

Chapter 5: Threads

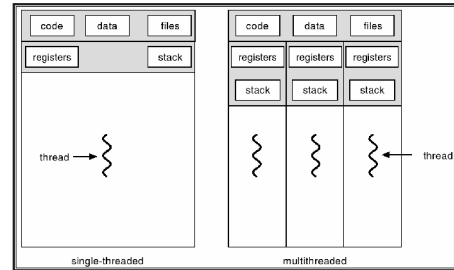
- Overview
- Multithreading Models
- Threading Issues
- Pthreads
- Solaris 2 Threads
- Windows 2000 Threads
- Linux Threads
- Java Threads

Operating System Concepts

5.1

Silberschatz, Galvin and Gagne ©2002

Single and Multithreaded Processes



Operating System Concepts

5.2

Silberschatz, Galvin and Gagne ©2002

Benefits

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

Operating System Concepts

5.3

Silberschatz, Galvin and Gagne ©2002

User Threads

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

Operating System Concepts

5.4

Silberschatz, Galvin and Gagne ©2002

Kernel Threads

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

Operating System Concepts

5.5

Silberschatz, Galvin and Gagne ©2002

Multithreading Models

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

Operating System Concepts

5.6

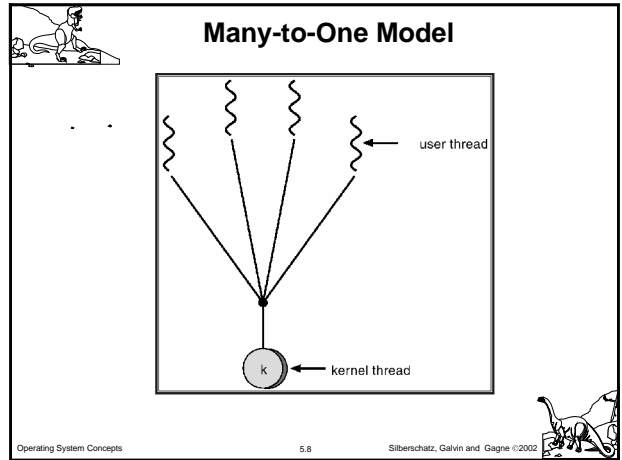
Silberschatz, Galvin and Gagne ©2002

Many-to-One

- Many user-level threads mapped to single kernel thread.

Used on systems that do not support kernel threads.

Operating System Concepts 5.7 Silberschatz, Galvin and Gagne ©2002



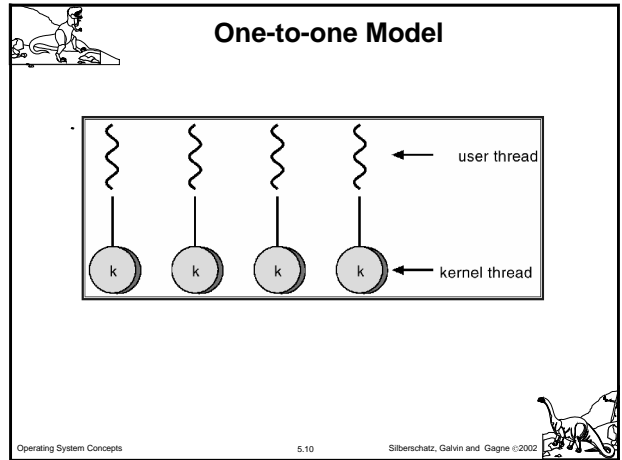
One-to-One

- Each user-level thread maps to kernel thread.

Examples

- Windows 95/98/NT/2000
- OS/2

Operating System Concepts 5.9 Silberschatz, Galvin and Gagne ©2002



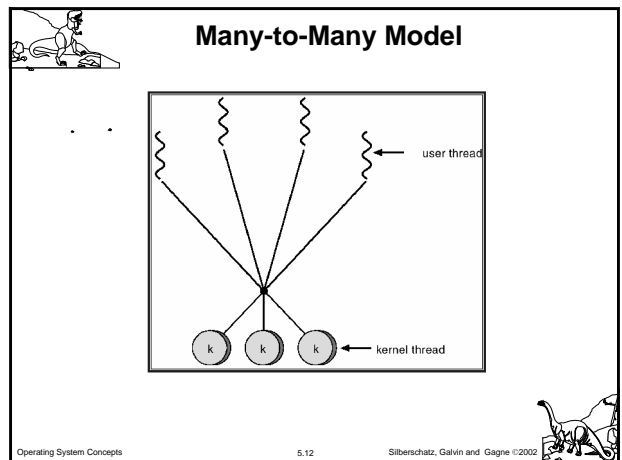
Many-to-Many Model

- 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 2
Windows NT/2000 with the *ThreadFiber* package

Operating System Concepts 5.11 Silberschatz, Galvin and Gagne ©2002



Threading Issues

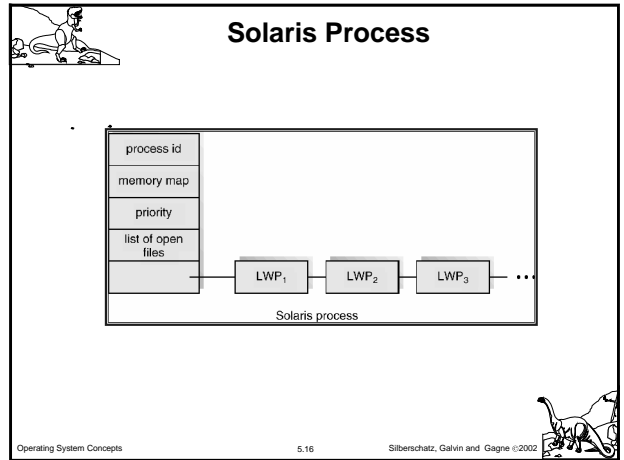
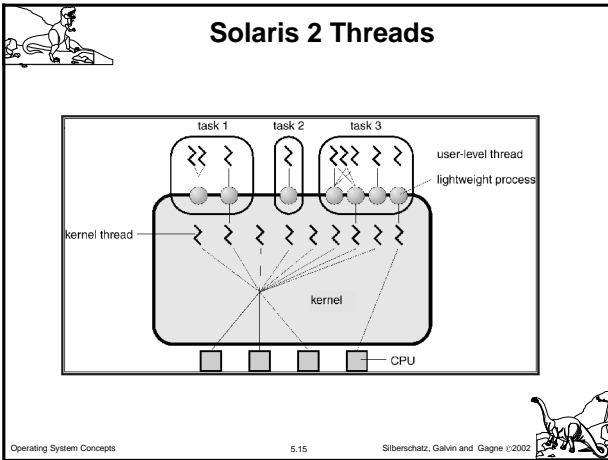
- Semantics of fork() and exec() system calls.
- Thread cancellation.
- Signal handling
- Thread pools
- Thread specific data

Operating System Concepts 5.13 Silberschatz, Galvin and Gagne ©2002

Pthreads

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

Operating System Concepts 5.14 Silberschatz, Galvin and Gagne ©2002



Windows 2000 Threads

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

Operating System Concepts 5.17 Silberschatz, Galvin and Gagne ©2002

Linux Threads

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

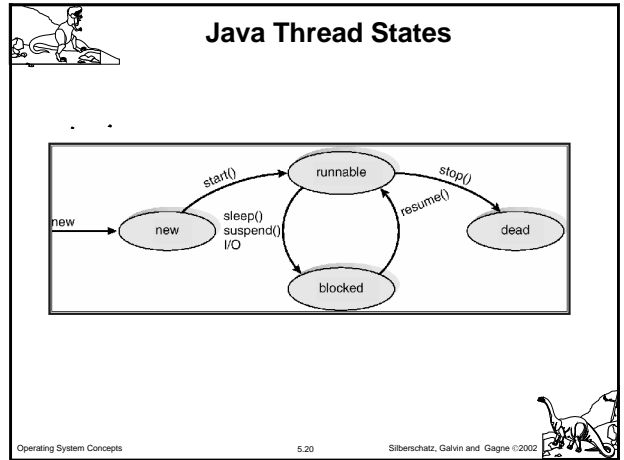
Operating System Concepts 5.18 Silberschatz, Galvin and Gagne ©2002

Java Threads

- Java threads may be created by:
 - Extending Thread class
 - Implementing the Runnable interface

Java threads are managed by the JVM.

Operating System Concepts 5.19 Silberschatz, Galvin and Gagne ©2002



This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.