Lecture 6: File Systems

Last Day:
- Space Management
- File Directories

Today: File Directory Structures
- Tree structures
- Generalized structures/shared files
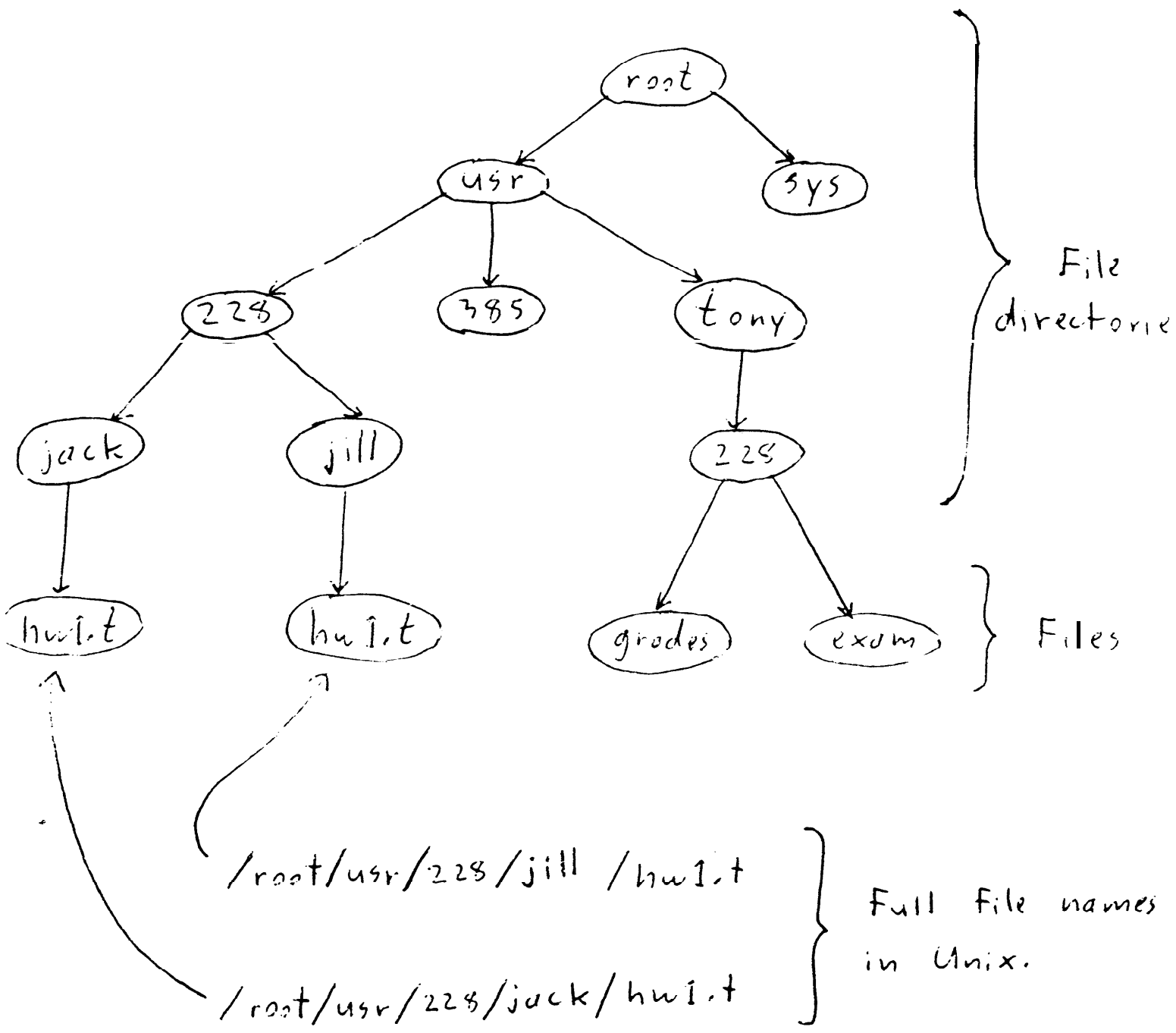- Reference Counters
- Garbage Collection

See handout on File systems.

& Folk & Zoelick § 2.7

File Directory Structures

- A single file directory does not allow different users to have files of the same name (eg. temp.c, hw1.t).

- ∴ Each user is given a separate file directory

- <u>Tree-Structured Directories</u>:

    • A file directory may have subdirectories which may have sub-subdirectories, etc.
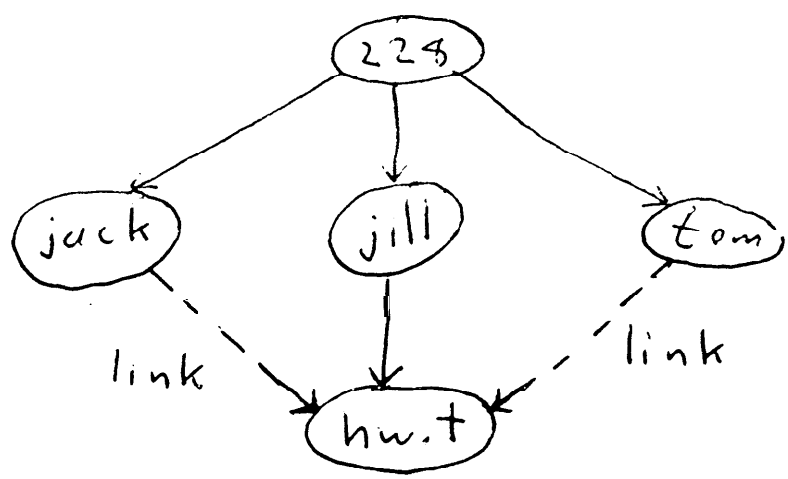
Example

File directories

Files

/root/usr/228/jill /hw1.t

/root/usr/228/jack/hw1.t

Full File names in Unix.

Generalized Directory Structures

- A tree-structured directory does not allow users to <u>share</u> files

- ie, A file can be in only one users directory at a time,

- <u>Links</u> allow users to share files.

eg.

It appears to jack that <u>hw.t</u> is a file in <u>his</u> directory. Likewise for Tom.

– In UNIX, the command

$$\text{ln} \quad /228/jill/hw.t$$

creates a (hard) link to the file <u>hw.t</u> in the directory /228/jill/
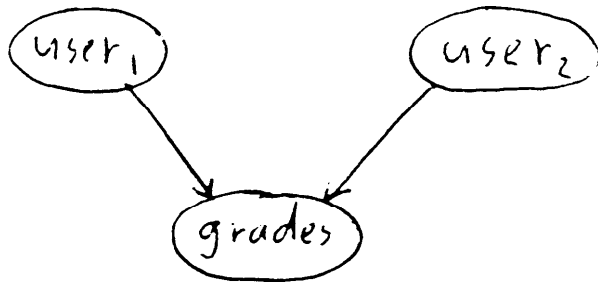
– To see the link, type "ls -l" in UNIX.

– <u>Note:</u> This is different from storing separate copies of one file.

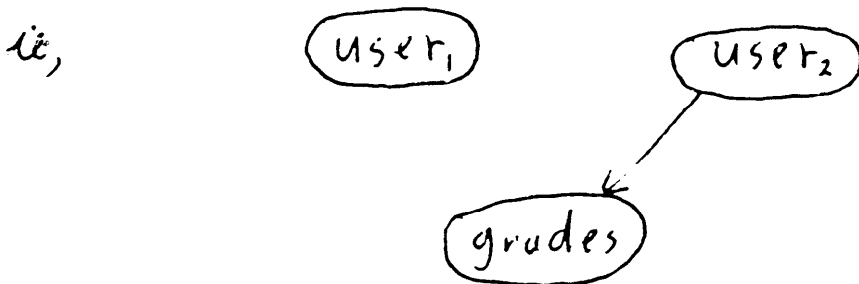- Copied files require extra disk space.

- When one copy is changed, other copies are not changed.

- In contrast, a shared file is stored only once, and all users linked to the file see all changes to it.

Deleting Shared Files

- If user$_1$ deletes "grades", does user$_2$ still see the file, or not?

- In most systems, "delete file" means delete the link to the file.

i.e,

This is the result of user$_1$ deleting grades.

- When the <u>last</u> link to a file is deleted, the system deletes the file too.

- <u>Problem 1:</u> How to determine when there are no more links to a file.
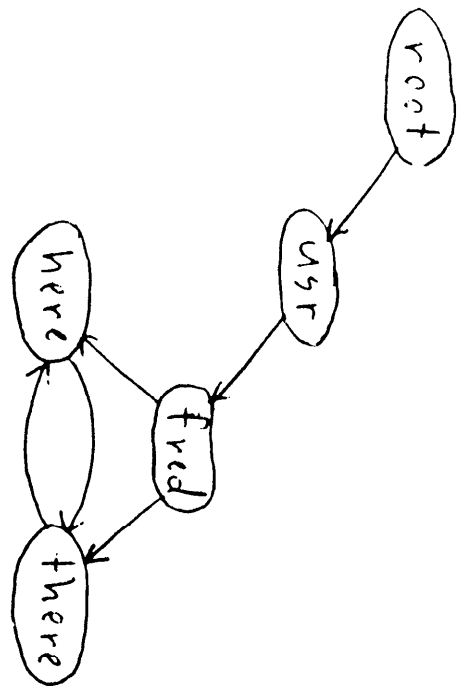
- <u>Solution 1:</u>   Reference Counters.

For each file, maintain a counter of the number of references (links) to the file.
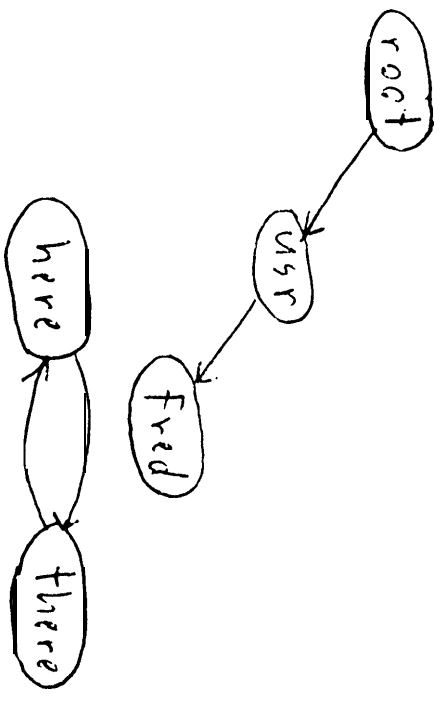
- When a link to the file is created, increment the counter.

- When a link is removed, decrement the counter.

- When the counter is zero, delete the file.

Problem 2: Cycles

 by

- Suppose Fred deletes <u>here</u> and <u>there</u>:

root → usr → fred → here ⇄ there

root → usr → fred → here ⇄ there (fred → there, here ⇄ there)

- <u>here</u> and <u>there</u> both have a reference count of 1 but they cannot be accessed, and should be deleted (returned to the free pool).

- Unfortunately, the file system has lost track of the files <u>here</u> and <u>there</u>!

Solution 2:   Garbage Collection

- Each disk block is given a special bit.

- The block is said to be "marked" when the bit is 1, and "unmarked" when it is 0.

- When the free pool is low or empty, the following steps are executed:

Garbage Collection

Step 1: Scan the entire disk, and <u>unmark</u> every block (ie, set its bit to 0).

Step 2: Starting at the root directory, traverse the directory structure, and <u>mark</u> every block of every file (ie, set its bit to 1)

<u>Note</u>: Any blocks left unmarked are free (ie, not pointed to by any directory).

Step 3: Scan the entire disk, and add any unmarked blocks to the free pool.

Note: Garbage collection is expensive (since it scans the entire disk).

A Practical Compromise: Use both techniques.

- Cycles are rare, so reference counters will work for a long time.

- Periodically (eg, once a week, at night), run a garbage collection to sweep any unreferenced cycles into the free pool.