Announcements

- Reading Chapter 11 (8th ed)
- Project #4 Due on Sunday at 6:00 PM
- Midterm #2 is next Tuesday

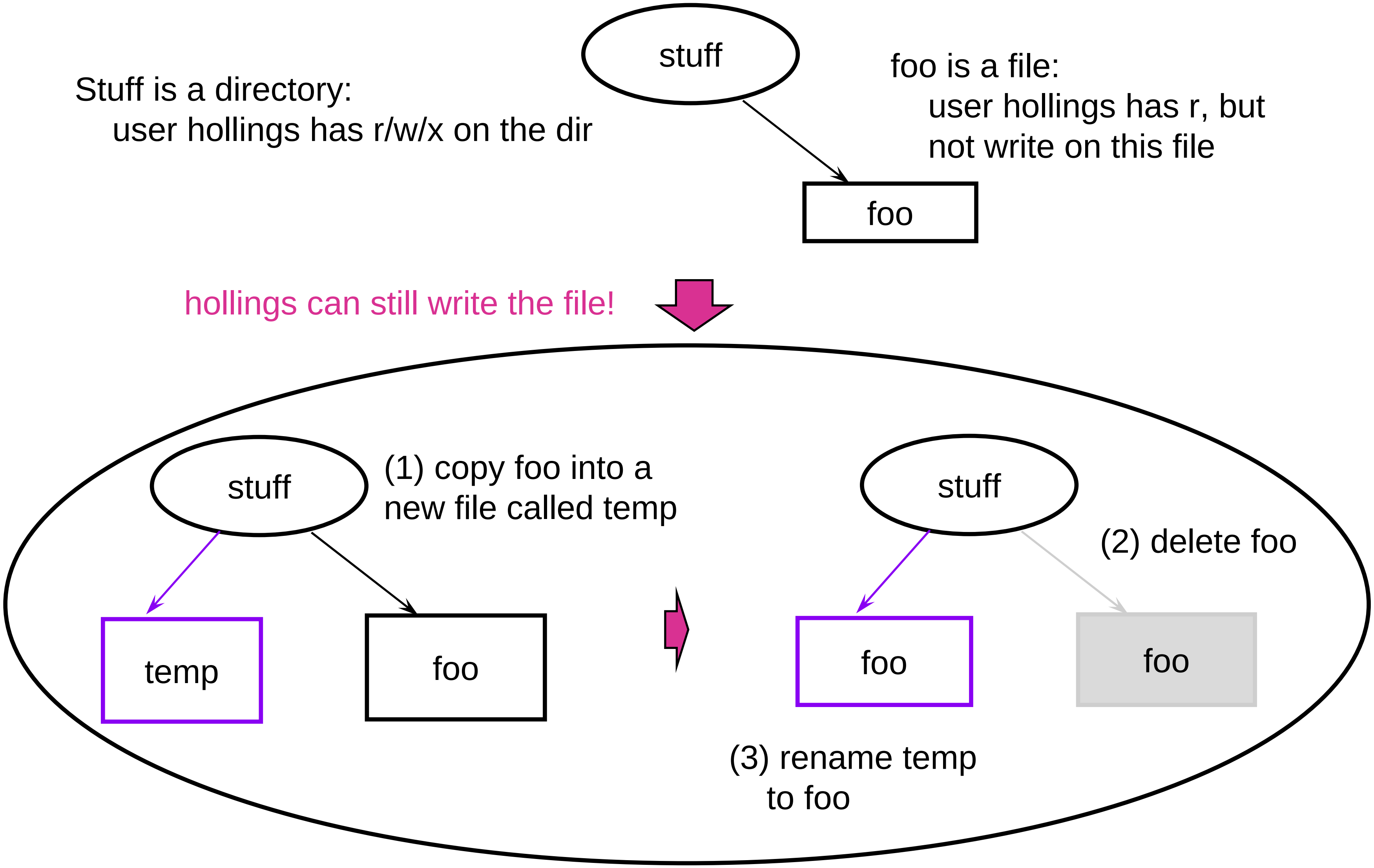
File Protection Example (UNIX)

- 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., 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