Announcements

- Project #6 is available

Project #6 Notes

- Uid
  - – First process has uid of 0
  - – Spawned processes
    - Inherit uid of parent
    - Unless setuid bit is set on program to run, then the uid of the owner of that file is used
- ACLs
  - – First ACL entry is owner
  - – Others are for other users
    - Can delete these entires with setACl(file, uid, 0)
  - – Uid 0 can open any file regardless of ACLs

Distributed Systems

- Provide:
  - access to remote resources
  - security
  - location independence
  - load balancing
- Basic Services:
  - remote login (telnet and rlogin protocols)
    - extends basic access provided by normal login
  - file transfer (ftp, rcp)
    - can support anonymous transfers
  - information services (http)
    - two way protocols (request/response)

Distributed Systems

- A unified view of local and remote access
- Typical Services
  - data migration
    - provide only the data required, not the whole file
    - manage multiple copies as versions of the same object
  - process migration
    - a process can move from one machine to another
    - reasons for migration:
      - load balancing
      - data affinity
      - hardware/software preference (better configuration)

Distributed OS Design Issues

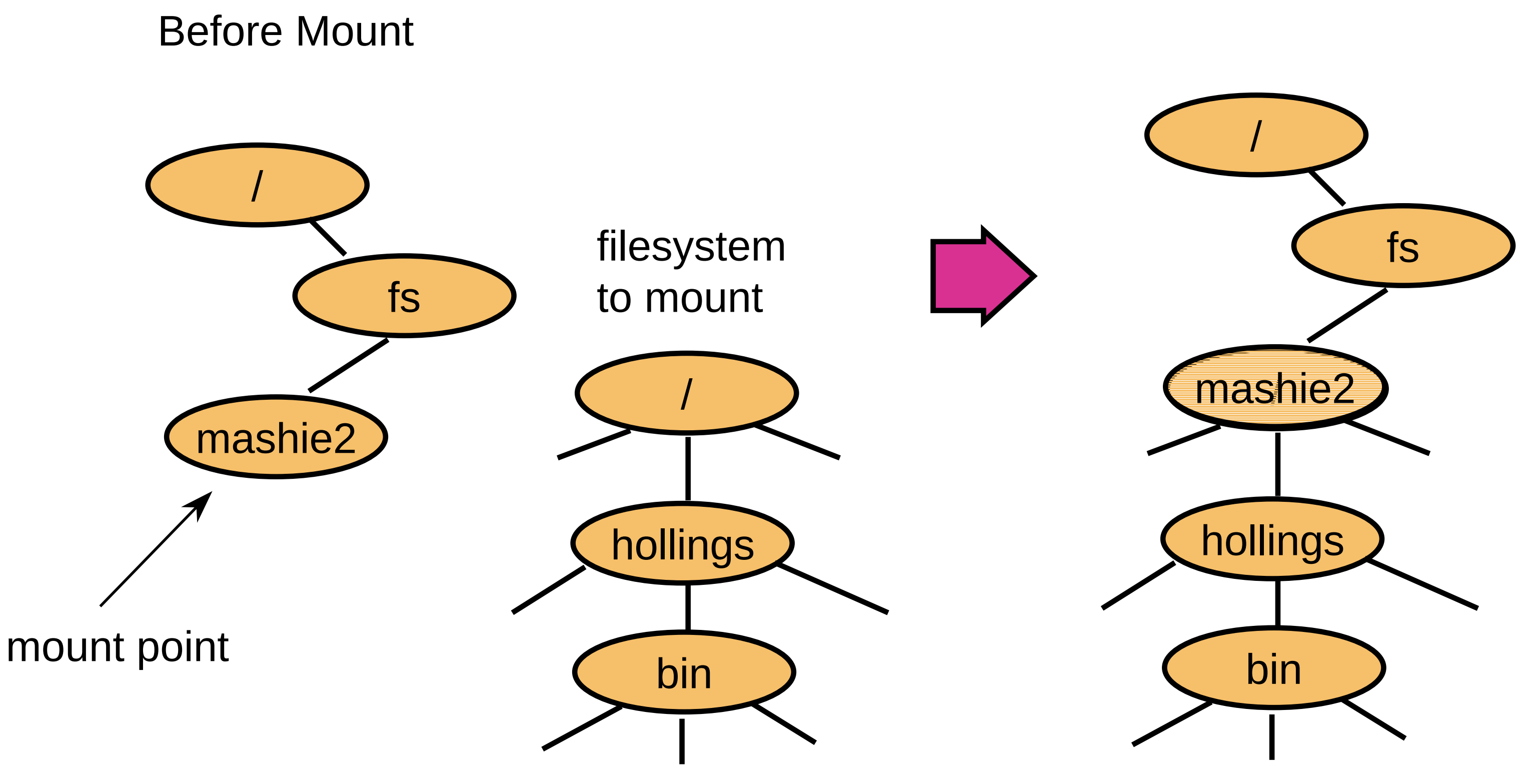
- Should provide same model as a central system
  - easy to understand for users
- Needs to be scaleable
  - will it work with 100, 1,000, or 10,000 nodes?
- Failure Modes
  - avoid a single central failure point
  - can loss performance or functionality with failure
    - but loss should be proportional to size of failure
- Security
  - should provide same guarantees on data integrity as a local system

File Server State

- Does the fileserver maintain information between requests?
- Stateless
  - example: NFS
  - each request contains a request to read/write a specific part of a file
  - requests must be *itempotent*
    - the same request can be applied several times
  - makes recovery of failed clients/servers easier
- Stateful
  - example: AFS
  - servers maintain connections for clients
  - improves performance
  - required for server based cache management

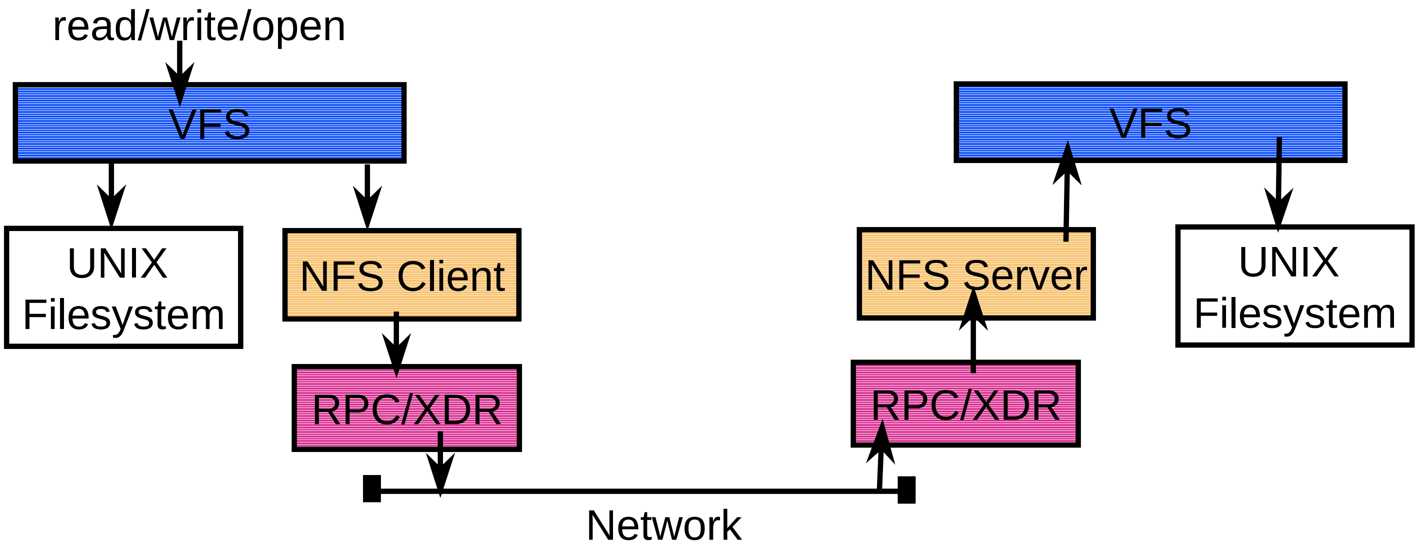
Mounting a filesystem

- Mount attaches a filesystem to a directory
  - can be used for local or remote (NFS) filesystems



NFS

- Provides a way to mount remote filesystems
  - – can be done explicitly
  - – can be done automatically (called an automounter)
  - – clients are provided “file handle” by the server for future use
- Uses VFS: extended UNIX filesystem
  - – inodes are replaced by vnodes
    - network wide unique inodes
    - can refer to local or remote files



NFS (cont.)

- Requests
  - are sent via RPC to the server
  - include read/write
  - query: lookup this directory info
    - must be done one step (directory) at a time
  - change meta data: file permissions, etc.
- Popular due to free implementations
- Provides no coherency

AFS

- Designed to scale to 5,000 or more workstations
- Location independent naming
  - within a single cell
- volumes
  - basic unit of management
  - can vary in size
  - can be migrated among servers
- names are mapped to “fids”
  - 96 bit unique id’s for a file
  - three parts: volume, vnode, and uniqidentifier
  - location information is stored in a volume to location DB
    - replicated on every server

AFS (cont.)

- File Access
  - open: file is transferred from server to client
    - very large files may only be partially transferred
  - read/write: performed on the client
  - close: file (if dirty) is written back to server
    - can fail if the disk is full
- Consistency
  - clients have callbacks
  - sever informs client when another client writes data
  - only applies to open operation
  - only requires communication when:
    - more than one client wants to write
    - one client wants to write and others to read

Display and Window Management

- The screen is a resource in a workstation system
  - multiple processes desire to access the device and control it
  - OS needs to provide abstractions to permit the interaction
- Services
  - protection
  - windows
  - multiplex keyboard and mouse
  - configuration and placement
- Issues
  - how to get good performance and remain device independent
  - how much policy to dictate to users