ECS 36A, June 2, 2023

Announcements

1. The due date for both Homework 3 and Homework 4 (and Extra Credits 3 and 4) are due June 8
2. The final exam is on June 9, in the classroom, from 10:30am–12:30pm
3. *Do not dally or wait until the last minute for these!*

Review of Linux in the CSIF

- Logging in
- Looking around the file system

Logging into the CSIF

- You *must* use your University login name and password
  - That’s what you type to the Central Authentication System
- Use the Library VPN
  - See the web page https://library.ucdavis.edu/vpn/ for how to do this
- Here is the command:
  - `ssh your-cas-name@pcnn.cs.ucdavis.edu`

  where *nn* is a number between 01 and 43.
- To find the status of systems, look here:
  - http://iceman.cs.ucdavis.edu/nagios3/cgi-bin/status.cgi?hostgroup=all

Copying Files to the CSIF

Do these from your laptop or your other system:

- 1 file (called *localfile* to emphasize it's on your computer) from your computer to the CSIF:

```
scp localfile pcnn.cs.ucdavis.edu:csif_directory
```

- Example: `scp wordsort1.c pc14.cs.ucdavis.edu:.`
  - The "." means current directory, which for scp is your home directory

Copying Files from the CSIF

Do these from your laptop or your other system:

- 1 file (called *localfile* to emphasize it's on your computer) from the CSIF to your computer:

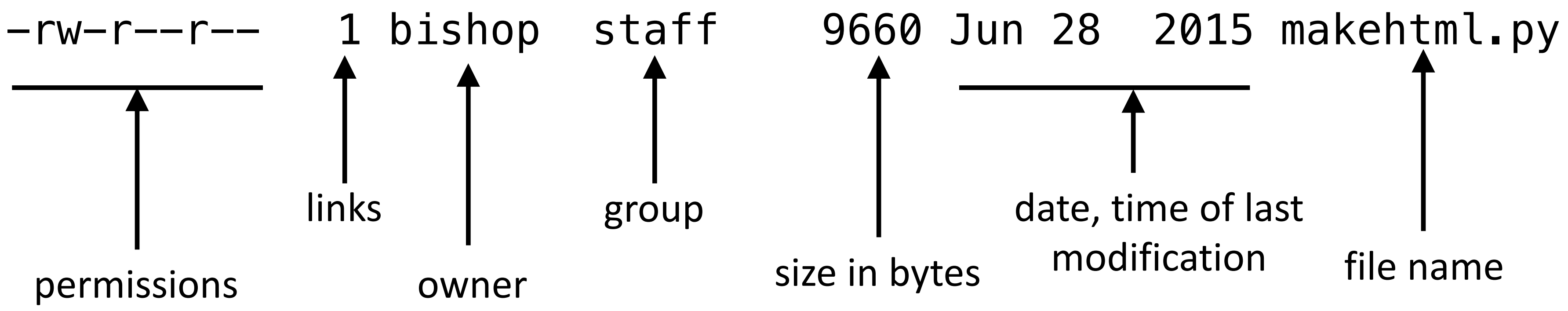
```
scp pcnn.cs.ucdavis.edu:full_path_name local_file
```

- Example: `scp pc14.cs.ucdavis.edu:wordsort1.c .`
  - The "." means current directory, which is the directory in which you execute *scp*

Seeing What Is There

- ls
  - Lists contents of the current directory (except anything that begins with “.”)
- ls *dir*
  - Lists contents of directory *dir* (except anything that begins with “.”)
- Useful options:
  - -a: list the contents *including* anything that begins with “.”
  - -l: list information about the contents, including permissions, size, owner, group
  - -R: list the contents recursively

ls –l



```
-rw-r--r--   1 bishop  staff    9660 Jun 28  2015 makehtml.py
```

Special Directories

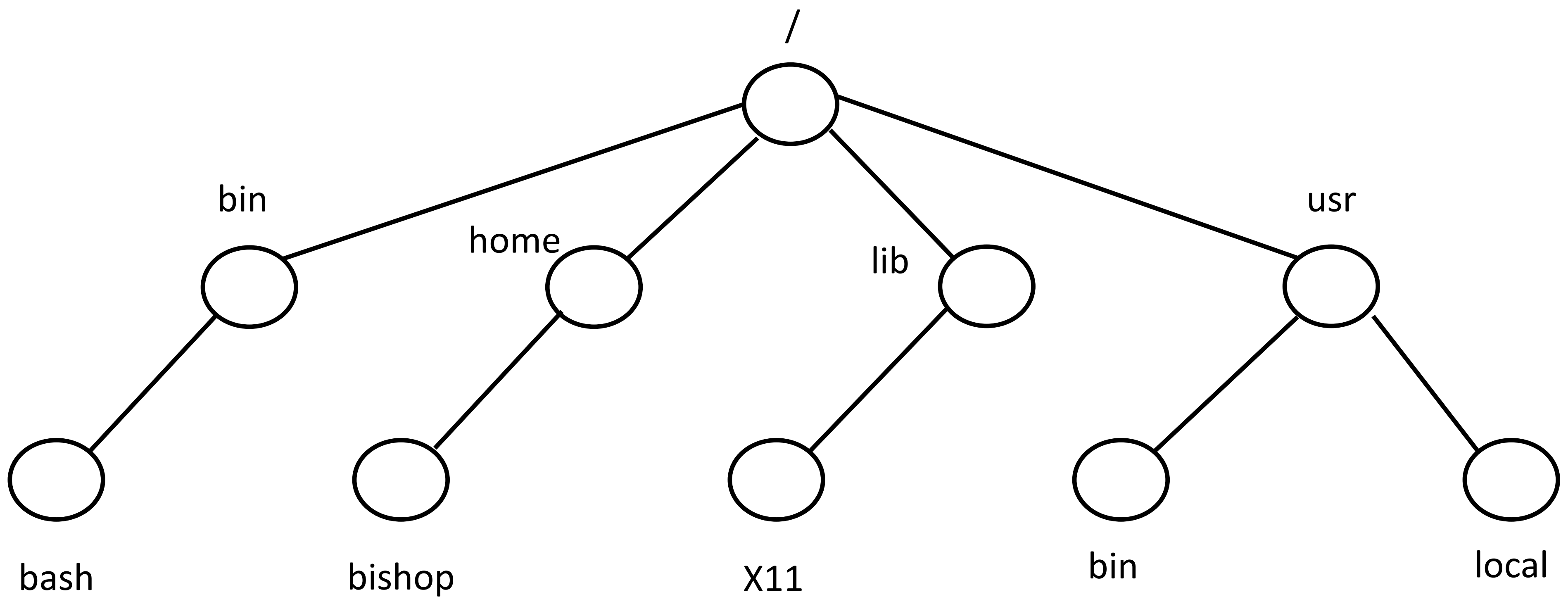
- /
  - Root directory
- .
  - Current working directory
- . .
  - Parent directory, except in /; . . in / is / as it has no parent
- ~
  - Home directory (actually a synonym, not a real directory)

Looking at Files

- cat *file*
  - Shows the contents of *file*
- more *file*
  - Shows the contents of *file* with pagination, so it doesn't scroll off the screen
  - *less* is a variant of *more* but the idea is the same
- head [–*n*] *file*
  - Show the first *n* lines of *file* ; if *n* is not given, it defaults to 10
- tail [–*n*] [–f] *file*
  - Show the last *n* lines of *file* ; if *n* is not given, it defaults to 10
  - The –f option displays lines added to the file (useful for growing files)

Directories

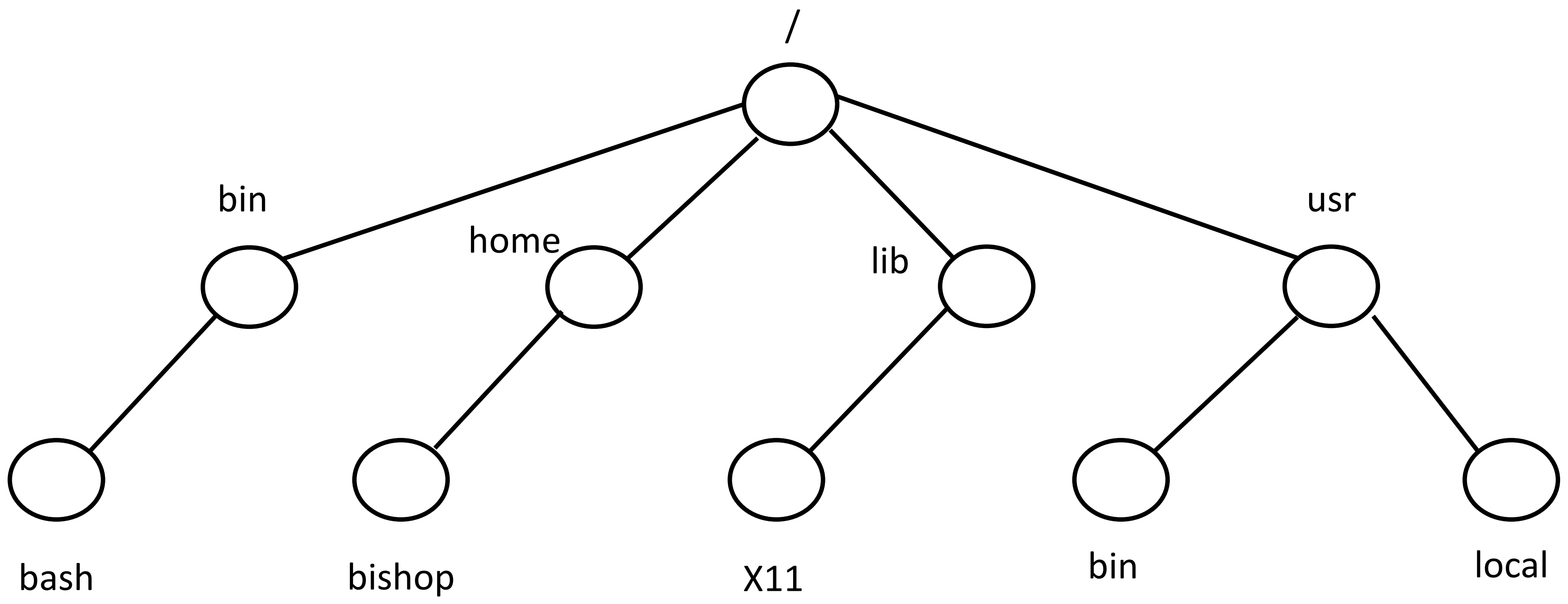
- Like folders; contains files and subdirectories
- File system is best thought of as an upside-down tree:



Full (Absolute) Path Names

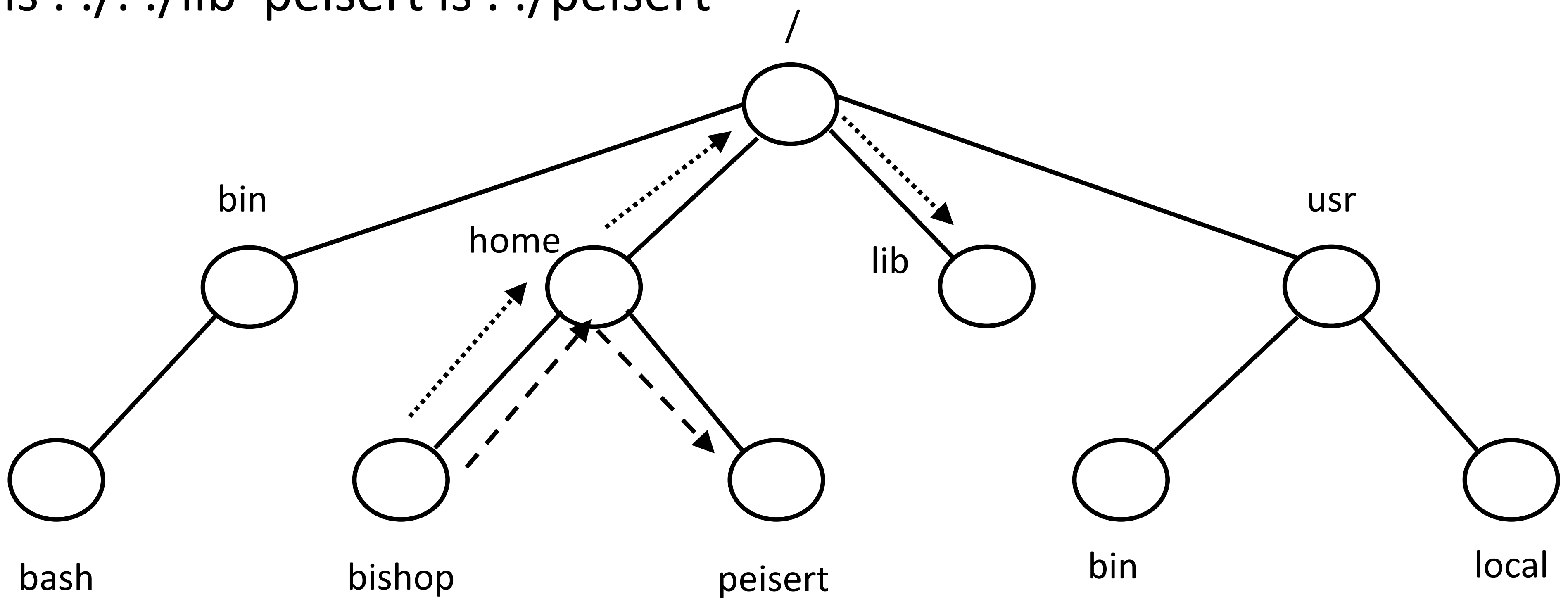
- Start at / and work down, separating directories with ”/”

/bin/bash /home/bishop /lib/X11 /usr/bin /usr/local



Relative Path Names

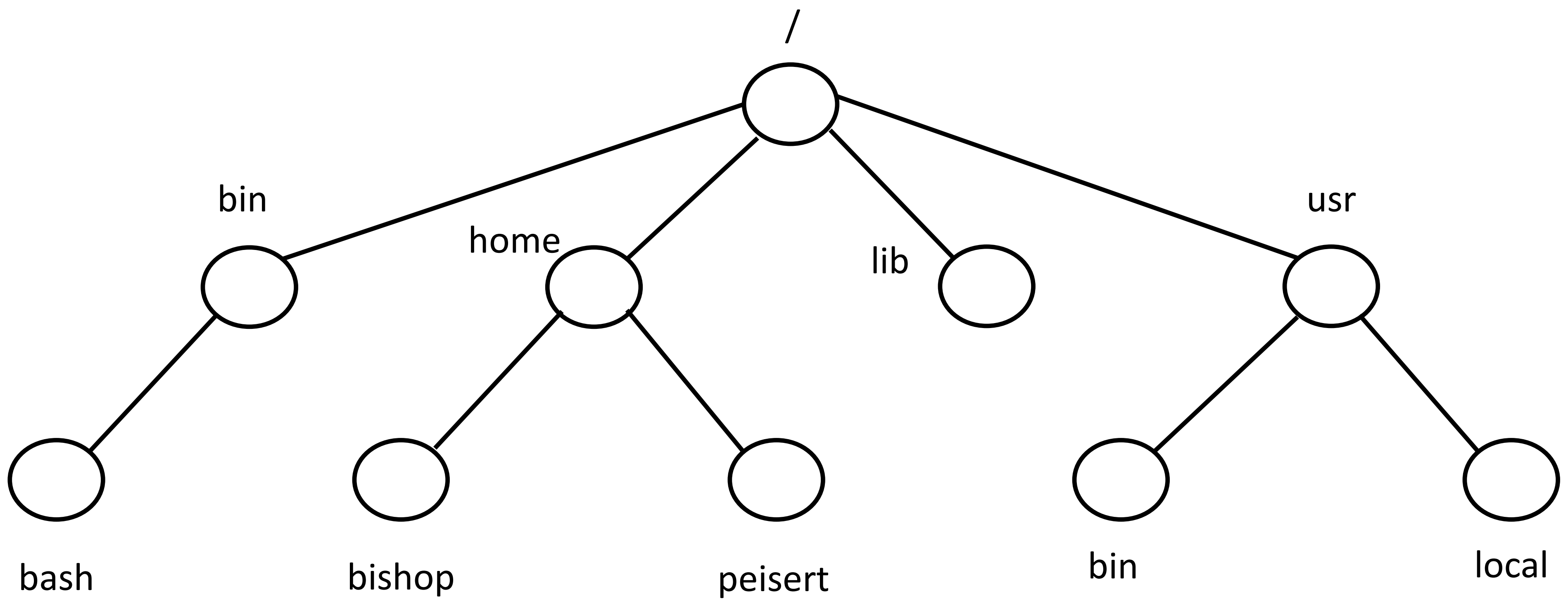
- Start at current working directory and go up and down the tree
- Assume current working directory is /home/bishop
- lib is . ./. ./lib peisert is . ./peisert



Moving Around

- cd *dir* — change current working directory to *dir*

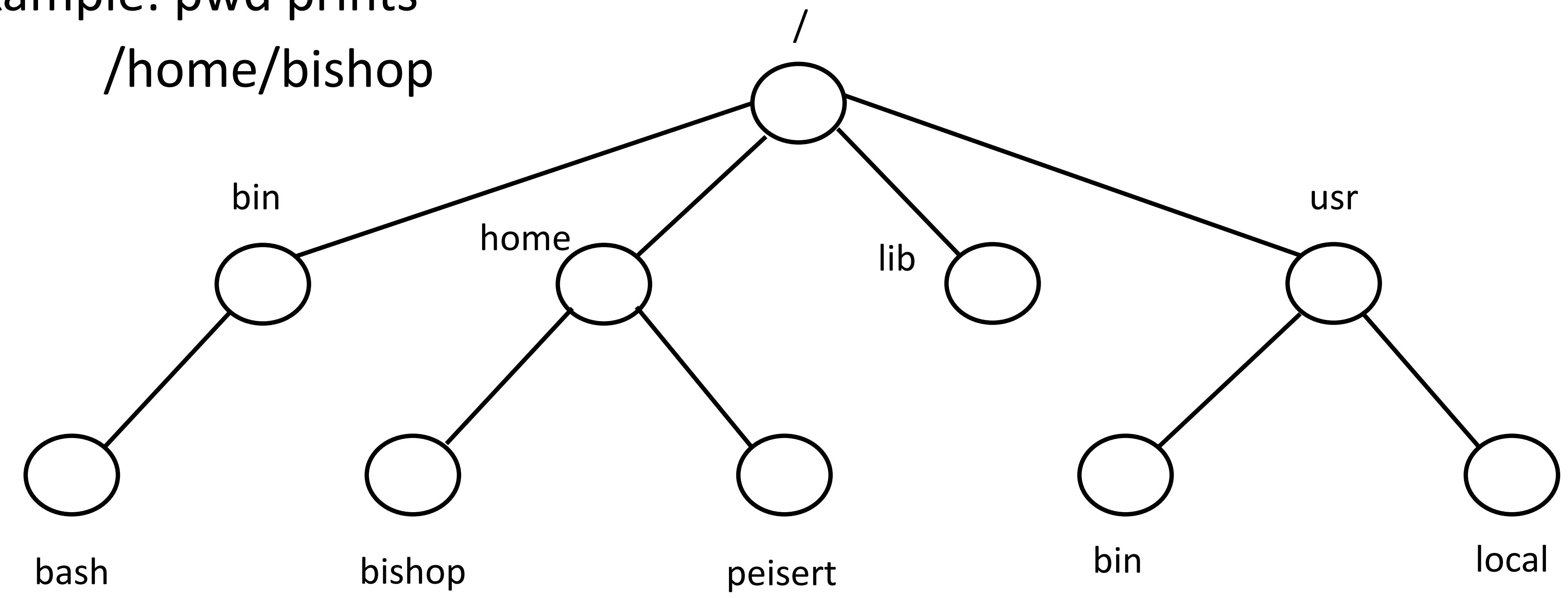
cd /home/peisert   cd . ./peisert



Finding Where You Are

- pwd – print working directory
  - Always gives the full path name
- Example: pwd prints

  /home/bishop



Creating and Removing Directories

- mkdir *dir*
  - Creates directory *dir* (either full or relative path name)
  - Fails if *dir* exists
- rmdir *dir*
  - Removes directory *dir* (either full or relative path name)
  - Fails if *dir* does not exist
  - Fails if *dir* is not empty (except for “.” and “. .”)

Removing Files

- rm *file*
  - Removes file *file* (either full or relative path name)
- **Very Dangerous!!!!!**
  - rm abc*
    - Delete all files in the current directory that begin with "abc"
  - rm abc *
    - Delete the file abc and *then all files in the current directory!!!*
- Best way to do this: rm –i *file*
  - -i interactive; for each file, ask if it should be deleted

Creating Files

- Usually done with a text editor or other program
- You can always do this:
- touch *file*
  - If *file* does not exist, create it
  - If *file* exists, update its time of last access and time of last modification
  - You must have write permission on *file*
  - On some older systems, you *must* also be the owner

Copying Files

- cp *srcfile destfile*
  - Copy the contents of *srcfile* to *destfile*
  - If *destfile* is a directory, a copy of *srcfile* is placed in it
  - If *destfile* is a file and exists, its contents are deleted
- cp –r *srcdir destdir*
  - Copy the directory *srcdir* and its contents to the directory *destdir*
- cp –i *src dest*
  - Copy src to dest as above, but if any file would be overwritten during the copying, ask if it is to be deleted *before* overwriting it

Moving (Renaming) Files

- mv *srcfile destfile*
  - Move *srcfile* to *destfile*
  - If *destfile* is a directory, *srcfile* is moved into it
  - If *destfile* is a file and exists, it is deleted
- mv –n *srcfile destfile*
  - As above, but if *destfile* exists, do not overwrite it
- mv –i *src dest*
  - Move src to dest as above, but if any file would be deleted during the moving, ask if it is to be deleted *before* deleting it