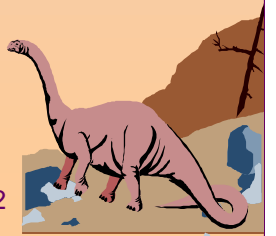


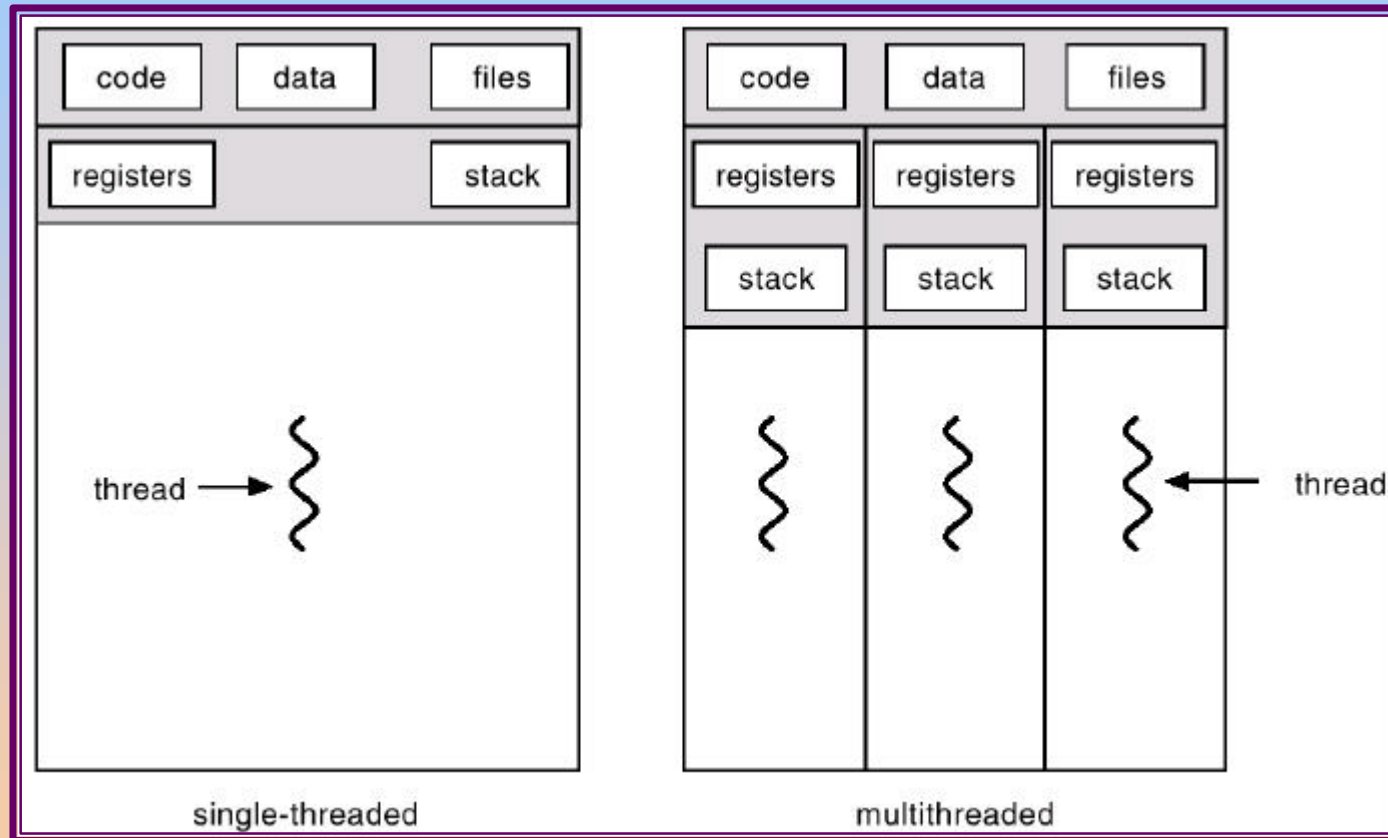


Chapter 5: Threads

- n Overview
- n Multithreading Models
- n Threading Issues
- n Pthreads
- n Solaris 2 Threads
- n Windows 2000 Threads
- n Linux Threads
- n Java Threads



Single and Multithreaded Processes





Benefits

- n Responsiveness
 - n Resource Sharing
 - n Economy
 - n Utilization of MP Architectures
- 




User Threads

- 
- n Thread management done by user-level threads library
 - n Examples
 - POSIX *Pthreads*
 - Mach *C-threads*
 - Solaris *threads*



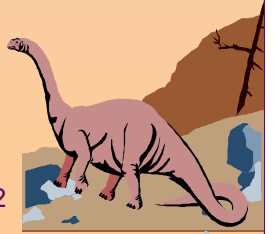
Kernel Threads

- 
- n Supported by the Kernel
 - n Examples
 - Windows 95/98/NT/2000
 - Solaris
 - Tru64 UNIX
 - BeOS
 - Linux



Multithreading Models

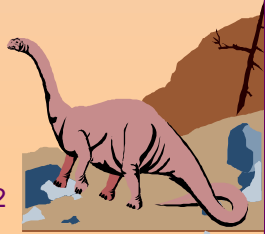
- n Many-to-One
- n One-to-One
- n Many-to-Many



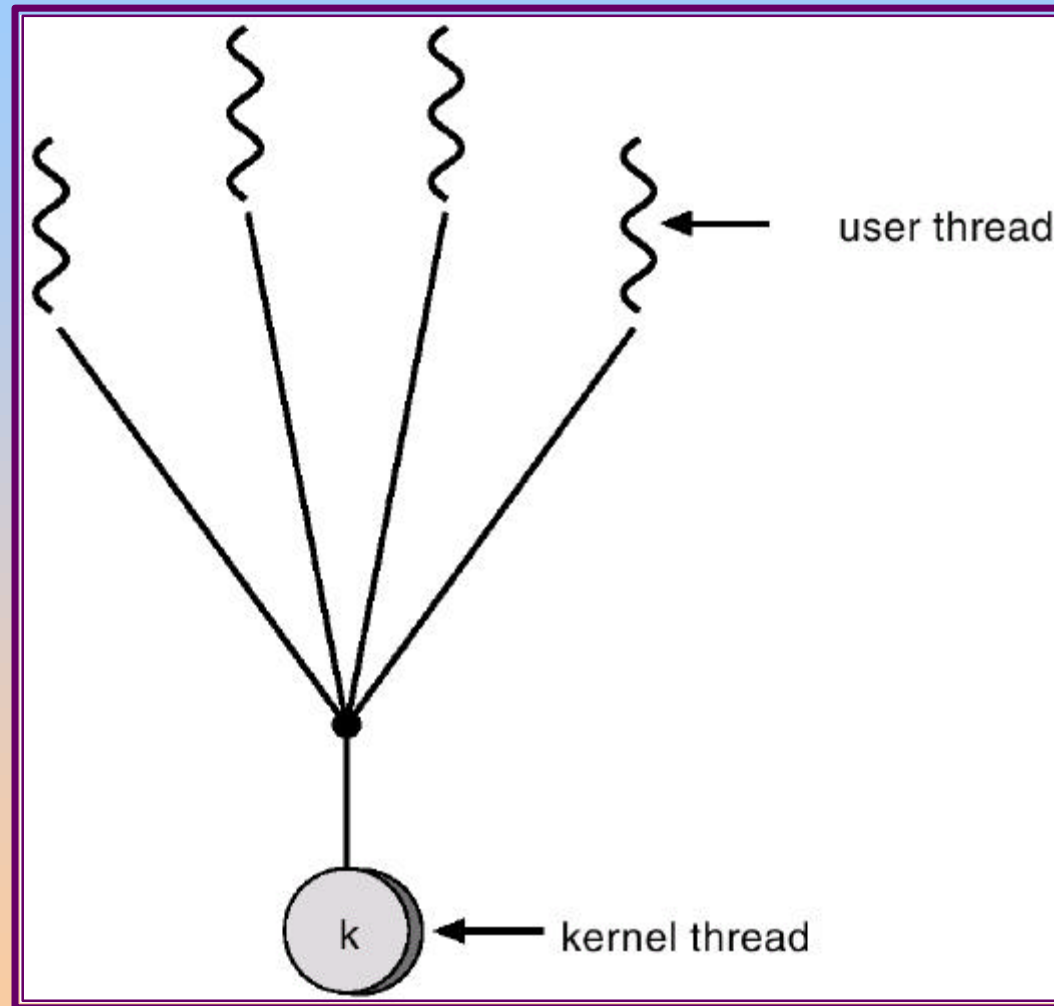


Many-to-One

- n Many user-level threads mapped to single kernel thread.
- n Used on systems that do not support kernel threads.




Many-to-One Model

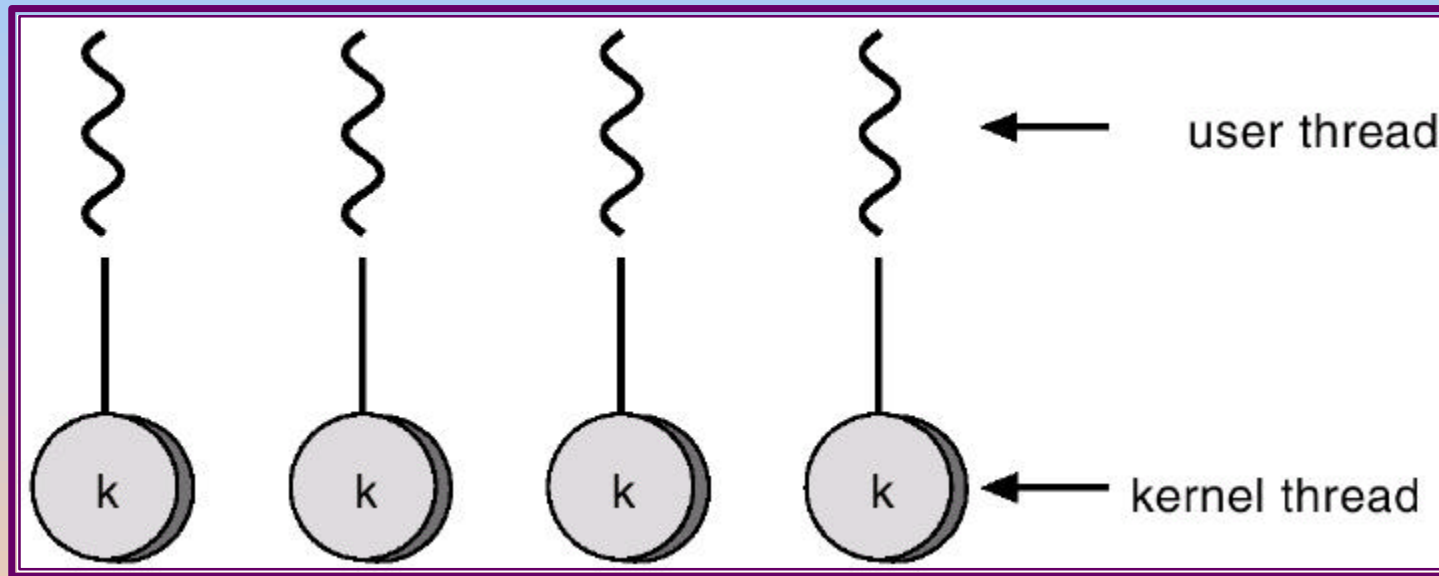




One-to-One


- 
- n Each user-level thread maps to kernel thread.
 - n Examples
 - Windows 95/98/NT/2000
 - OS/2

One-to-one Model

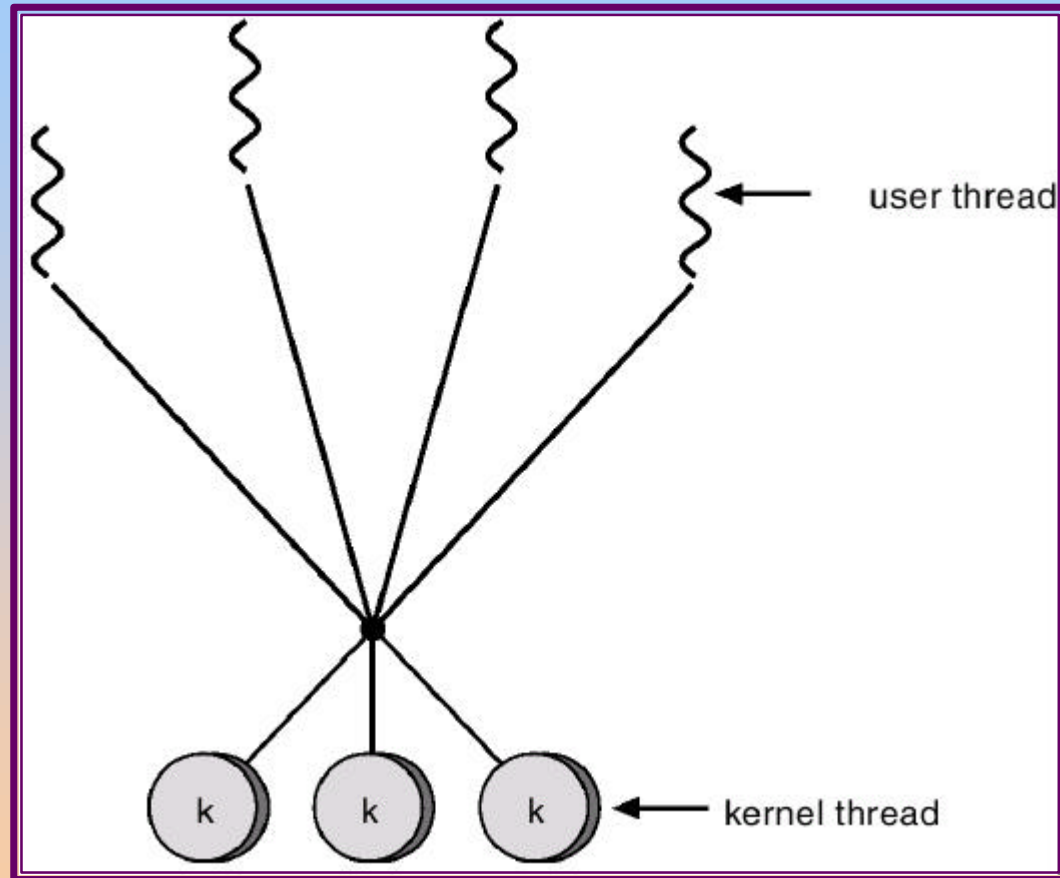




Many-to-Many Model

- 
- n Allows many user level threads to be mapped to many kernel threads.
 - n Allows the operating system to create a sufficient number of kernel threads.
 - n Solaris 2
 - n Windows NT/2000 with the *ThreadFiber* package

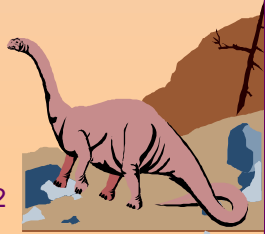
Many-to-Many Model






Threading Issues

- n Semantics of fork() and exec() system calls.
- n Thread cancellation.
- n Signal handling
- n Thread pools
- n Thread specific data

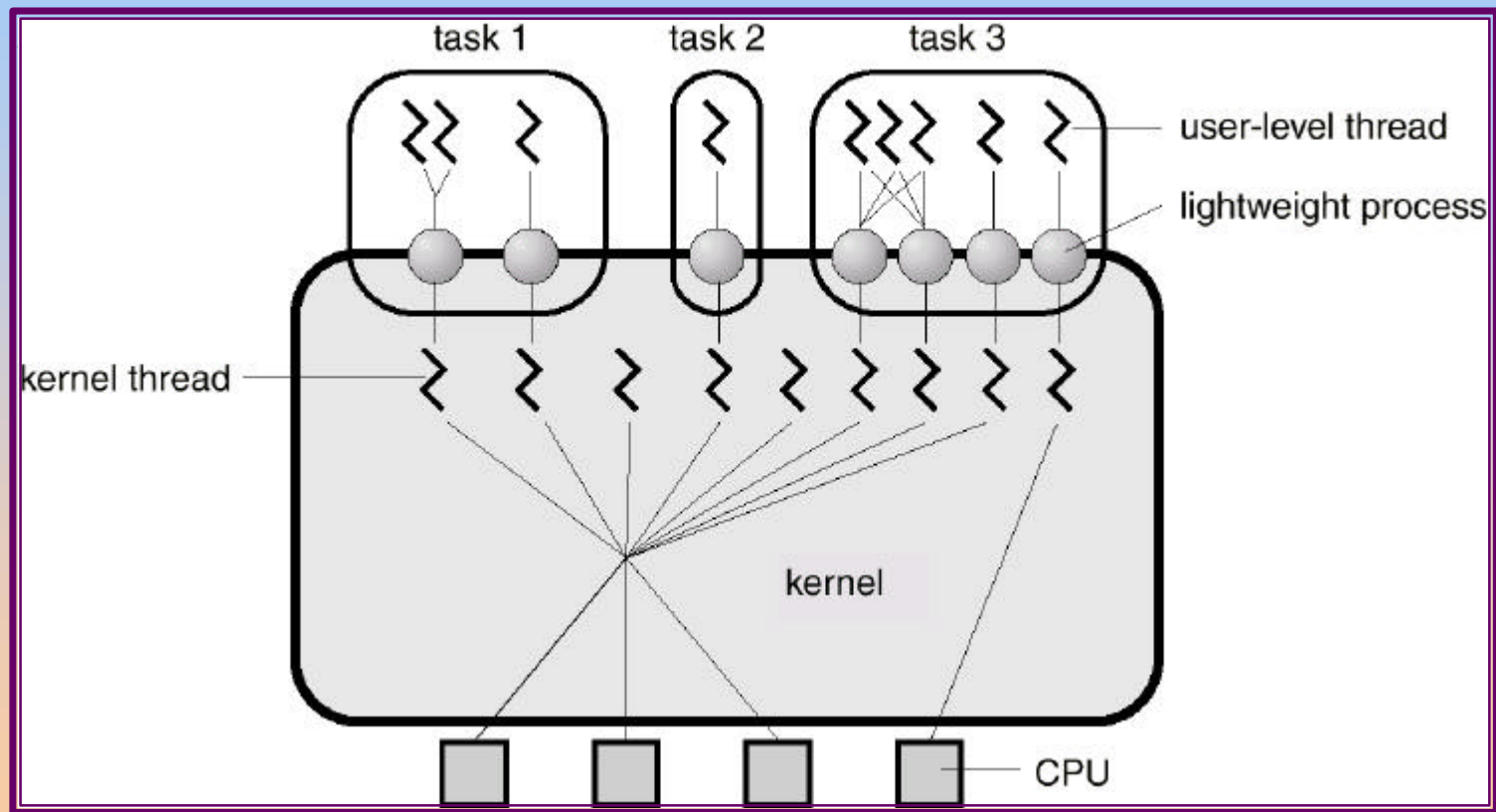




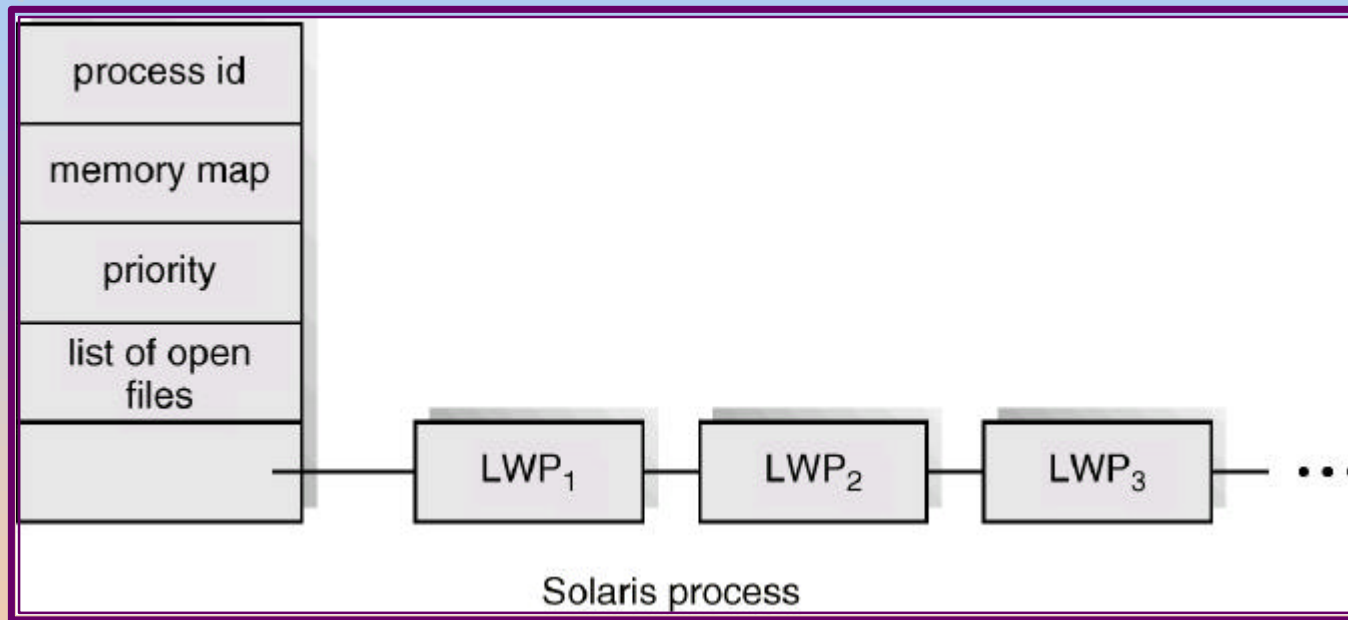
Pthreads

- 
- n a POSIX standard (IEEE 1003.1c) API for thread creation and synchronization.
 - n API specifies behavior of the thread library, implementation is up to development of the library.
 - n Common in UNIX operating systems.

Solaris 2 Threads




Solaris Process






Windows 2000 Threads

- 
- n Implements the one-to-one mapping.
 - n Each thread contains
 - a thread id
 - register set
 - separate user and kernel stacks
 - private data storage area



Linux Threads

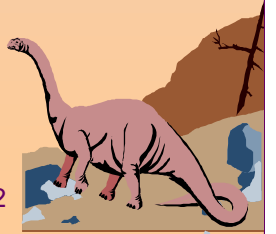
- 
- n Linux refers to them as *tasks* rather than *threads*.
 - n Thread creation is done through clone() system call.
 - n Clone() allows a child task to share the address space of the parent task (process)



Java Threads



- n Java threads may be created by:
 - F Extending Thread class
 - F Implementing the Runnable interface
- n Java threads are managed by the JVM.



Java Thread States

