

Announcements

- Reading Chapter 13
- Project #5 handout available
- Midterm #2 will be returned Today

Log Structured File Systems

- Key Idea
 - Use transactions like model for filesystem updates
- Write data to a log (also called a journal)
 - Records meta data changes
 - Records data blocks written
 - File operation is committed once it is to the log
 - Partial updates to log are lost on failure
- Next Step
 - Eliminate the filesystem and just keep the log
 - Requires a process called a cleaner
 - Copies old data from log to head of log to allow compaction

NTFS

- File system may
 - Be a partition (fraction) of a disk
 - May span multiple disks
- Clusters
 - Group sectors into a larger group (typically 4KB)
 - Logical cluster numbers (0...N) describe where a cluster is
- File consists of a set of attributes
 - Attributes
 - arbitrary sized
 - Linear ordering from 0...n
 - Examples
 - Filename
 - File data
 - Security
 - Mac Resource fork

NTFS Files

- Each file is stored in an entry in the Master File Table (MFT)
 - Each entry 1-4KB
 - Small attributes stored directly in MFT
 - Larger attributes are stored in one or more extents (contiguous clusters on the disk)
- Special Files
 - MFT – file 0
 - Copy of first 16 entries in MFT
 - Log file – log of changes to file system
 - Attribution definition table
 - Root directory
 - Bitmap free list
 - Boot file (must be at a standard disk address)
 - Bad cluster file

NTFS Files

- Each file is stored in an entry in the Master File Table (MFT)
 - Each entry 1-4KB
 - Small attributes stored directly in MFT
 - Larger attributes are stored in one or more extents (contiguous clusters on the disk)
- Special Files
 - MFT – file 0
 - Copy of first 16 entries in MFT
 - Log file – log of changes to file system
 - Attribution definition table
 - Root directory
 - Bitmap free list
 - Boot file (must be at a standard disk address)
 - Bad cluster file

NTFS (cont)

- Each Directory is stored in a B+ Tree
 - Permits fast lookup of information
 - Index root is a cluster with the top of the B+ tree
 - Larger directories contain additional clusters
- Security
 - Each file has an access token (owner info) and an ACL list
 - Permissions not normally checked on directories

NTFS

- Logs
 - Stores redo and undo info for changes to FS state
 - Only represents data structures of the file system not data
 - Writes a commit record when the update is done
- Multi-partition files
 - Volume set
 - Concat up to 32 partitions into one larger file system
 - Stripe set (basically RAID0)
 - Round robin among partitions on a per LCN basis
 - Stripe set with parity (RAID5)
 - Disk mirroring (RAID1)

NTFS Plugin Features

- **Change Journal**
 - Way for user space processes to learn what files have changed
 - Useful content index services
 - Used by replication services
- **Volume Shadow Copy**
 - Provides copy on write of files after the shadow copy
 - Permits
 - Backups in a consistent state
 - User file undo operations

Extent Based Storage

- Try to keep blocks together
- Allocate blocks prior to use
 - Reserves space for file (user can specify size)
 - Ensures clusters are together

UNIX Shell and Current Directory

- **Current Directory**
 - Maintained on a per process basis by kernel
 - System Calls: get/set the current directory
 - Open system Call
 - File name checked and if it lacks a leading /, pre-pend cwd onto path
- **Shell (file path)**
 - Entirely implemented in user space
 - PATH Environment variable
 - Lists directories to search
 - Hash table of commands and their location (file, or internal)