The image contains a presentation slide with the following content:

### Threads

**Topics**
- Definition
- Advantages using Threads
- User and Kernel Threads
- Multithreading Models
- Java and Solaris Threads
- Examples

### Threads

- Light weight processes
  Every thread is a different stream of control - executing instructions independently.

- Multi-Threading
  One program to do multiple tasks concurrently.

### Differences

- Process structure is in kernel space.
- Thread usually consists of registers, stack and some data.
- Rest of the process structure is shared by all threads (address space, file descriptor)
- Thread structure is usually in user space.
Advantages

1. Responsiveness - may allow a thread to run even if some other threads of the same task are blocked.
2. Resource sharing - threads share the memory and the resources of the process to which they belong.
3. Economy - Allocating memory and resources for the process is costly.

Advantages (Contd)

4. Utilization of multiprocessor architecture - If more than one processor is present, multiple threads could be executed at the same time.

User and Kernel Threads

User threads - they are supported above the kernel and are implemented by a thread library at the user level.

Kernel threads - they are supported at the operating system or kernel level. Thread creating, scheduling and management are done by the kernel in the kernel space.
Multithreading Models
- Many to one model - Many user threads mapped one kernel thread. Execution is efficient. But entire process will block if any one of the thread is blocked on a system call.
- One-to-one thread - One kernel thread for each user thread.
- Many to many thread - Many user threads mapped onto many kernel threads.

Java Threads - API
- Thread Creation - start method, run method
- Thread Management - suspend, sleep, resume and stop methods.
States of threads: new, running, blocked and done.

Java Thread Examples
- Simple threads.
- Producer - Consumer
- Clock applet from the book.
Solaris Threads
User level threads with a library containing API's for thread creation and management.

Between user threads and kernel threads are light weight processes.
The thread library multiplexes user-level threads on the pool of LWP's for the process and only user level threads connected to a LWP accomplish work.

Solaris Threads - Contd
A bound user-level thread is permanently attached to LWP. Only this thread runs on the LWP and by request the LWP can be dedicated to a single processor.

All unbound thread in an application are multiplexed on a processor.