Announcements

- Reading Chapter 11 (11.1-11.5)
- Programming Project #3
  - is due Monday April 1
  - needs to include a paragraph write-up about the results of using the two different scheduling algorithms
- Midterm #2
  - it’s in two weeks (4/11/96)
  - don’t forget to study synchronization

Does the OS know what is stored in a file?

- needs to know about some types of files
  - directories
  - executables
- should other file types be visible to the OS?
  - Example: word processing file vs. spreadsheet
  - Advantages:
    - OS knows what application to run
    - Automatic make (tops-20)
      - if source changed, re-compile before running
  - Problems:
    - to add new type, need to extend OS
    - OS vs. application features are blurred
    - what if a file is several types
      - consider a compressed postscript file

Example of File Types

- Macintosh
  - has a file type that is part of file meta-data
  - also has an application associated with each file type
- Windows 95
  - has a file type in the extension of the file name
  - has a table to map extensions to applications
- Unix
  - can use last part of filename like an extension
  - applications can decide what (if anything) to do with it

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)

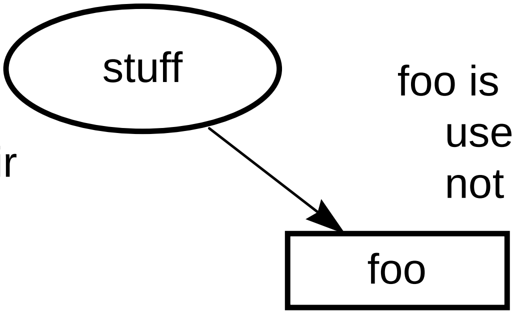
- each file has three classifications
  - user: the user who owns the file
  - group: a named group of other users
  - world: all others
- each file has three access types:
  - read, write, execute
- directory protection
  - read: list the files in the sub dir
  - write: delete or create a file
  - execute: 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

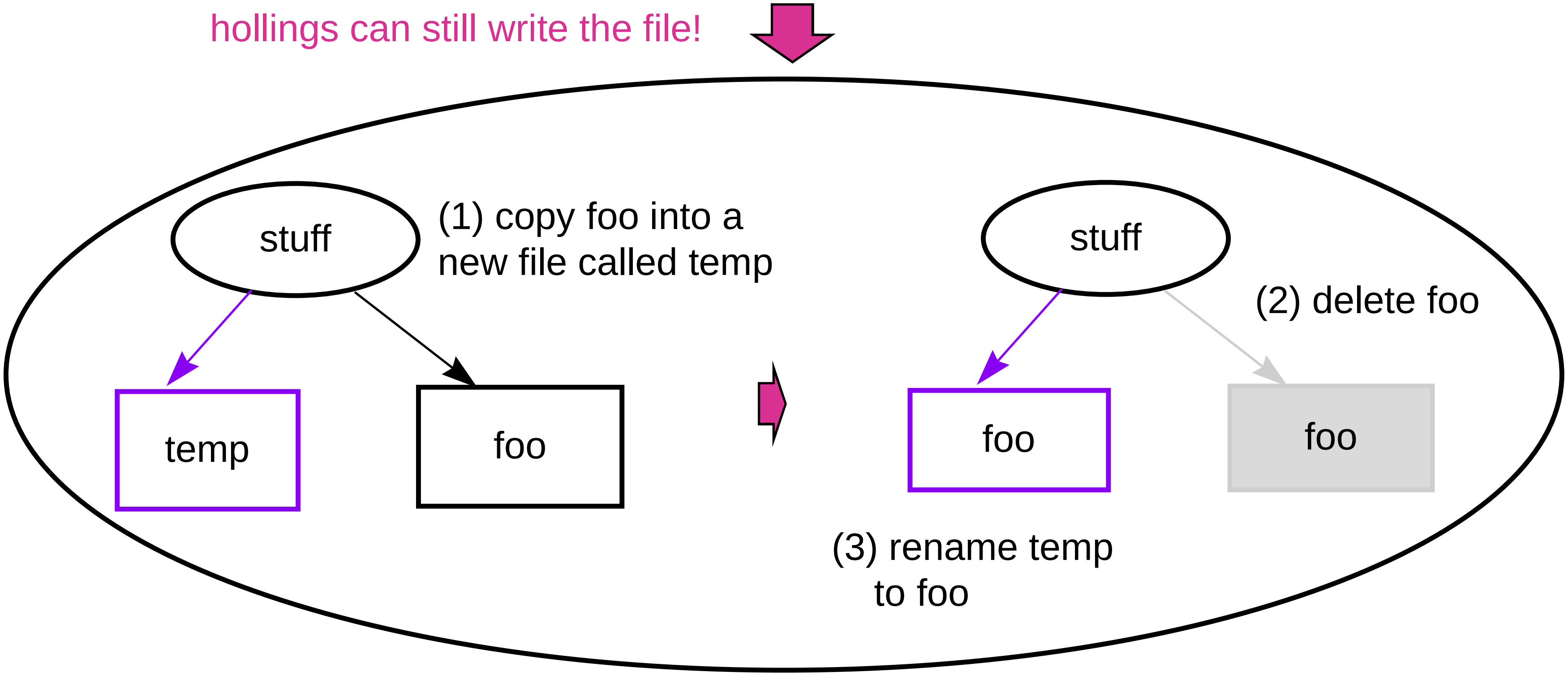
UNIX File Protection Example

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!



(1) copy foo into a
new file called temp

(2) delete foo

(3) rename temp
to foo

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. wam)

File Consistency semantics

- How to multiple processes see updates to files
- UNIX
  - writes and 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