

Slide 1

### Administrivia

- Reminder: Homework 5 programming problems due Monday.
- Link to `mail-files` script moved. If you downloaded a copy early in the semester, please get a new one — the one first posted was wrong.

Slide 2

### Homework 2 Programming Problems — Review

- Problems graded and grades sent out (finally!).
- For the only-shared-variables program:
  - If you don't declare shared variable `volatile`, an optimizing compiler may do things to trip you up (and in fact many of you didn't, and it did).
  - Almost no one got the use of memory fences right (*before* reads and *after* writes).
- For the program using one of the library synchronization mechanisms, it's almost surely not necessary to both with either of those, since the library implementation should take care of it.
- I guess I wasn't surprised by how many people didn't think to change the "No synchronization" comment. If you did, good for you!

Slide 3

### Minute Essay From Last Lecture

- A few people reported trouble with the first couple of problems, especially the first one. May be interesting for me to grade?
- A few people mentioned the last question as tricky or interesting.
- Some people mentioned that for some problems they had trouble finding answers or similar problems in the textbook. I did rather mean for them to push you to apply some of what was in the book/lectures to somewhat-new situations!

Slide 4

### Filesystem Reliability — Backups

- (This was meant to be part of Monday's lecture but we ran out of time.)
- Why do backups? sometimes data is more valuable than physical medium, and might need to
  - Recover from disaster (rare these days, but possible).
  - Recover from stupidity (less rare – hence "recycle bin" idea).
- Many issues involved — which files to back up, how to store backup media, etc., etc. — see textbook.

Slide 5

### “Shared Libraries” — Recap/Review

- Recall the basic idea — allow processes to share one copy in memory of library code. “DLLs” in Windows; “shared libraries” in UNIX/Linux.
- A few details about how this works in Linux . . .
- Object files can be grouped into libraries two ways, as “archive files” and as “shared libraries”. Can have both, one for totally static linking (self-contained executables) and another for dynamic linking.

Slide 6

### Packaging Library Code in Linux — “Archive Files”

- For static linking (each executable has its own copy).
- Compile to object code as usual and make library file (name `libname.a`) with `ar` command. (Examples in `/usr/lib`.)
- Link with `-lname`. By default, searches system directories. Use `-L` to add other directories.
- (Example.)

Slide 7

### Packaging Library Code in Linux — “Shared Libraries”

- For load-time or even run-time linking.
- Compile to “position-independent code” and make library file with `gcc -shared` command. Naming scheme is somewhat elaborate and involves a library name (`libname.so`), an “soname” (library name followed by version number), and a filename (soname followed by minor version number). In the filesystem, usually the first two are symbolic links. (Examples in `/usr/lib`.)
- Link with `-lname`. By default, searches system directories. Use `-L` to add other directories.
- At execution time, system directories are searched to find library name. To add other directories, can use environment variable `LD_LIBRARY_PATH`. Or link with `-rpath` to hardcode where to search.
- Can even do all of the linking at runtime.

Slide 8

- (Example.)

### Minute Essay

- (If you answered this Monday, okay to just say so. I meant to defer it until today, but it was in the slides on the Web, so I guess many were confused?)

If you have a system that supports multiple different file systems (such as Linux with Samba to access Windows files), what problems might arise in copying files between different file systems?

(We had an interesting problem many years ago with backing up `/users` to an OS X machine because the default for OS X is case-insensitive.)

Slide 9

### Minute Essay Answer

- Case sensitivity is one source of potential problems. Other potential problems include restrictions on what characters can appear in filenames and what notion of file ownership and permissions is supported.

Slide 10