

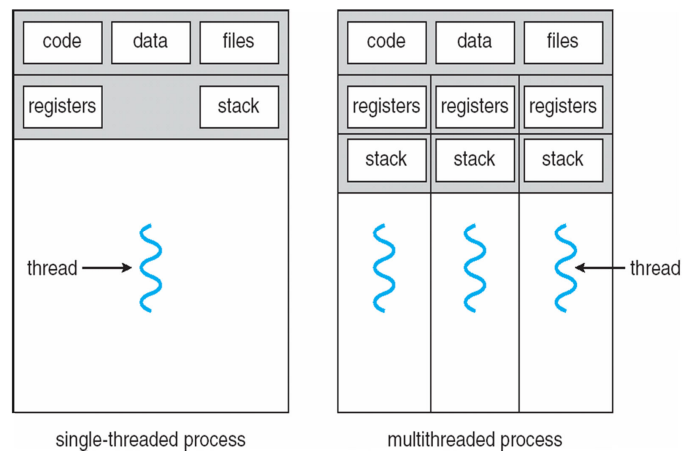
Chapter 5: Threads

Definition

- ❑ A thread is a flow of control within a process. A multithreaded process contains several different flows of control within the same address space.
- ❑ For example
A word processor may have a thread for displaying graphics, another thread for responding to keystrokes from the user, and a third thread for performing spelling and grammar checking in the background.

Slide 2

Single and Multithreaded Processes



Slide 3

Benefits of Threads

- ❑ The benefits of multithreading include increased responsiveness to the user, resource sharing within the process, economy, and the ability to take advantage of multiprocessor architectures.

Slide 4

Benefits of Threads

- ❑ Takes less time to create a new thread than a process
- ❑ Less time to terminate a thread than a process
- ❑ Less time to switch between two threads within the same process
- ❑ Threads enhance efficiency in communication between different executing programs. Since threads within the same process share memory and files, they can communicate with each other without invoking the kernel

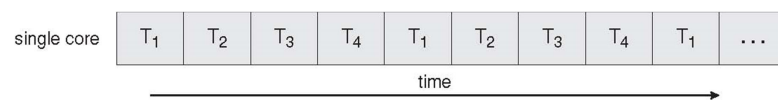
Slide 5

Multicore Programming

- ❑ Multicore systems putting pressure on programmers, challenges include
 - ❑ **Dividing activities**
 - ❑ **Balance**
 - ❑ **Data splitting**
 - ❑ **Data dependency**
 - ❑ **Testing and debugging**

Slide 6

Concurrent Execution on a Single-core System



Slide 7

Parallel Execution on a Multicore System

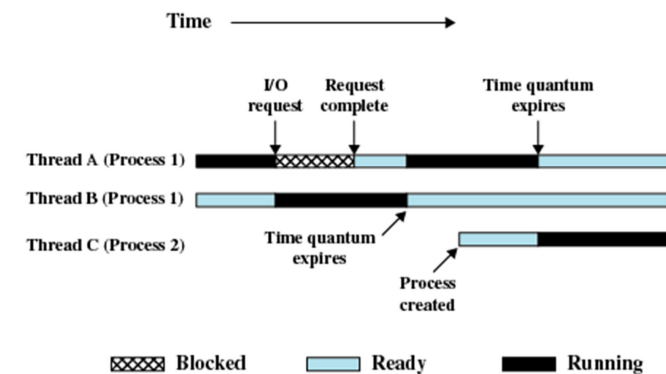


Figure 4.4 Multithreading Example on a Uniprocessor

Slide 8

User Threads

- ❑ Thread management done by user-level threads library
- ❑ Three primary thread libraries:
 - ❑ POSIX **Pthreads**
 - ❑ Win32 threads
 - ❑ Java threads

Slide 9

Kernel Threads

- ❑ Supported by the Kernel
- ❑ Examples
 - ❑ Windows XP/2000
 - ❑ Solaris
 - ❑ Linux
 - ❑ Tru64 UNIX
 - ❑ Mac OS X

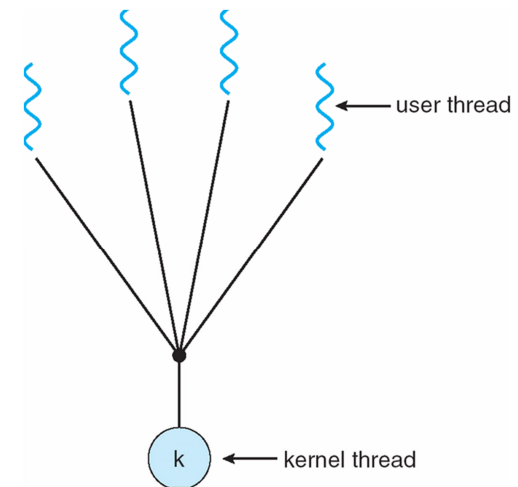
Slide 10

Multithreading Models

- ❑ Many-to-One
- ❑ One-to-One
- ❑ Many-to-Many

Slide 11

Many-to-One Model



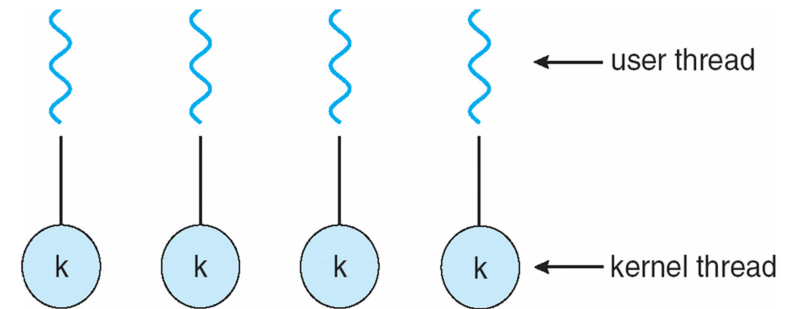
Slide 12

Many-to-One

- ❑ Many user-level threads mapped to single kernel thread
- ❑ Examples:
 - ❑ Solaris Green Threads
 - ❑ GNU Portable Threads

Slide 13

One-to-one Model



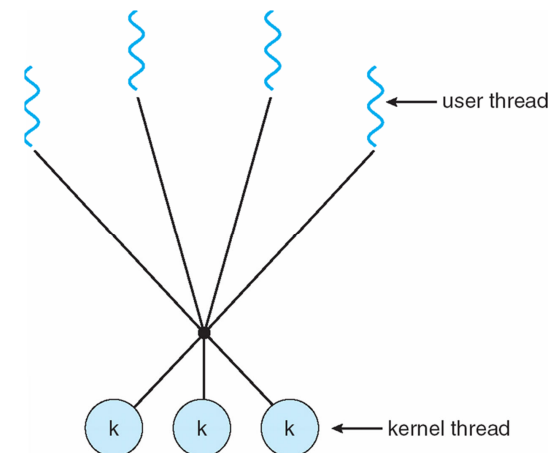
Slide 14

One-to-One

- ❑ Each user-level thread maps to kernel thread
- ❑ Examples
 - ❑ Windows NT/XP/2000
 - ❑ Linux
 - ❑ Solaris 9 and later

Slide 15

Many-to-Many Model



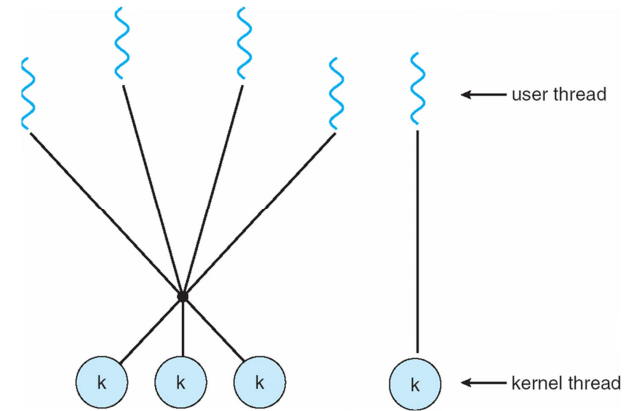
Slide 16

Many-to-Many Model

- ❑ Allows many user level threads to be mapped to many kernel threads
- ❑ Allows the operating system to create a sufficient number of kernel threads
- ❑ Solaris prior to version 9
- ❑ Windows NT/2000 with the *ThreadFiber* package

Slide 17

Two-level Model



Slide 18

Thread Libraries

- ❑ **Thread library** provides programmer with API for creating and managing threads
- ❑ Two primary ways of implementing
 - ❑ Library entirely in user space
 - ❑ Kernel-level library supported by the OS

Slide 19