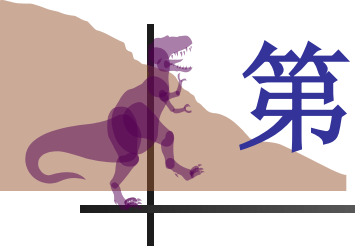


第4章 线程 Threads

- 本章目标 Chapter Objectives
  - 介绍线程的组成并比较线程与进程

    Identify the basic components of a thread, and contrast threads and processes.

4.1 概述 Overview

- 在操作系统中引入进程的目的是使多道程序能并发执行，以改善资源利用率及提高系统吞吐量；
- 在操作系统中再引入线程，则是为了减少程序并发执行所付出的时空开销，使操作系统具有更好的并发性。

4.1.1 动机 Motivation

- 进程具有两个属性：

  The concept of a process as embodying two characteristics :

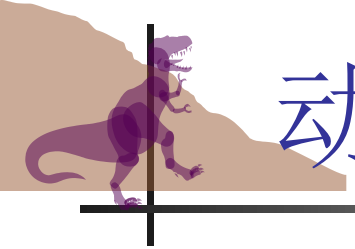
  - 拥有资源的独立单位

    Unit of Resource ownership

  - 调度和分派的基本单位

    Unit of Dispatching

- 为使进程并发执行，则必须进行诸如创建、撤消、切换等一系列操作，这些操作涉及到资源管理，所花费的时空开销较大，为此引入了线程。



动机2 Motivation

- 操作系统独立对待这两个属性

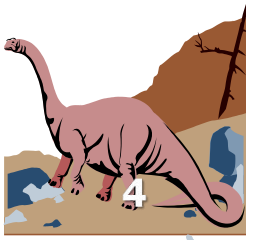
  These two characteristics are treated independently by the operating system

  - 调度及分派单位称为线程或轻型进程

    Dispatching is referred to as a thread or lightweight process

  - 拥有资源的单位称为进程或任务

    Resource of ownership is referred to as a process or task



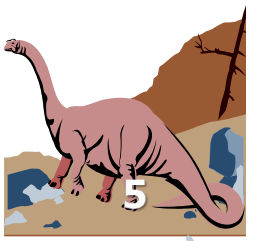
重型进程

- 进程有虚拟地址空间、打开文件及I/O资源等

  Process Have a virtual address space，open files, and I/O resources

- 传统进程或重型进程等价于只有一个线程的任务

  A traditional or heavyweight process is equal to a task with one thread.



线程定义

- 线程是CPU使用的一个基本单元，包括：

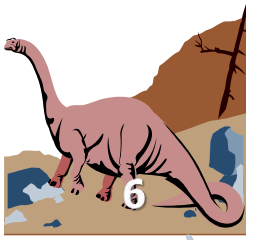
  A thread is a basic unit of CPU utilization, it consists of:

  - 线程标识 **a thread ID**
  - 程序计数器 **program counter**
  - 寄存器集 **register set**
  - 栈 **stack**

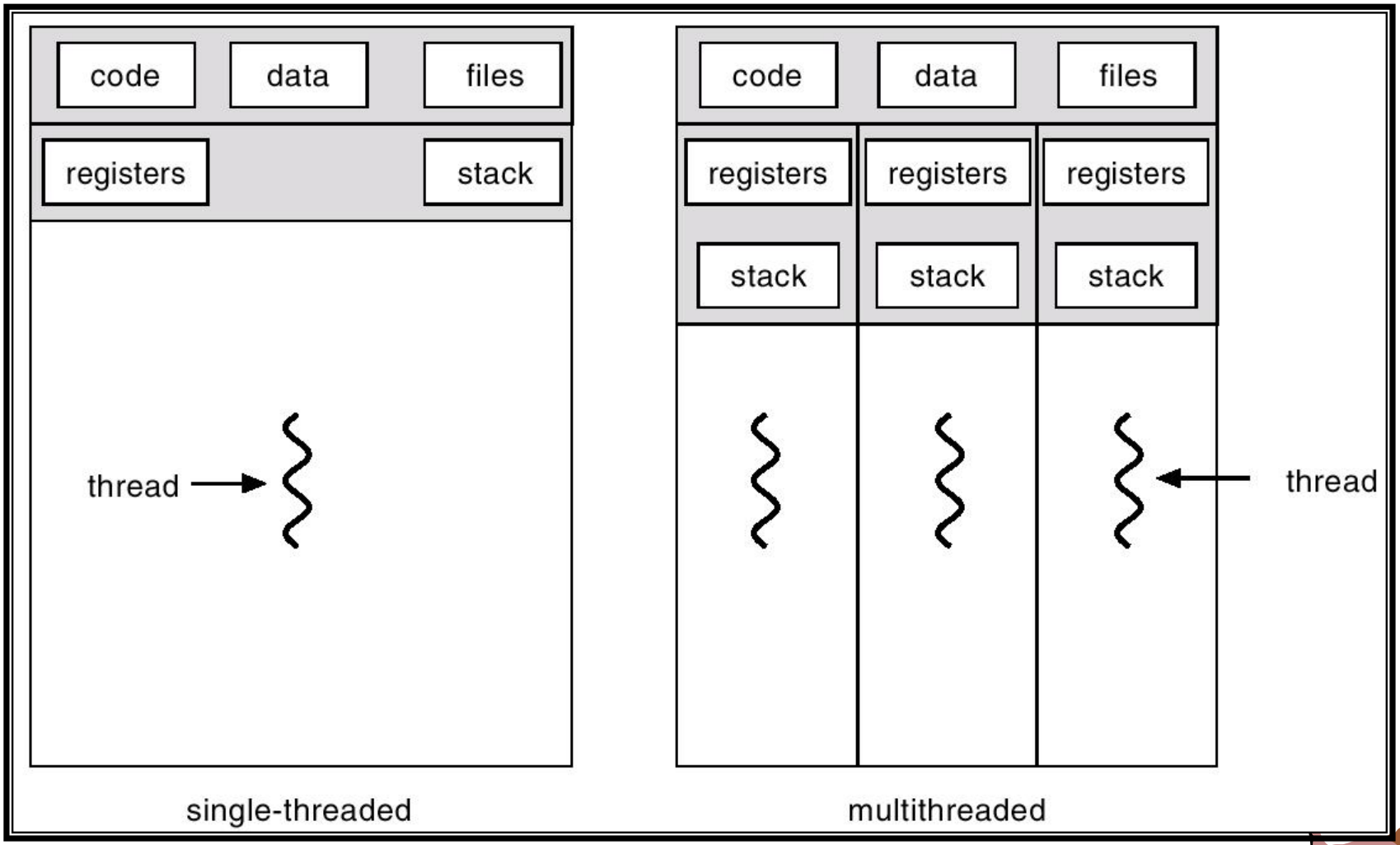
- 线程与属于同一进程的线程共享：

  A thread shares with threads belonging to the same process:

  - 代码段 **code section**
  - 数据段 **data section**
  - 其他操作系统资源 **other operating-system resources**

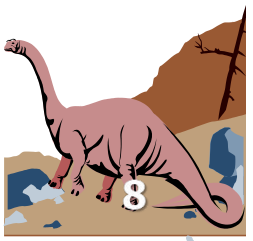


单线程进程和多线程进程
Single and Multithreaded Processes



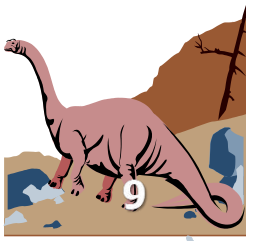
线程其他定义

- 线程的定义情况与进程类似，存在多种不同的提法。下面列出一些较权威的定义：
  - 线程是进程内的一个执行单元。
  - 线程是进程内的一个可调度实体。
  - 线程是程序（或进程）中相对独立的一个控制流序列。
  - 线程是进程内一个相对独立的、可调度的执行单元。



线程拥有资源

- 线程自己基本上不拥有资源，只拥有一点在运行时必不可少的资源（如程序计数器、一组寄存器和栈），
- 但它可以与同属一个进程的其他线程共享进程拥有的全部资源。

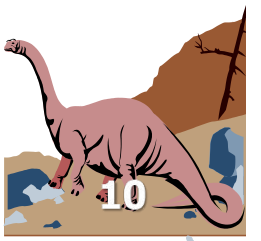


线程拥有的资源（续）

- 进程中的线程具有
  - 执行状态 **execution state**
  - 线程上下文 **thread context**
  - 执行栈 **execution stack**
  - 线程静态存储局部变量

    some per-thread static storage for local variables
  - 寄存器 **register**
  - 对所属进程资源的访问

    access to the memory and resources of its process

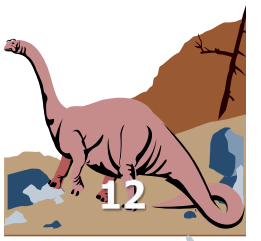


线程的状态

- 和进程类似，线程也有运行、就绪、阻塞等状态。
  - 创建：当创建一个新进程时，也为该进程创建了一个线程。线程还可以创建新线程。
  - 就绪：线程已获得除处理机外的所有资源。
  - 运行：线程正在处理机上执行。
  - 阻塞：线程因等待某事件而暂停运行。
  - 终止：一个线程已完成。
- 线程的同步与通信与进程类似。进程的挂起及终止将影响到其中的所有线程。

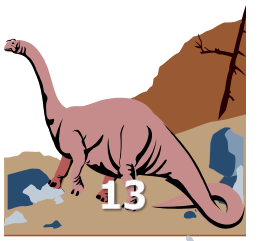
4.1.2 优点 Benefits

- 响应能力强 Responsiveness：**a program to continue running even if part of it is blocked**
- 资源共享 Resource Sharing：**threads share the resources of the process to which them belong**
- 经济性 Economy：**it is more economical to create and context-switch threads**
- 多处理器体系结构的利用 Utilization of MP Architectures：**threads may be running in parallel on different processors**



优点 续

- Takes less time to create (or to terminate) a new thread than a process
- Less time to switch between two threads within the same process
- Since threads within the same process share memory and files, they can communicate with each other without invoking the kernel

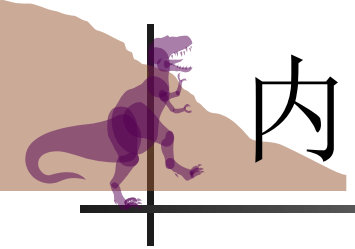


4.3 多线程模型
Multithreading Models

- 操作系统中有多种方式实现对线程的支持：
  - 内核线程　Kernel Threads
  - 用户线程　User Threads
  - 上述两种方法的组合实现

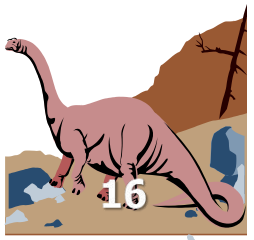
内核线程

- 内核线程（也称内核级线程 kernel-level thread ）是指依赖于内核，由操作系统内核完成创建和撤消工作的线程。
- 在支持内核线程的OS中，内核维护进程和线程的上下文信息并完成线程切换。
- 一个内核线程阻塞时不会影响其他线程的运行。
- 处理机时间分配的对象是线程，所以有多个线程的进程将获得更多处理机时间。



内核线程

- 例子 Examples
  - Windows 95/98/NT/2000
  - Solaris
  - Tru64 UNIX
  - BeOS
  - Linux

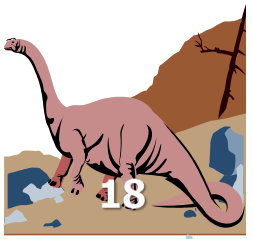


用户线程

- 用户线程（也称用户级线程user-level thread）是指不依赖于操作系统核心，由应用进程利用线程库提供创建、同步、调度和管理线程的函数来控制的线程。
- 用户线程的维护由应用进程完成，可以用于不支持内核线程的操作系统
- 当一个线程阻塞时，整个进程都必须等待，处理机时间是分配给进程的，进程内有多个线程时，每个线程的执行时间相对少一些。

用户线程续

- Examples例子
  - POSIX *Pthreads*
  - Mach *C-threads*
  - Solaris *threads*



两种方法的组合

- 在有些系统中，提供了上述两种方法的组合实现。
- 在这种系统中，内核支持多线程的建立、调度与管理；同时，系统中又提供使用线程库的便利，允许用户应用程序建立、调度和管理用户线程。
- 因此可以很好地将内核线程和用户线程的优点结合起来。

用户线程与内核线程的关系
relationship between user threads and kernel threads

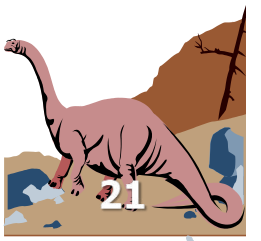
- 多对一　　Many-to-One
- 一对一　　One-to-One
- 多对多　　Many-to-Many

4.3.1 多对一模型 Many-to-One Model

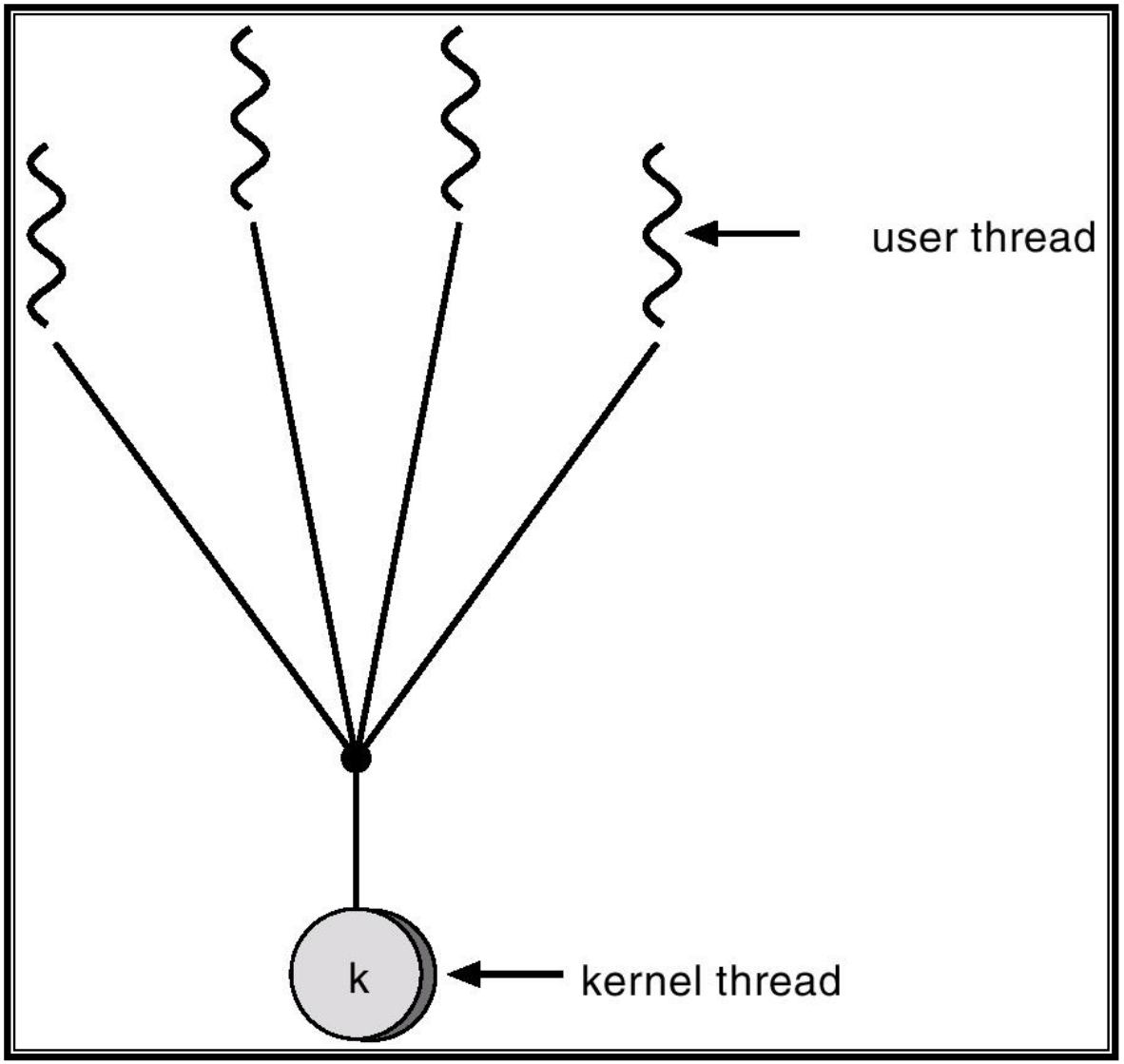
- 多对一模型将多个用户线程映射到一个内核线程

  The many to one model map many user threads to one kernel thread.
- 线程管理由线程库在用户空间进行

  Thread management is done in user space, so it is efficient
- Examples : Green threads -- a thread library available for Solaris 2



多对一模型2
Many-to-One Model



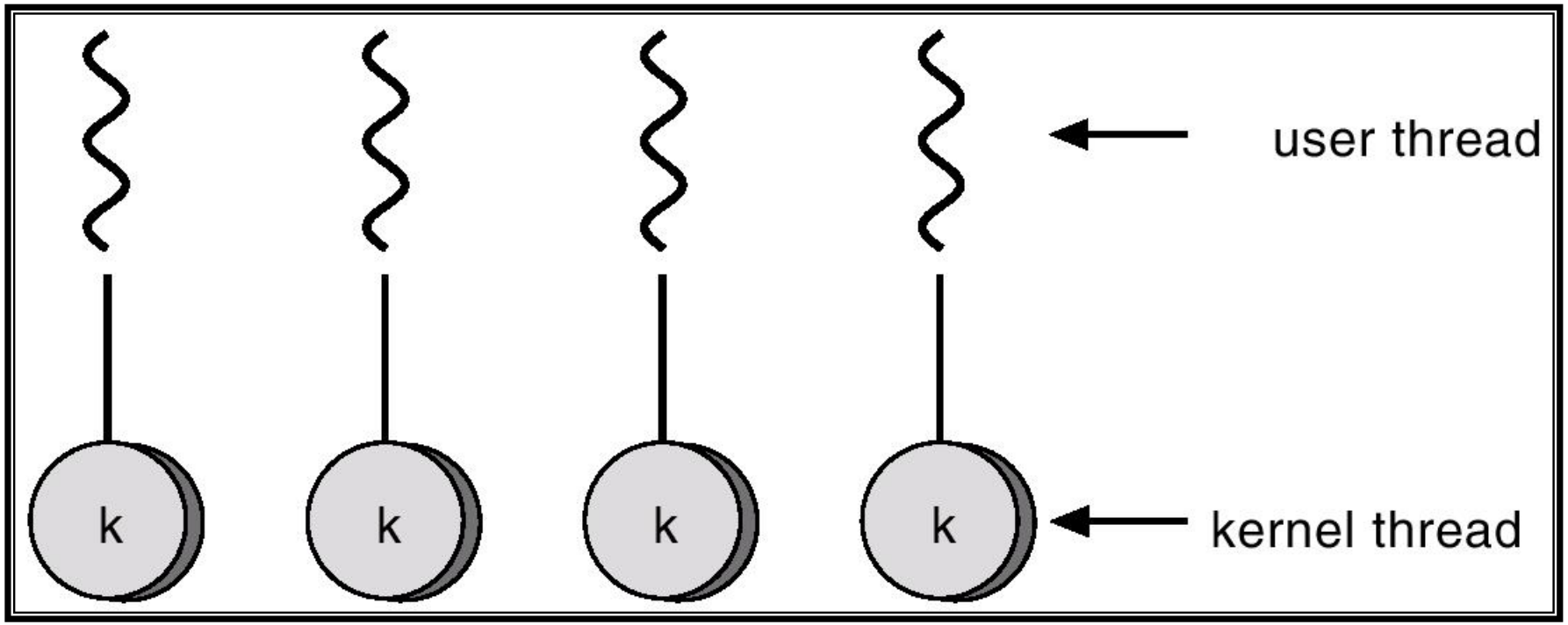
4.3.2 一对一模型
One-to-One Model

- 一对一模型将每个用户线程映射到一个内核线程

  The one to one model map each user-level thread maps to a kernel thread.
- 这种模型的绝大多数实现限制了系统支持的线程数量

  Most implementations of this model restrict the number of threads supported by the system .
- Examples
  - **Windows 95/98/NT/2000**
  - **OS/2**

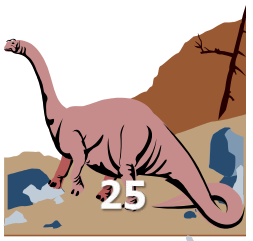
一对一模型2
One-to-One Model



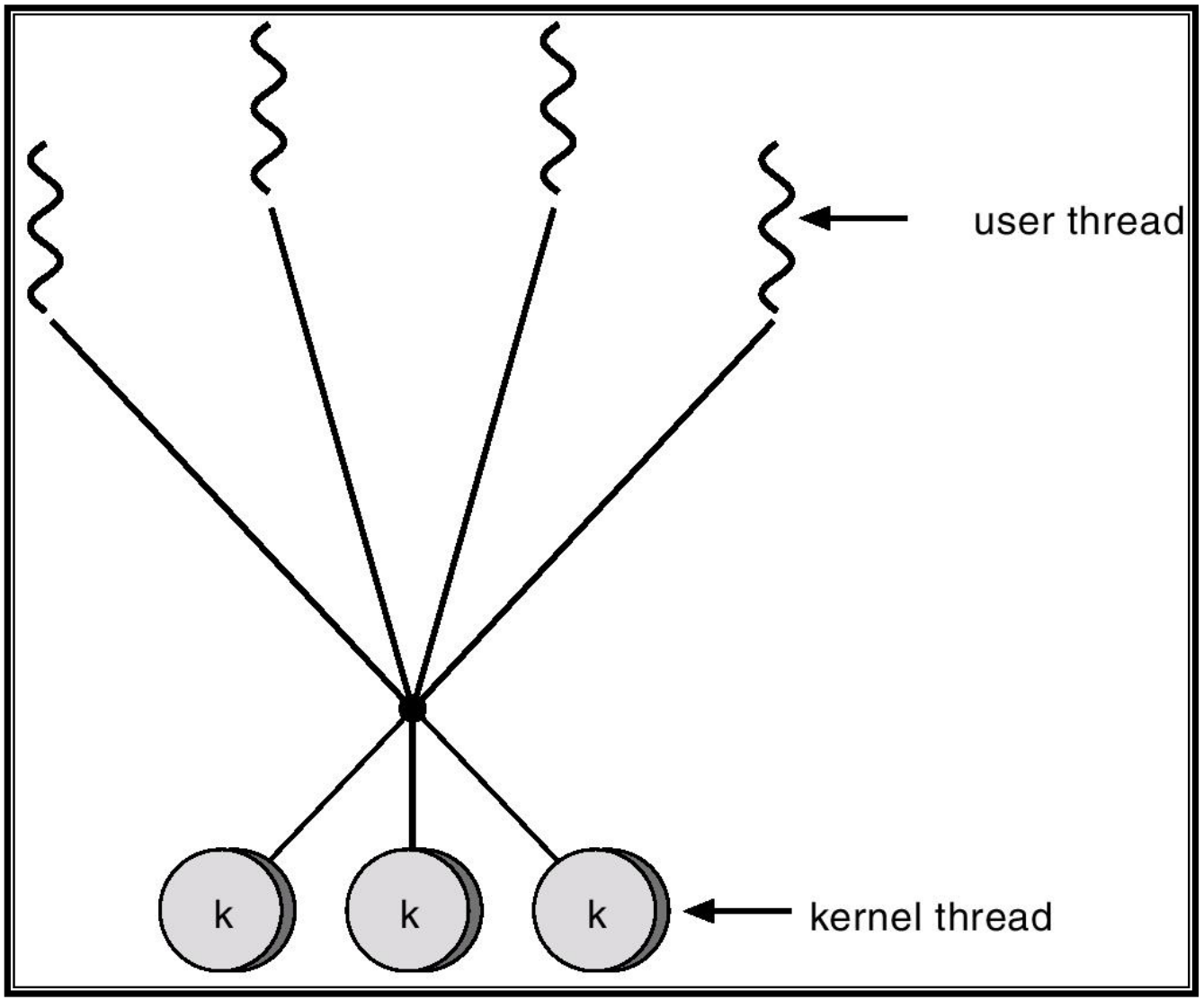
4.3.3 多对多模型
Many-to-Many Model

- 多对多模型多路复用多个用户线程到同样数量或数量更少的内核线程上

  The many to many model multiplexes many user threads to a smaller or equal number of kernel threads.
- Examples
  - Solaris 2
  - Windows NT/2000
  - IRIX
  - HP-UX
  - Tru64 UNIX

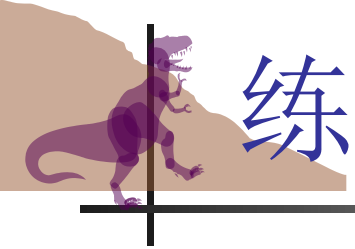


多对多模型2
Many-to-Many Model



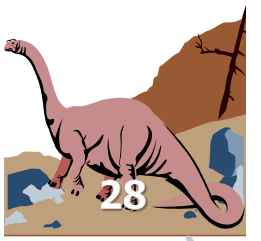
线程与进程的比较

- 调度：在传统OS中，进程是调度和分配资源的基本单位；引入线程后，线程是调度和分派的基本单位，进程是拥有资源的基本单位。
- 拥有资源：进程是拥有资源的基本单位，由一个或多个线程及相关资源构成。
- 并发性：进程之间可以并发执行，同一进程中的各线程之间也可以并发执行。
- 系统开销：进程创建、撤销及切换的开销大于线程。而同一进程的线程间同步与通信开销小。



练习

- 4.6 What resources are used when a thread is created? How do they differ from those used when a process is created?
- 4.10 Which of the following components of program state are shared across threads in a multithreaded process?
  - a. Register values
  - b. Heap memory
  - c. Global variables
  - d. Stack memory



选择题

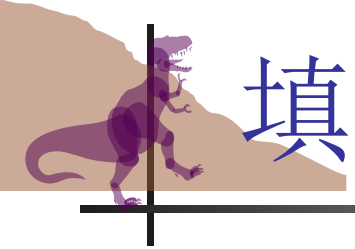
- 关于线程和进程，下列说法中正确的是_____。

A. 线程一定是分配处理机时间的基本单位

B. 进程一定是分配处理机时间的基本单位

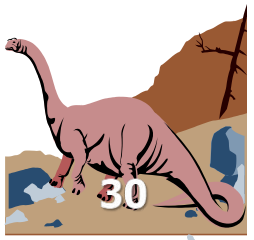
C. 一个线程可以属于多个进程

D. 一个进程可以拥有多个线程



填空题

在操作系统中引入线程概念的主要目的是_____。



考研题1

- 在支持多线程的系统中，进程P创建的若干个线程不能共享的是_____。11

  A、进程P的代码段

  B、进程P中打开的文件

  C、进程P的全局变量

  D、进程P中某线程的栈指针

考研题2

- 下列关于进程和线程的叙述中，正确的是（）12

  A. 不管系统是否支持线程，进程都是资源分配的基本单位

  B. 线程是资源分配的基本单位，进程是调度的基本单位

  C. 系统级线程和用户级线程切换都需要内核的支持

  D. 同一进程的各个线程拥有各自不同的地址空间