File System Implementation: issues

Unifying File Systems, Networks and Virtual Memory

Lecture 15

Reading: Bacon 7.6, 7.7

Free Space Management

How to keep track of free space?

- Keep a bit vector (0 or 1; 1 for a free disk block)
  - simple to find first free block (bits_per_word) x (0_words) + (offset_first_1_bit)
  - 4GB disk with 512bytes block needs 1MB space
- Linked list of free blocks
  - Need I/O to locate *n* free blocks
  - Alternatively: keep info in the FAT
- Grouping
  - for every *n* free blocks record info in *1* about the *n-1*

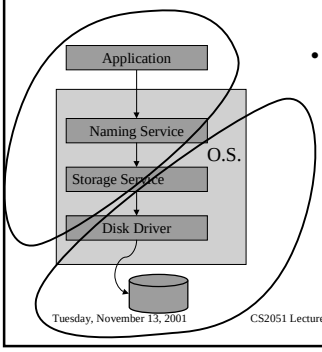
Other issues...

- Efficiency
  - dependent on allocation, directory algorithms, data, ...
- Performance
  - keep some disk blocks in memory
    - replacement policy: LRU not good! free-behind, read-ahead
  - allocate space in clusters of blocks
- Recovery
  - consistency checking (eg crash with directories in memory)
  - keep backups!

Practical Considerations

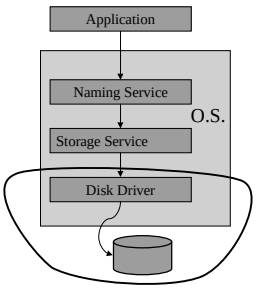
- How to serve concurrent requests to the disk?
  - access time depends on transfer rate + positioning time(seek&rotate)
- Disk formatting
  - low-level (sector structure)
  - OS needs to record its own data structures on disk
- Partitions and mounting
  - different file systems may be available
- Bootstrapping
  - stored at a fixed location on disk (called from ROM).

Network Based File Servers



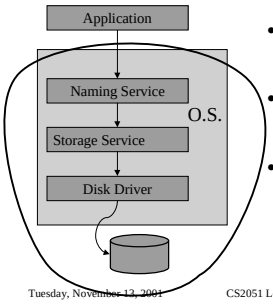
- Problem:
  - split functionality between *client* and *server*

Solution 1: Network Disk



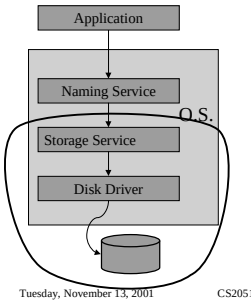
- Server provides just physical storage
- Problems:
  - coordinating multiple client systems?
  - freespace management?

Solution 2: File System Export



- All names resolved on server
- "name" means same file to every client
- Not ideal if clients are different (linux/ winNT/Sun/etc.)

Solution 3: Sun NFS



- Each "name" component of filename is translated on client
- Storage service implemented on server

Issues in Network File Systems

- Administration
- Protection:
  - Trust, Validation
  - different administration domains
- State
  - Client "knows about" server
  - Server "knows about" client?
  - Server crash? Multiple servers?

Virtual Memory and Storage

- VM:
  - copies parts of memory image to disk
  - copies areas from disk to memory
- File Manager:
  - copies from memory to disk
  - copies areas from disk to memory

???

Unified VM and File Manager

(memory-mapped files)

File Operation	Equivalent VM Operation
open	map file into Virtual Address Space
read	access virtual address: page fault causes disk read
write	access virtual address: (eventual) page rejection causes disk write [n.b. zero fill on write to unallocated page?]
close	unmap page

- Ideal for sharing:
  - useful for programs and libraries