Persistent Storage

- Datastructures and Algorithms
L 07: Case Study: Unix FS
Questions: Encoding

- What is an encoding?
  - Name some examples of codes
  - Which are used in computers?
Questions: Encoding

- Differences between Morse and Baudot?
- What is ASCII?
  - Is there more than printable characters?
- What is EBCDIC?
  - Difference to ASCII?
- What is more in ISO-8859?
- The goal of Unicode?
  - What is different to previous codes?
  - UTF-8 / UTF-16
Questions: Consistency

• What is ACID?
• Name 3 consistency methods / classes!
  – What is necessary to make it work?
  – Storing a new set of informations, how?
  – How does the recovery work after reboot?
Questions: FAT

- The 4 parts of a FAT filesystem?
- Which variants of FAT do you know?
  - What does the number indicate?
- What is in a directory entry?
- Where is the free space?
- What is to do to...
  - read from a file
  - create a file and write into it
Filesystem - what is that?
Filesystem - what is that?

- Management of smaller objects on a disk?
  - Since some time also multiple disks
- Listing of the objects
  - Names, size, attributes
- Management of the disks space (or other media)
- POSIX (Unix Standard)
  - Requires multiple names for a file
A Unix Filesystem

- Management of Objects
  - Directory
    - Path to the file, f.e.: /home/ulrich/funny.mp3
    - In Unix: hierarchical, with subdirectories
  - Rights
    - user(s), group(s)
    - read, write, execute, delete, ...
  - Attributes
    - length, extended attributes
  - Space allocation
UFS predecessor: Unix Filesystem

- Unix System 6 Filesystem
  - Bootblock
  - Inodes
  - Directories
  - Data blocks

- Used in
  - Unix System 6 and System 7
  - Available in BSD, Linux, Solaris until S9
Unix Filesystem

- Bootblock (superblock):
  - Magic number
  - Position and size of the other parts
  - This is an F Format (using the full sector)
  - Superblock in some copies for recovery
Unix Filesystem

• Directory
  – Table of files
  – Contains records with 2 elements
    • Filename (14 Byte)
    • inode-# (2 Byte = only 65536 possible...)
    • ... this is FB16 record format ...
    • Nothing else (! different to FAT filesystem)
      – All attributes are stored in the inode
  – One inode entry can have 2 directory entries
    • => One file can have 2 names
Unix Filesystem

- Directory
  - The root directory has a defined inode-
  - Other special entries:
    - "lost+found" special purpose directory
    - For file system check (command: fsck)
      - Stores anything surplus into "lost+found"
        - blocks belonging to two files
        - blocks not in file or free list
        - files not in directories
        - ...
Unix Filesystem

• Inodes
  - Indexed by a number (inode-#)
    • see ls -li
  - Represents a file
  - Rights, Attributes, Allocation
  - Symlinks
    • Some implementations: use allocation space
  - NOT: The name of the file!
  - ... also FB256 or FB512 format ...
Unix Filesystem

- Root block (super block)
- Freespace (bitmap)
- Inodes (fixed amount)
  - Has to be defined at creation
- Data blocks
  - (Sub-) directories
  - Files
  - Perhaps: symlinks
Unix Filesystem: Read from a file

- Read the super block
- Find the root directory (fixed inode)
  - Read content of root directory (like file below)
  - Get inode-# of path element
  - Recursive descent until directory of file is found
  - Get inode-# of file
- Read file through its inode entry
  - Read allocation information
  - Compute block-# and read it
Unix Filesystem: Write to a file

- Find inode-# of file (like in reading)
- Compute the block-# from the allocation info
- Write to the correct block

- Extending a file (or sequentially creating a file)
  - Take block from free space (bitmap)
  - Add it to allocation (perhaps multiple changes)
  - Then write to the block
Unix Filesystem: Create Objects

- Create a file
  - Allocate an inode (perhaps: bitmap)
  - Write to the corresponding directory as it would be a file

- Create a directory
  - Like file creation, but use other attributes
  - Initial directory entries: "." and ".."
Unix Filesystem: Delete

- Delete objects
  - Delete the directory entry
  - Delete the inode, if not referenced
    - Active references count!
      (a file stays as long it is used)
- Delete the inode then frees the blocks below
Unix Filesystem: Problems

- Problems (see also: FAT)
  - New files are probably placed at the end
  - Metadata placed at a fixed location
    - At the beginning of the device
    - Fixed number of inodes
    - Creation and writing of a file need a lot of large seeks
      (destroys elevator algorithm)
  - Changes in the filesystem create some seeks
Unix File System (UFS)

- Developed 198x under the name (Berkeley) fast file system (FFS)
  - Marshall K. McKusick, Bill Joy, …
- Basic idea: System V FS + cylinder groups
  - Locality of Metadata
  - More Performance
- Later additions:
  - Size up to 16 TiB
  - Log
  - 64 Bit allocation (BSD)
UFS Usage

• Used in
  - BSD*
  - SunOS (since 1983), Solaris (since 1992)
  - A/UX (Apple Unix previous to MacOS X)
  - HP/UX
  - True 64 (Digital, now HP)
  - Windows NT filesystem is similar
  - Linux (as a kernel module)
UFS Resources

- Overview: en.wikipedia.org: UFS
- Details: Solaris Internals, Chapter 15
  - Can be found ...