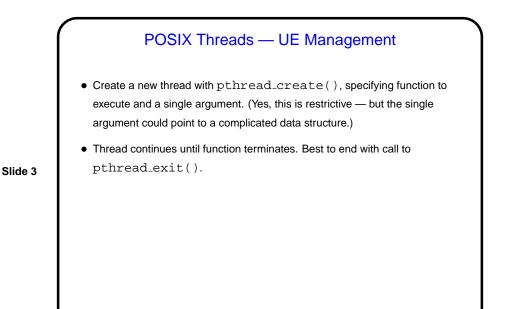


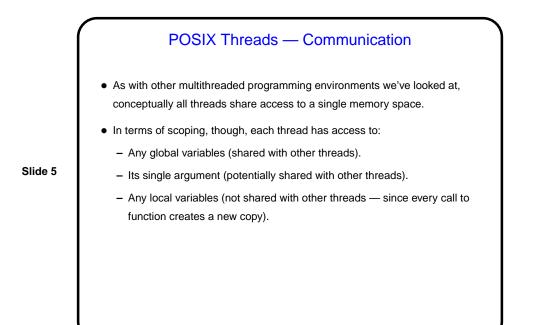
Multithreaded Programming with POSIX Threads
POSIX threads ("pthreads"): widely-available set of functions for multithreaded programming, callable from C/C++. ("POSIX" is Portable Operating System Interface, a set of IEEE standards defining an API for UNIX-compatible systems. Implemented to varying degrees by most UNIX-like systems; implementations also exist for other systems — e.g., Cygwin for Windows.)
Same ideas as multithreaded programming with OpenMP and Java, but not as nicely packaged (my opinion). At one time probably more widely available than OpenMP compilers, though that has probably changed with gcc OpenMP support.

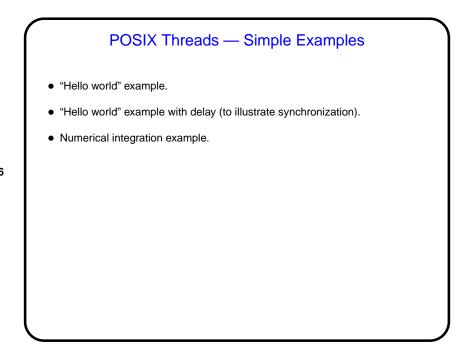
Slide 2

Slide 4

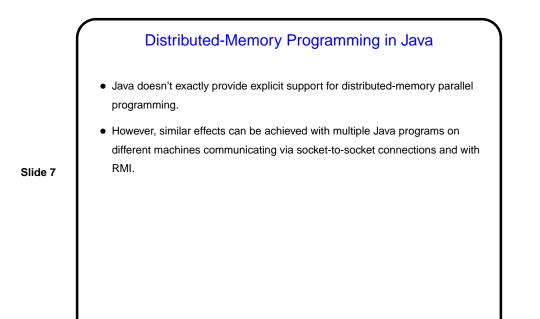


POSIX Threads — Synchronization
• pthread_join() waits until another thread finishes — similar to join
in Java's Thread class.
• Various synchronization mechanisms:
 Mutexes (locks): pthread_mutex_init(),
 pthread_mutex_destroy(), pthread_mutex_lock(),
 pthread_mutex_unlock().
 Condition variables: pthread_cond_init(),
 pthread_cond_destroy(), pthread_cond_wait(),
 pthread_cond_signal().
 Semaphores: sem_init(), sem_destroy(), sem_wait(),
 sem_post().





Slide 6



Distributed-Memory Programming in Java Using Sockets

• Client/server model:

- Server sets up "server socket" specifying port number, then waits to accept connections. Connection generates socket.
- Client connects to server by giving name/IPA and port number generates a socket.
- On each side, get input/output streams for socket. Program must define protocol for the two sides to communicate.

Slide 8

