

操作系统原理

Operating System Principle

田丽华

4-1 线程概念

The introduction of threads

线程的引入

进程具有二个基本属性：

- 是一个拥有资源的独立单位：它可独立分配虚地址空间、主存和其它
- 又是一个可独立调度和分派的基本单位。
- 这二个基本属性使进程成为并发执行的基本单位
- 在一些早期的OS中，比如大多数UNIX系统、Linux等，进程同时具有这二个属性。

由于进程是一个资源的拥有者，因而在进程创建、撤销、调度切换时，系统需要付出较大的时空开销。

进程的数目不宜过多，进程切换频率不宜过高，限制了并发程度。

The introduction of threads

线程的引入

操作系统的设计目标



- 提高并发度
- 减小系统开销

将进程的两个基本属性分开



对于拥有资源的基本单位，不对其进行频繁切换，对于调度的基本单位，不作为拥有资源的单位，“轻装上阵”

引入线程以小的开销来提高进程内的并发程度。

The introduction of threads

线程的引入

进程:

资源分配单位（存储器、文件）和CPU调度（分派）单位。又称为“任务(task)”

线程:

作为CPU调度单位，而进程只作为其他资源分配单位。

- 只拥有必不可少的资源，如：线程状态、程序计数器、寄存器上下文和栈
- 同样具有就绪、阻塞和执行三种基本状态
- 与同属一个进程的其它线程共享进程拥有的全部资源
- 可并发执行

The introduction of threads

线程的引入

线程的优点：

减小并发执行的时间和空间开销（线程的创建、退出和调度），因此容许在系统中建立更多的线程来提高并发程度。

- 线程的创建时间比进程短；
- 线程的终止时间比进程短；
- 同进程内的线程切换时间比进程短；
- 由于同进程内线程间共享内存和文件资源，可直接进行不通过内核的通信；

线程

- A thread (or lightweight process) is a basic unit of CPU utilization; it consists of:

线程（轻型进程）是CPU运用的一个基本单元，包括

- program counter 程序计数器
- register set 寄存器集
- stack space 栈空间

- A thread shares with its peer threads its:

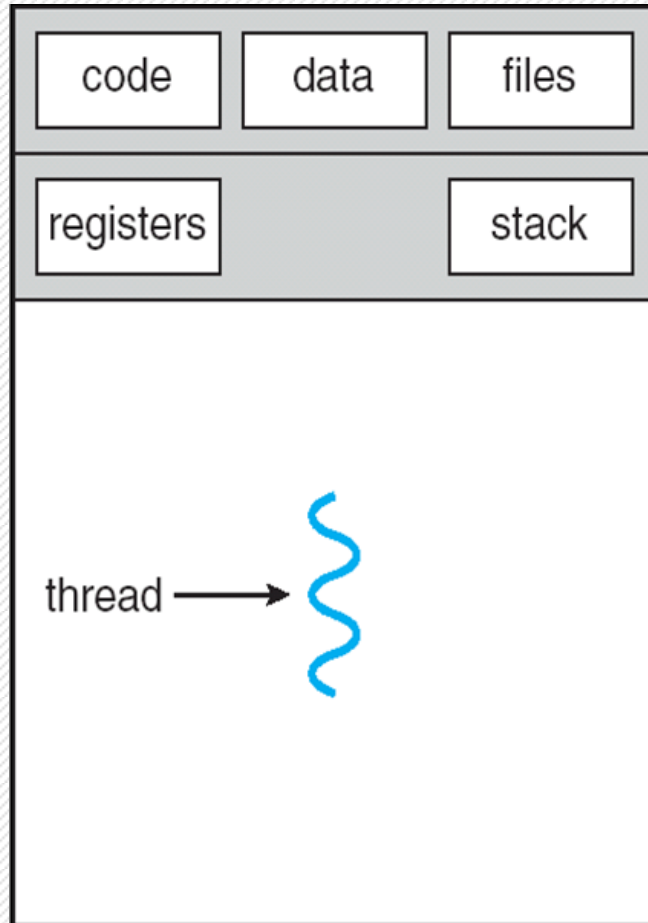
一个线程与它的对等线程共享：

- code section 代码段
- data section 数据段
- operating-system resources 操作系统资源
such as open files and signals

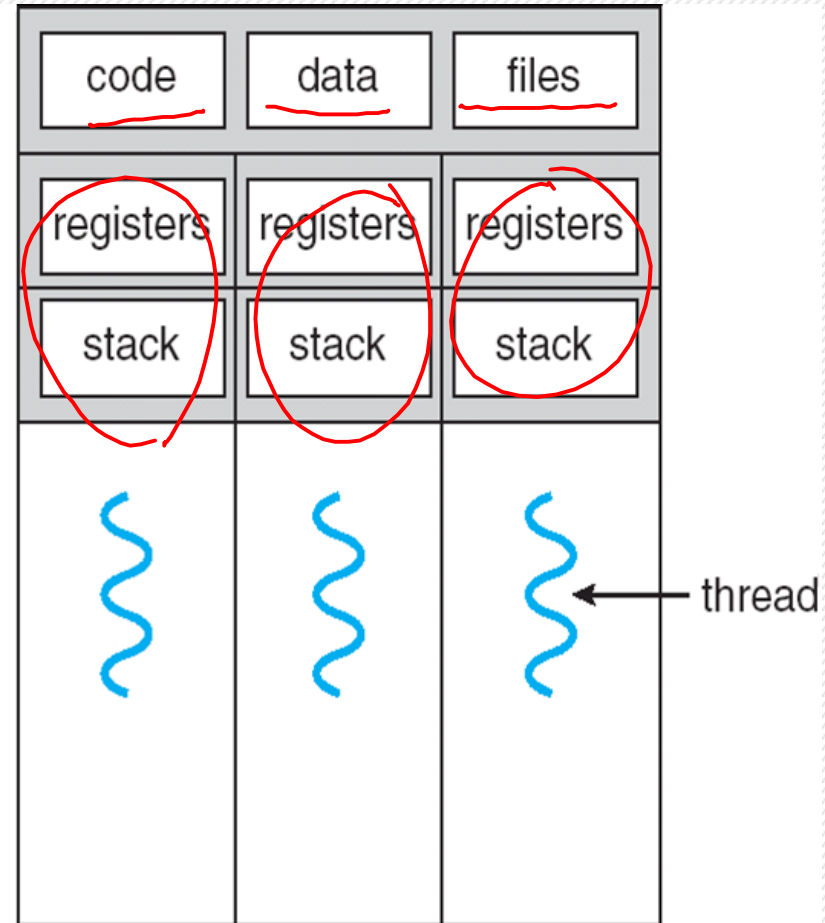
collectively know as a *task*. 总体作为一个任务

- A traditional or heavyweight process is equal to a task with one thread
传统的或重型进程等价于只有一个线程的任务

Single and Multithreaded Processes



single-threaded process



multithreaded process

进程和线程的比较

- 并行性：在引入线程的OS中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间亦可并发执行，因而使OS具有更好的并行性，从而能更有效地使用系统资源和提高系统吞吐量。
- 拥有资源：进程是拥有资源的独立单位
- 系统开销：在创建或撤消进程时，系统都要为之分配或回收资源，如内存空间、I/O设备等。因此，OS所付出的开销将明显地大于在创建或撤消线程时的开销。

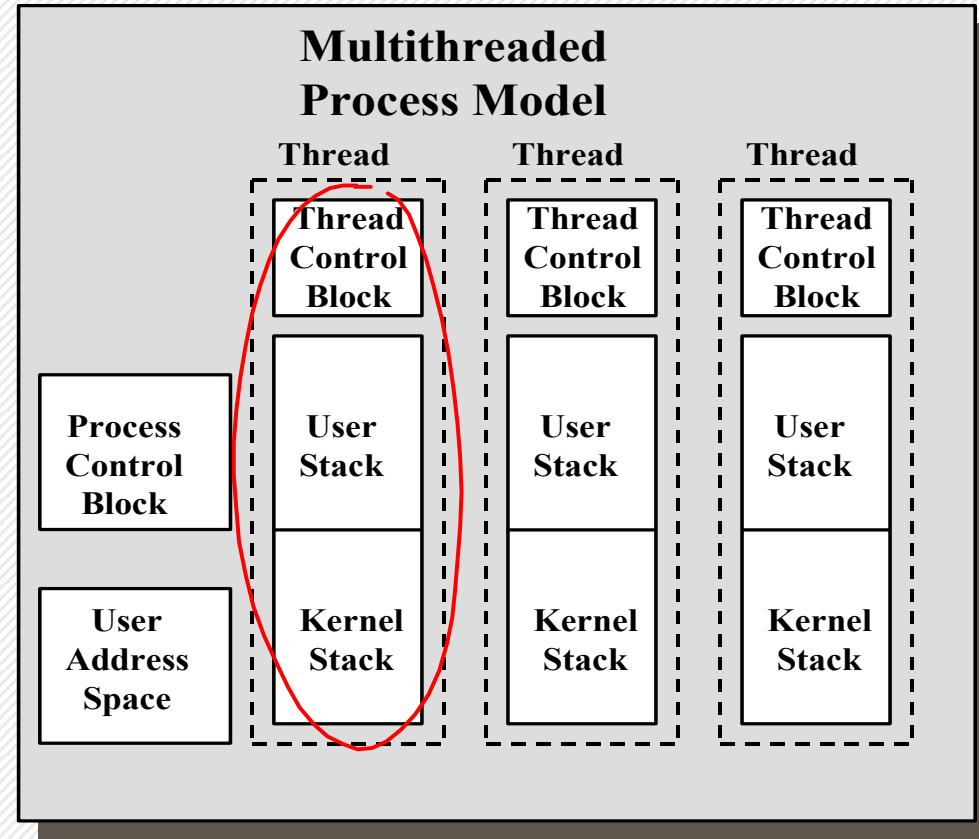
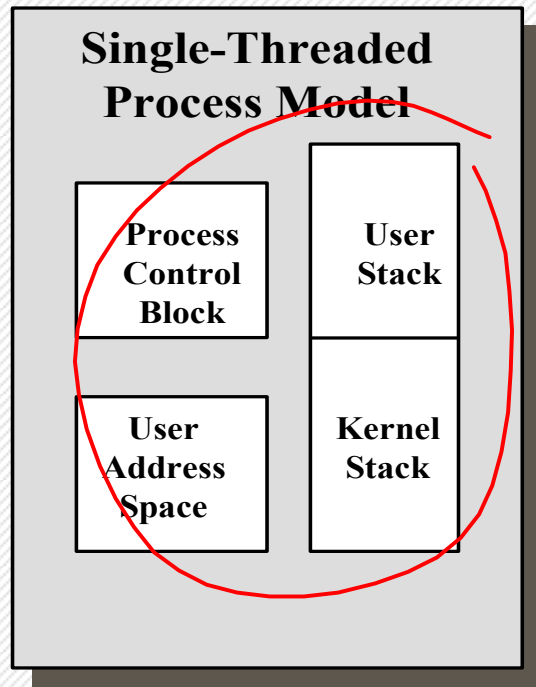
A comparison between a process and a thread

进程和线程的比较

- 地址空间和其他资源（如打开文件）：进程间相互独立，同一进程的各线程间共享 - - 某进程内的线程在其他进程不可见
- 通信：进程间通信IPC，线程间可以直接读写进程数据段（如全局变量）来进行通信 - - 需要进程同步和互斥手段的辅助，以保证数据的一致性
- 调度：线程上下文切换比进程上下文切换要快得多；

A comparison between a process and a thread

进程和线程的比较



线程切换和进程切换

Benefits

益处

Responsiveness

响应度高：一个多线程的应用在执行中，即使其中的某个线程阻塞，其他的线程还可继续执行，从而提高响应速度

Economy

经济性：创建和切换线程的开销要低于进程。比如，Solaris中进程创建时间是线程创建的30倍，进程切换时间是线程切换的5倍。

Resource Sharing

资源共享：同一进程的多个线程共享该进程的内存等资源

Utilization of MP Architectures

MP体系结构的运用：多线程更适用于多处理机结构。