Quizlet

Operating Systems Chapter 4: Threads

Study online at quizlet.com/_21x7z1

1.	**Benefits**	Responsiveness□ Resource Sharing□ Economy□ Utilization of MP Architectures
2.	**Kernel Threads**	Supported by the Kernel Examples: Windows, XP/2000, Solaris, Linux, Tru64 UNIX, Mac OS X
3.	**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)
4.	**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 Examples: Solaris prior to version 9, Windows NT/2000 with the ThreadFiber package
5.	**Many-to-One**	Many user-level threads mapped to single kernel thread Examples: Solaris Green Threads GNU Portable Threads
6.	**Multithreading Models**	Many-to-One□ One-to-One□ Many-to-Many
7.	**One-to-One**	Each user-level thread maps to kernel thread Examples: Windows NT/XP/2000 Linux Solaris 9 and later
8.	**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 (Solaris, Linux, Mac OS X)
9.	**Thread Cancellation**	Terminating a thread before it has finished
10.	**Threading Issues**	Semantics of fork() and exec() system calls Thread cancellation Signal handling Thread pools Thread specific data Scheduler activations
11.	**Three primary thread libraries:**	POSIX Pthreads Win32 threads Java threads
12.	**Two general approaches:**	Asynchronous cancellation terminates the target thread immediately Deferred cancellation allows the target thread to periodically check if it should be cancelled
13.	**Two-level Model**	Similar to M:M, except that it allows a user thread to be bound to kernel thread Examples: IRIX HP-UX Tru64 UNIX Solaris 8 and earlier
14.	**User Threads**	Thread management done by user-level threads library
15.	**Windows XP Threads**	Implements the one-to-one mapping Each thread contains A thread id Register set Separate user and kernel stacks Private data storage area The register set, stacks, and private storage area are known as the context of the threads The primary data structures of a thread include: ETHREAD (executive thread block) KTHREAD (kernel thread block) TEB (thread environment block)