

Announcements

- Reading Chapter 11

File Protection

- How to give access to some users and not others?
- Access types:
 - read, write, execute, append, delete, list
 - rename: often based on protection of directory
 - copy: usually the same as read
- Degree of control
 - access lists
 - list for each user for each file the permitted operations
 - groups
 - enumerate users in a list called a group
 - provide same protection to all members of the group
 - depending on system:
 - files may be in one or many groups
 - users may be in one or many groups
 - per file passwords (tedious and a security problem)

File Protection Example (UNIX)

- each file has three classifications
 - user: the user who owns the file
 - group: a named group of other users
 - world: all others
- each file has three access types:
 - read, write, execute
- directory protection
 - read: list the files in the sub dir
 - write: delete or create a file
 - execute: see the attributes of the files in the subdir
 - sticky bit: can only modify directory entries owned by yourself

Unix File Protection (cont)

- Files have 12 bits of protection
 - 9 bits are user, group, and world for:
 - read: list the files in the sub dir
 - write: delete or create a file
 - execute: see the attributes of the files in the subdir
 - sticky bit: leave executable in memory after is done
 - setuid: run the program with the uid of the file's owner
 - used to provide extra privilege to some processes
 - example: passwd command
 - setgid: run the program with the group id of the file's owner

UNIX File Protection Example

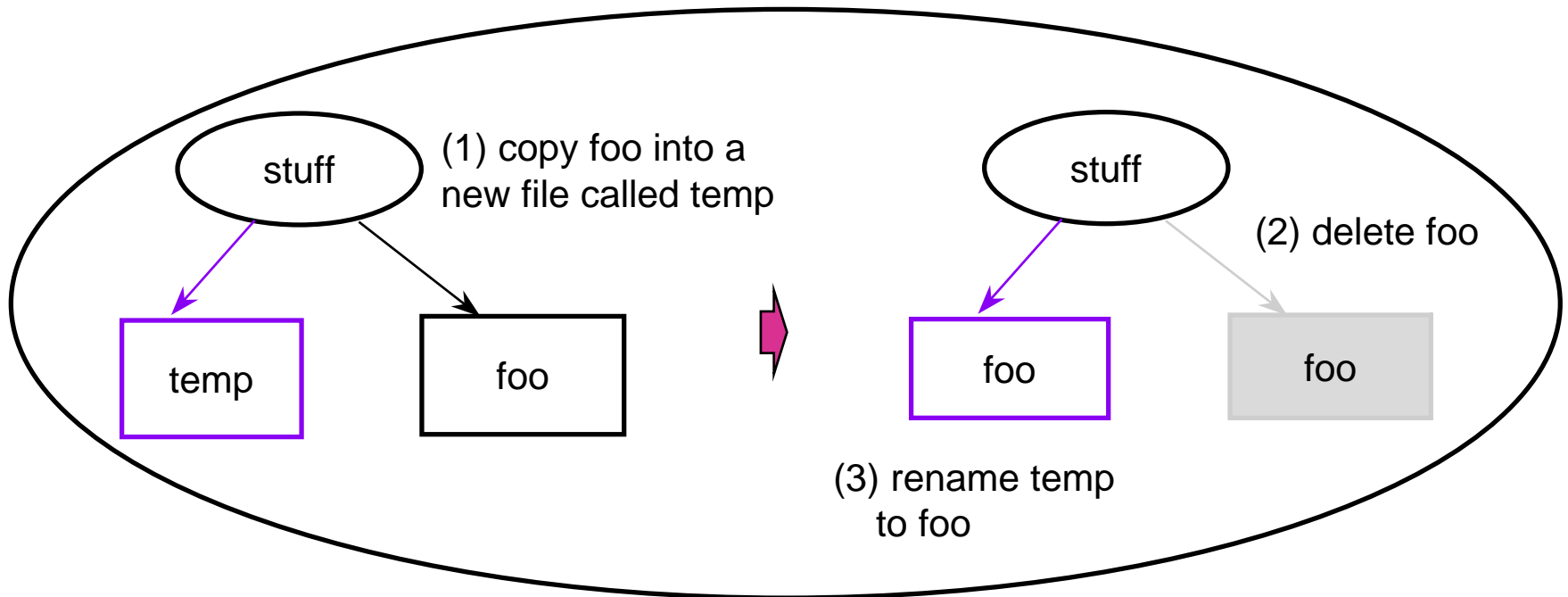
Stuff is a directory:
user hollings has r/w/x on the dir



foo is a file:
user hollings has r, but
not write on this file



hollings can still write the file!



File Protection Example (AFS)

- Each Directory has an ACL
 - protection information applies to all files in a directory
 - file access types are:
 - read, write, lookup, delete, insert, lock (k), administer
 - an ACL may be for a user or a group
 - ACL may contain negative rights
 - everyone but Joe Smith may read this file
- Groups
 - are collections of users
 - each user can create up to a fixed number of groups
 - users can administrate their own groups
- Cells
 - collections of computers (e.g., csic, wam)

File Consistency semantics

- How do multiple processes see updates to files
- UNIX
 - writes are visible immediately
 - have a mode to permit processes to share file pointers
- AFS
 - open/close semantics
 - “copy” the file on open
 - write-back on close
- Immutable files
 - once made visible to the world, the file never changes
 - usually done by attaching a version # to the filename
 - new versions of the file must be given a new name

Filesystems

- **Raw Disks can be viewed as:**
 - a linear array of fixed sized units of allocation, called blocks
 - assume that blocks are error free (for now)
 - typical block size is 512 to 4096 bytes
 - can update a block in place, but must write the entire block
 - can access any block in any desired order
 - blocks must be read as a unit
 - for performance reasons may care about “near” vs. “far” blocks (but that is covered in a future lecture)
- **A Filesystem:**
 - provides a hierarchical namespace via directories
 - permits files of variable size to be stored
 - provides disk protection by restricting access to files based on permissions

File System Implementation

Application Programs



Logical file system:
Knows about directories, application view of file names



File Organization Module:
Can translate logical block addresses to physical block addresses



Basic File System:
Issues physical block read/write commands



Low Level I/O Control
Interfaces to hardware