	
······································	
Goals of Directory Logical	
Organization	
 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, ...)

















Tree-Structured Directories
(Cont.)
Absolute or relative path name.
Creating a new file is done in current directory by default.Delete a file
<pre>rm <file-name></file-name></pre>
Creating a new subdirectory is done in current directory.
<pre>mkdir <dir-name></dir-name></pre>
Example: if in current directory /mail
mkdir count
mail
prog copy prt exp count
rm -rf . \Rightarrow doesn't mean "read mail really fast"



Acyclic-Graph Directories (Cont.)	
• Different names (<u>aliasing</u>) for the same file or	
directory.	
 Must be careful with removals to avoid dangling pointer 	
Solutiona	
Solutions:	
 Backpointers, so we can delete all pointers. Variable size records a problem. 	
– Backpointers using a daisy chain organization.	
- Entry-hold-count solution.	
, ,	





General Graph Directory (Cont.)	
 How do we guarantee no cycles? Allow only links to file not subdirectories. Garbage collection. Every time a new link is added use a cycle detection algorithm to determine whether it is 	
OK.	

File Sharing

- Sharing of files on multi-user systems is desirable.
- 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-System Structure

- File structure:
 - Logical storage unit,
 - Collection of related information.
- File system resides on secondary storage (disks).
- File system is organized into layers.
- File control block storage structure consisting of information about a file.







Virtual File Systems
• Virtual File Systems (VFS) provide an object- oriented way of implementing file systems.
 VFS allows the same system call interface (the API) to be used for different types of file systems.
The API is to the VFS interface, rather than any specific type of file system.



Directory Implementation

The directory is a <u>symbol table</u> that maps file names to pointers that lead to the blocks comprising a file.

- Linear list of file names with pointer to the data blocks:
 - simple to program, but...
 - time-consuming to execute.
- Hash Table:
 - decreases directory search time,
 - *collisions* situations where two file names hash to the same location,
 - fixed size.



Contiguous Allocation	
	 Each file occupies a set of contiguous blocks on the disk.
	 Simple: only starting location (block #) and length (number of blocks) are required.
	• Suitable for sequential and random access.
	 Wasteful of space: dynamic storage-allocation problem; external fragmentation.
	 Files cannot grow unless more space than necessary is allocated when file is created (clearly this strategy can lead to internal fragmentation).















Free-Space Management

- Bit vector (I bit per disk block)
- Linked list (free list)
- Grouping
 - like linked list: first free block has n block addresses (the n-l addresses are free blocks, the nth is the address of a block with the next bundle of addresses)
- Counting
 - like linked list, but each node points to a cluster of contiguous, free blocks

The OS can cache in memory the free-space management structures for increased performance. Depending on disk size, this may not be easy.

Bit Vector
Bit Vector (or Bit Map)
101010111111010011111000001111
$ \begin{array}{c} 0 \\ 1 \\ 10 \\ 11 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 15 \\ 16 \\ 17 \\ 18 \\ 19 \\ 10 \\ 11 \\ 18 \\ 19 \\ 19 \\ 19 \\ 19 \\ 10 \\ 11 \\ 11 \\ 11$









Counting
first free block
$\begin{array}{c c c c c c c c c c c c c c c c c c c $

	Counting
first free block	
1, 5	
	$\begin{array}{c}3\\ \hline \end{array} \begin{array}{c}4\\ \hline \end{array} \begin{array}{c}5\\ \hline \end{array} \begin{array}{c}6\\ \hline \end{array} \begin{array}{c}7\\ \hline \end{array} \begin{array}{c}8\\ \hline \end{array} \begin{array}{c}9\\ \hline \end{array} \end{array}$
	23 24 25 26 27 28 29

Recovery

- Consistency checking compares data in directory structure with data blocks on disk, and tries to fix inconsistencies.
- Use system programs to back up data from disk to another storage device (floppy disk, magnetic tape).
- Recover lost file or disk by *restoring* data from backup.

Log Structured File Systems	
	 Log structured (or journaling) file systems record each update to the file system as a transaction.
	• All transactions are written to a log. A transaction is considered committed once it is written to the log. However, the file system may not yet be updated.
	• The transactions in the log are asynchronously written to the file system. When the file system is modified, the transaction is removed from the log.
	 If the file system crashes, all remaining transactions in the log must still be performed.