第4章 线程 ★★★★★

- 主要内容
  - 4.1 进程和线程 ★★★★★
  - 4.2 线程分类 ★★★★★
  - 4.3 多核和线程 ★★★★★
  - 4.4 Windows 8 线程 （略）
  - 4.5 Solaris 线程（略）
  - 4.6 Linux 的进程和线程管理（略）
  - 4.7 Android 的进程和线程管理（略）
  - 4.8 Mac OS X的GCD技术（略）

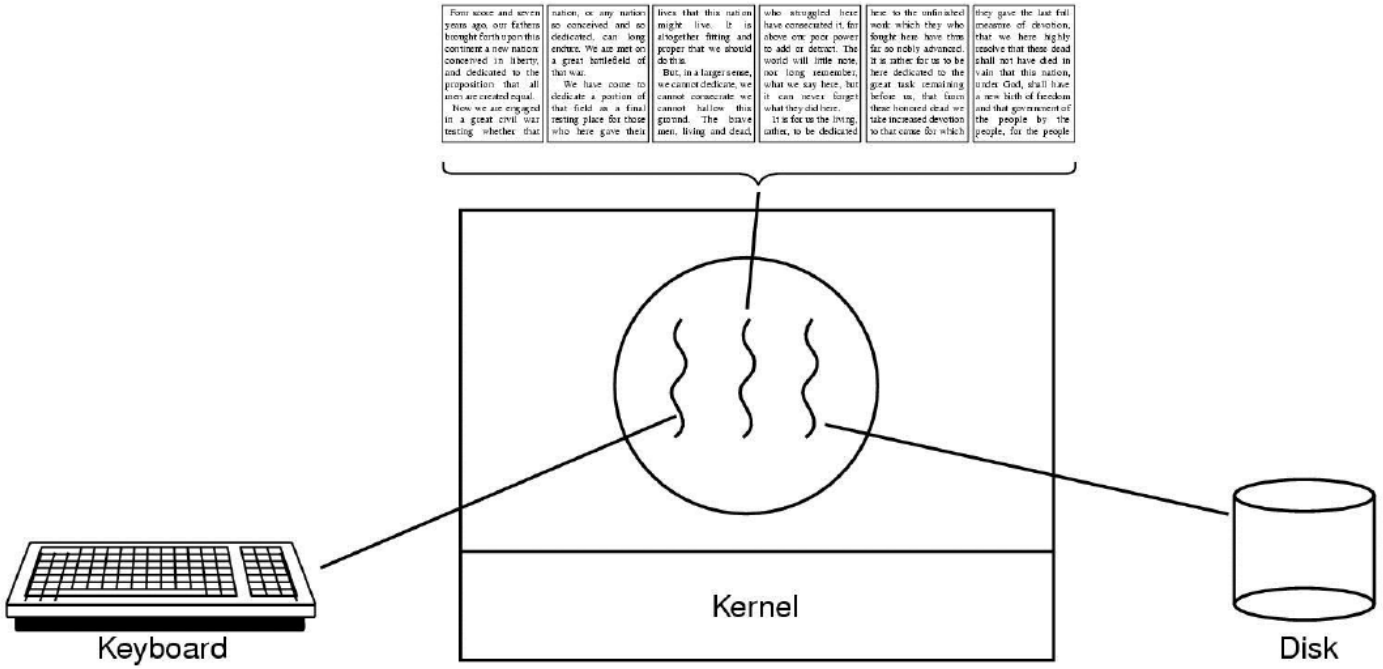
4.1 进程和线程 ★★★★★

- 为什么在进程中再派生线程？

三个理由

- 应用的需要
- 开销的考虑
- 性能的考虑

4.1 进程和线程 ★★★★★



有三个线程的字处理软件

选自Tanenbaum《现代操作系统》第3版

4.1 进程和线程 ★★★★★

- 典型的应用

  Web服务器

- 工作方式
  - 从客户端接收网页请求（http协议）
  - 从磁盘上检索相关网页，读入内存
  - 将网页返回给对应的客户端
- 如何提高服务器工作效率？

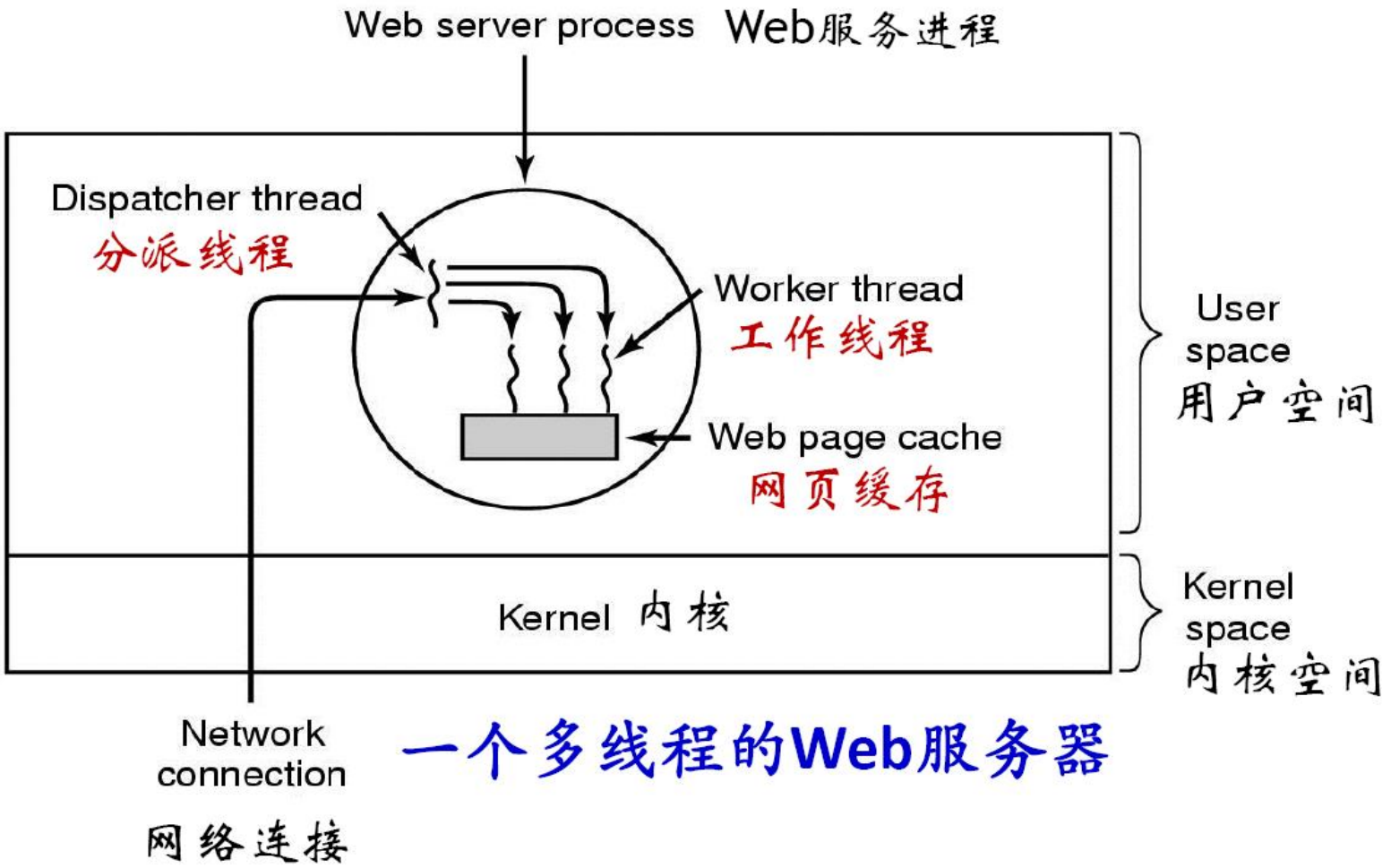
4.1 进程和线程 ★★★★★

- 典型的应用

  Web服务器

- 工作方式
  - 从客户端接收网页请求（http协议）
  - 从磁盘上检索相关网页，读入内存
  - 将网页返回给对应的客户端
- 如何提高服务器工作效率？

4.1 进程和线程 ★★★★★

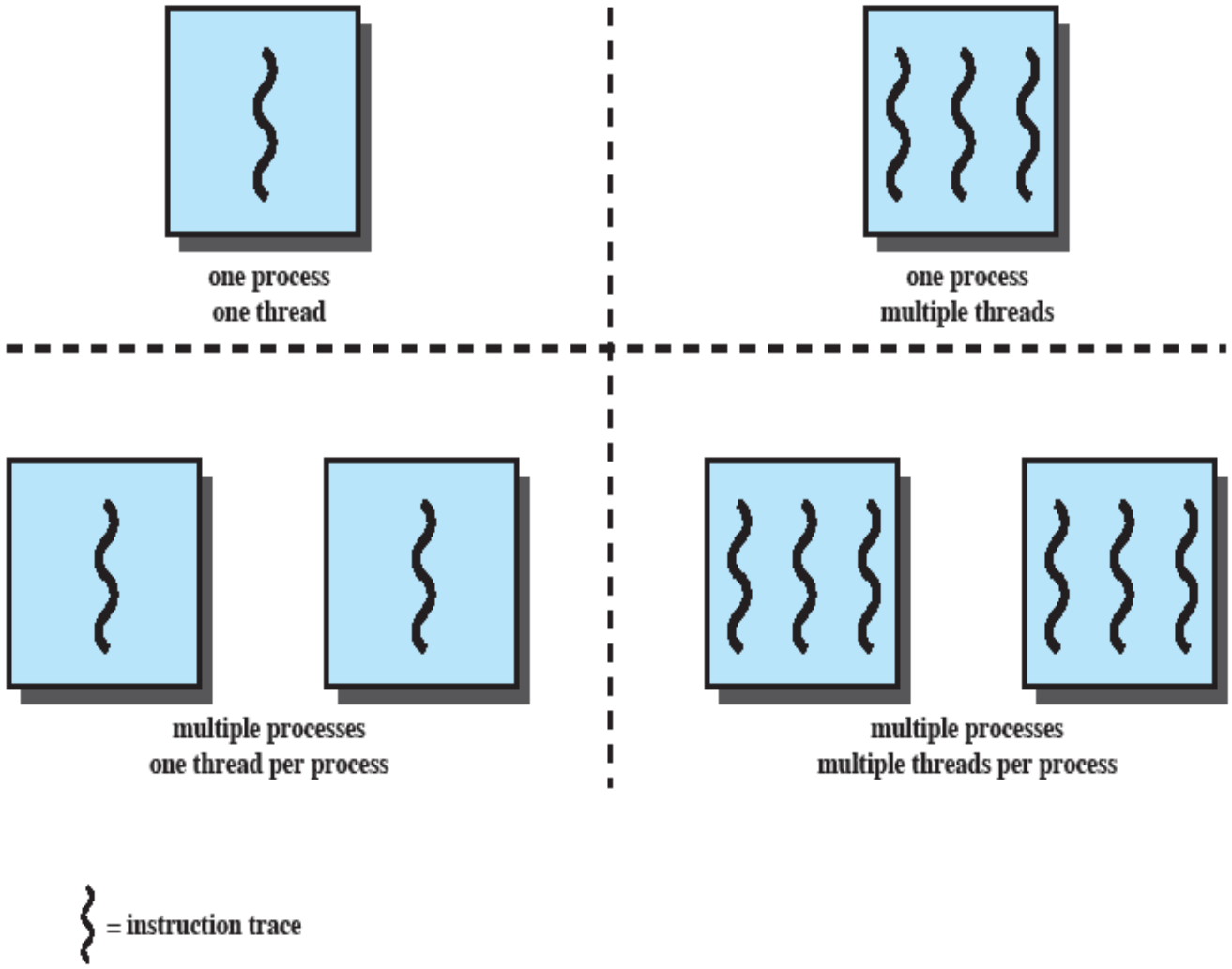


一个多线程的Web服务器

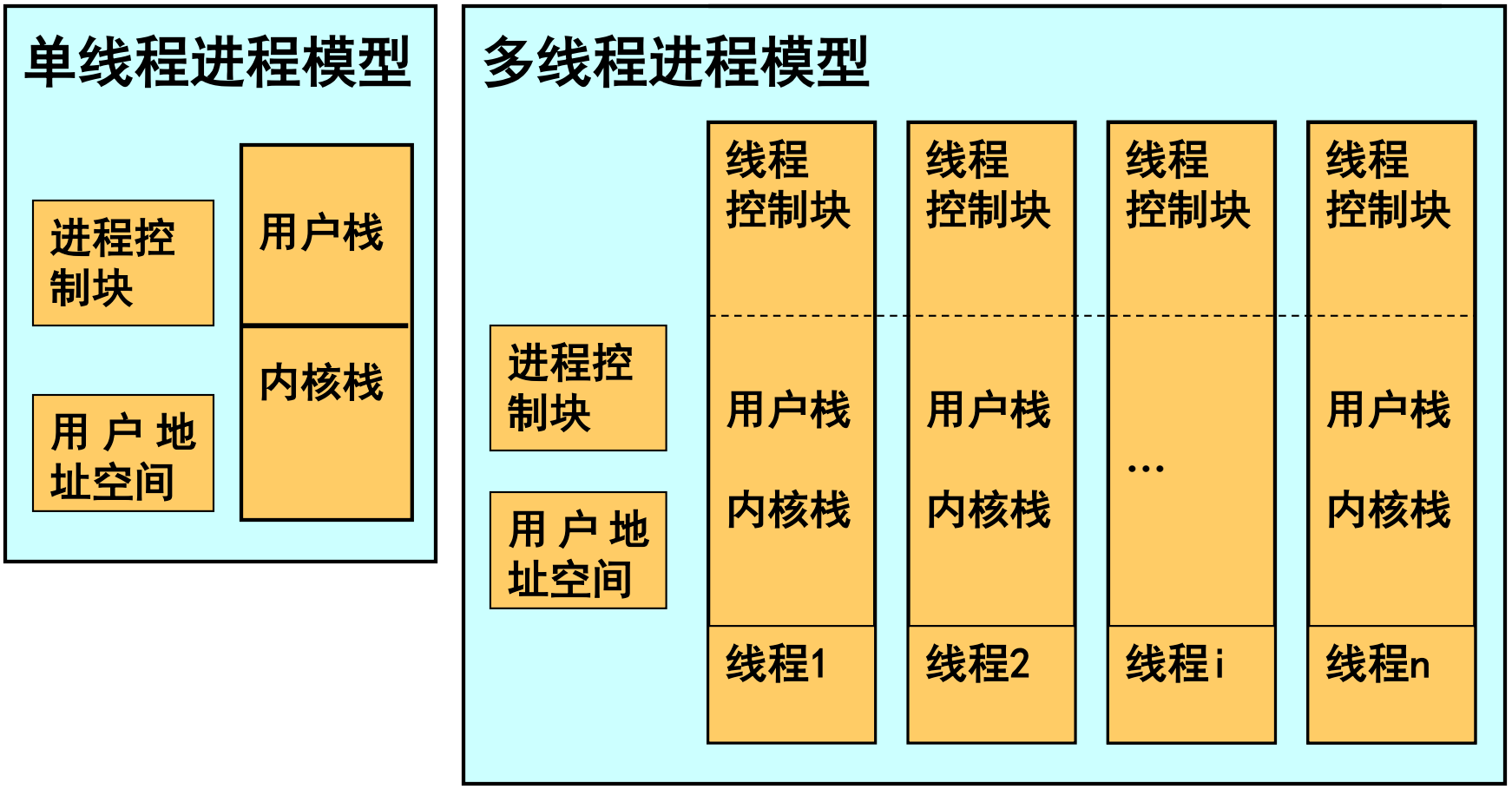
4.1 进程和线程 ★★★★★

- 进程包含两个特点：
  - 资源所有权：进程拥有对资源的控制权或所有权。
  - 调度/执行：进程是一个可被操作系统调度和分派的单位。
- 进程的两个特点是独立的，操作系统可以独立地对其进行处理。
  - 线程（轻量级进程）：分派（调度运行）的单位。
  - 进程（任务）：拥有资源所有权的单位。

4.1.1 多线程


one process
one thread
one process
multiple threads
multiple processes
one thread per process
multiple processes
multiple threads per process
= instruction trace

线程和进程的区别


单线程进程模型
进程控制块
用户栈
内核栈
用 户 地址空间
多线程进程模型
进程控制块
用 户 地址空间
线程控制块
用户栈
内核栈
线程1
线程控制块
用户栈
内核栈
线程2
线程控制块
…
线程i
线程控制块
用户栈
内核栈
线程n

线程的优点

- 在一个已有进程中创建一个新线程比创建一个全新进程所需的时间少很多。
- 终止一个线程比终止一个进程花费的时间少。
- 同一进程内线程间切换比进程间切换花费的时间少。
- 线程提高了不同的执行程序间通信的效率。

4.1.2 线程功能特性

1、线程状态

- 线程的关键状态
  - 就绪
  - 运行
  - 阻塞
- 挂起对线程没有意义。
- 一个线程阻塞是否会导致整个进程阻塞？

2、线程同步

- 同进程同步，后续章节会涉及。

4.1.2 线程功能特性

在操作系统中，并发是指一个时间段中有几个程序都处于已启动运行到运行完毕之间，且这几个程序都是在同一个处理机上运行，但任一个时刻点上只有一个程序在处理机上运行。

在关系数据库中，允许多个用户同时访问和更改共享数据的进程。SQL Server 使用锁定以允许多个用户同时访问和更改共享数据而彼此之间不发生冲突。[1]

Ref[1]: http://baike.baidu.com/link?url=-4GCrOhdI0smFLxtuJtPP8_aO4DvgZT0xHroJbwX_m-hW9PZJSb2-xLNZAsC-J4nXBGull1ySdSZHvPXZKYrWK

4.1.2 线程功能特性

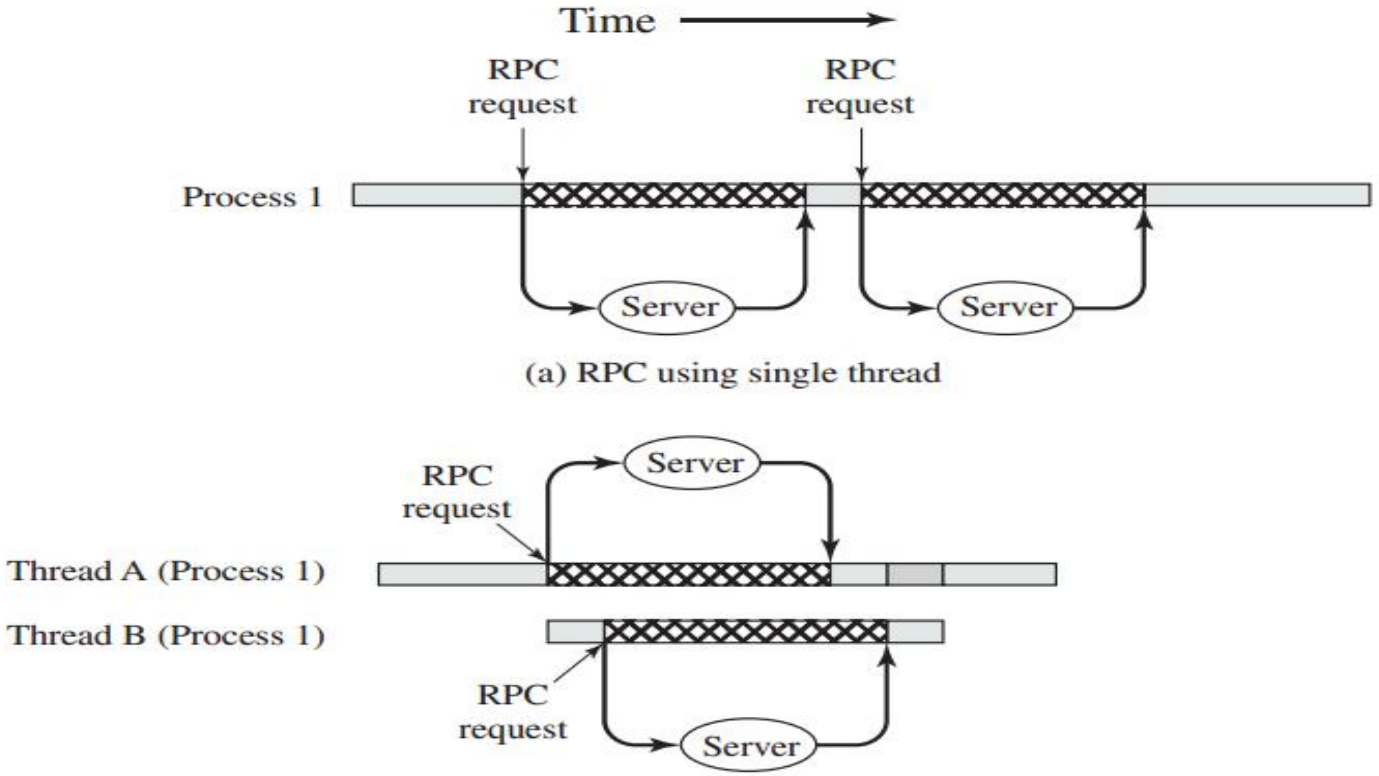


Figure 4.3 Remote Procedure Call (RPC) Using Threads

4.1.2 线程功能特性

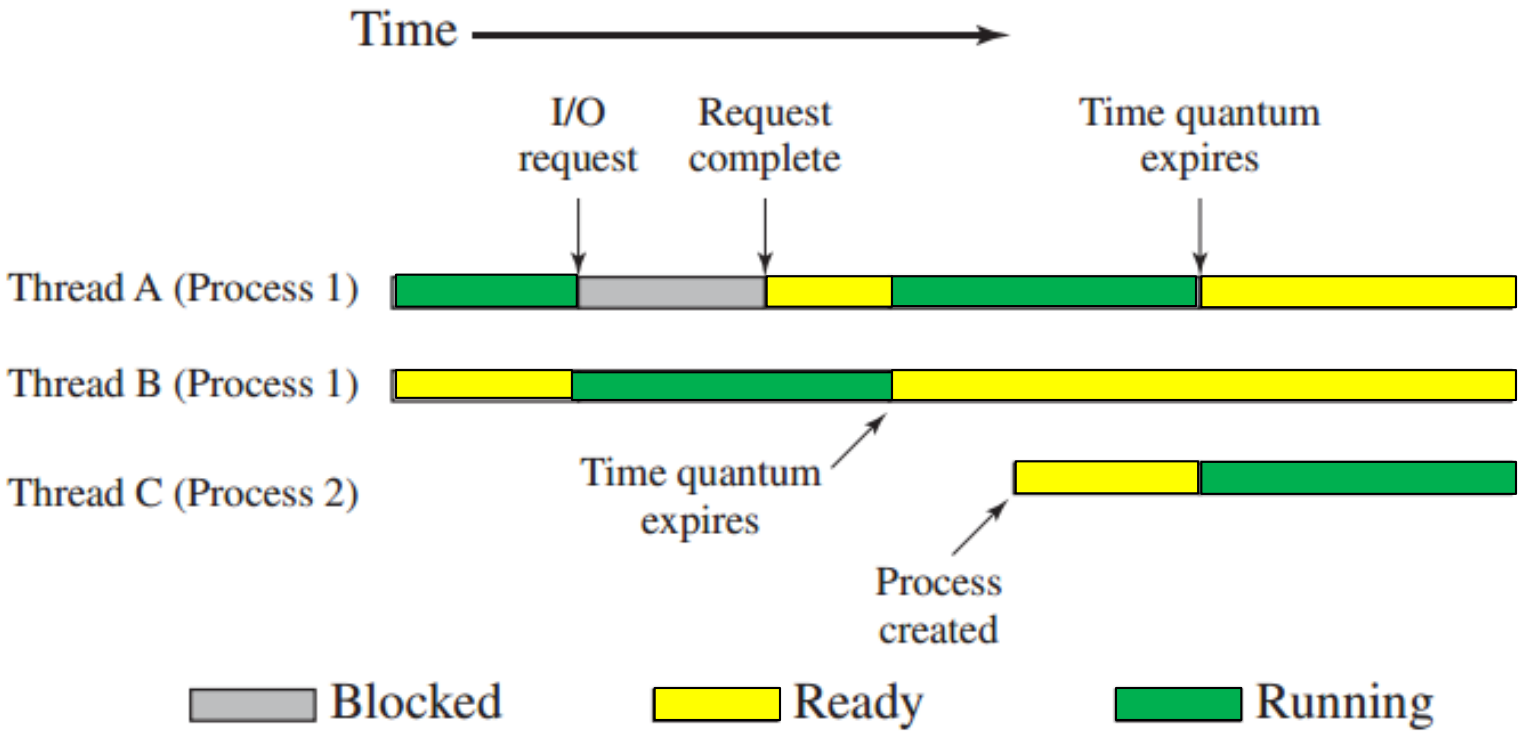


Figure 4.4 Multithreading Example on a Uniprocessor

提问

1.判断题：线程提高了不同执行程序间通信的效率，这是因为同一进程中的多个线程共享内存和文件，无须调用内核就可以相互通信。

A.正确　　B.错误

2.判断题：单CPU系统中，由于有线程的存在，同一进程中的多个线程可以同时运行。

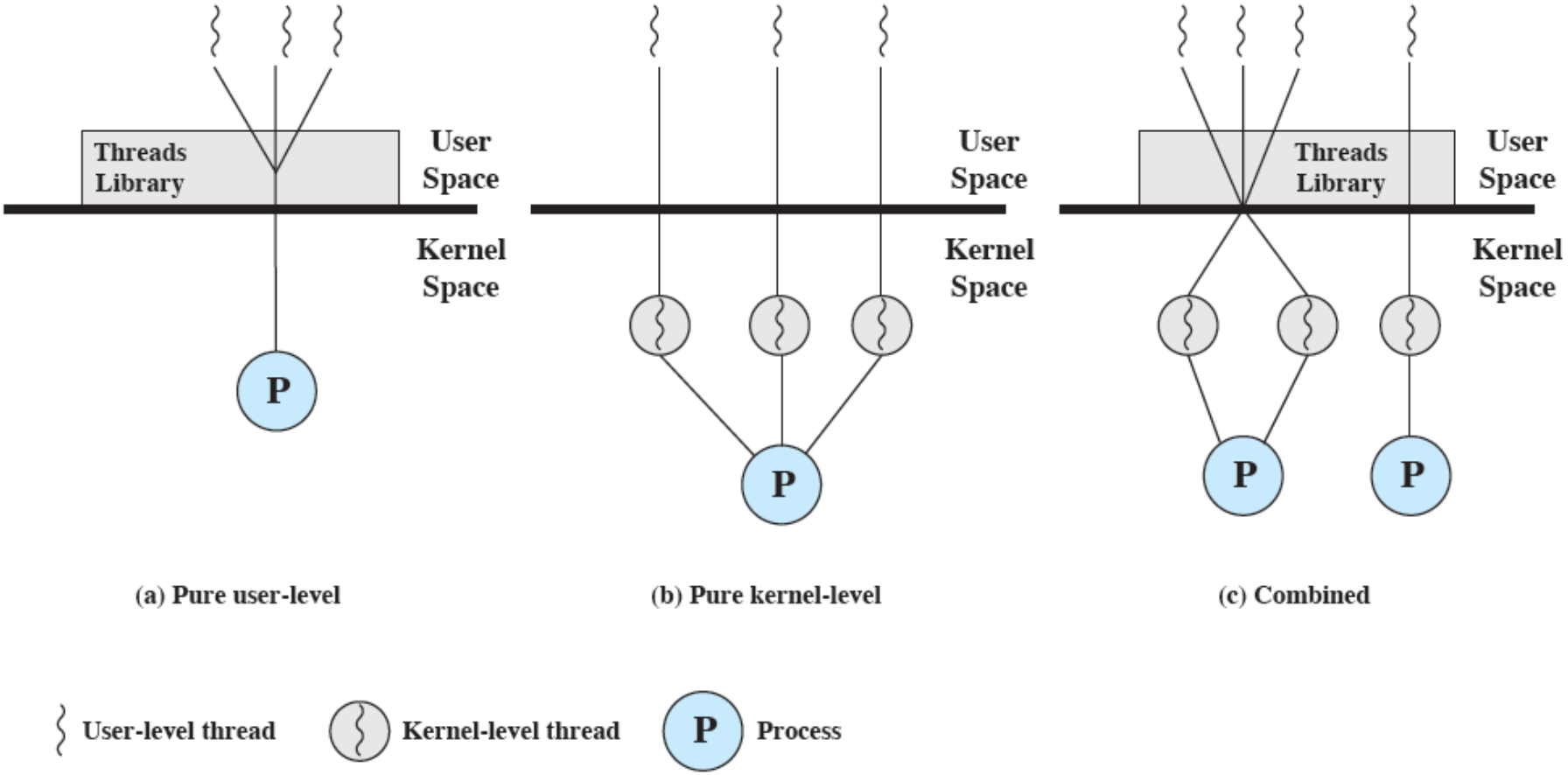
A.正确　　B.错误

3. 判断题：在多线程环境中，操作系统可以将进程的一部分线程挂起，留另一部分线程在内存中，以便充分利用处理器。

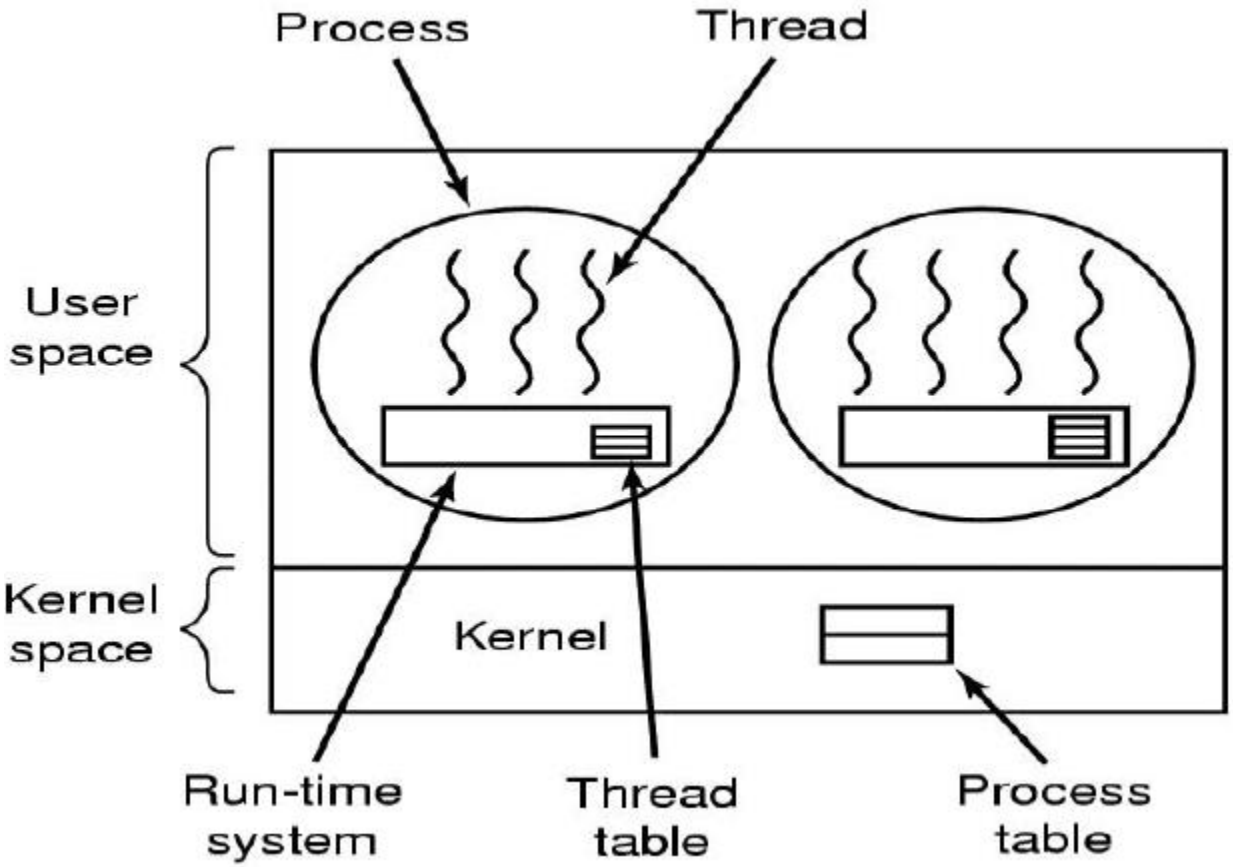
A.正确　　B.错误

4. 2 线程分类 ★★★★★

4.2.1 用户级和内核级线程 ★★★★★



1、用户级线程

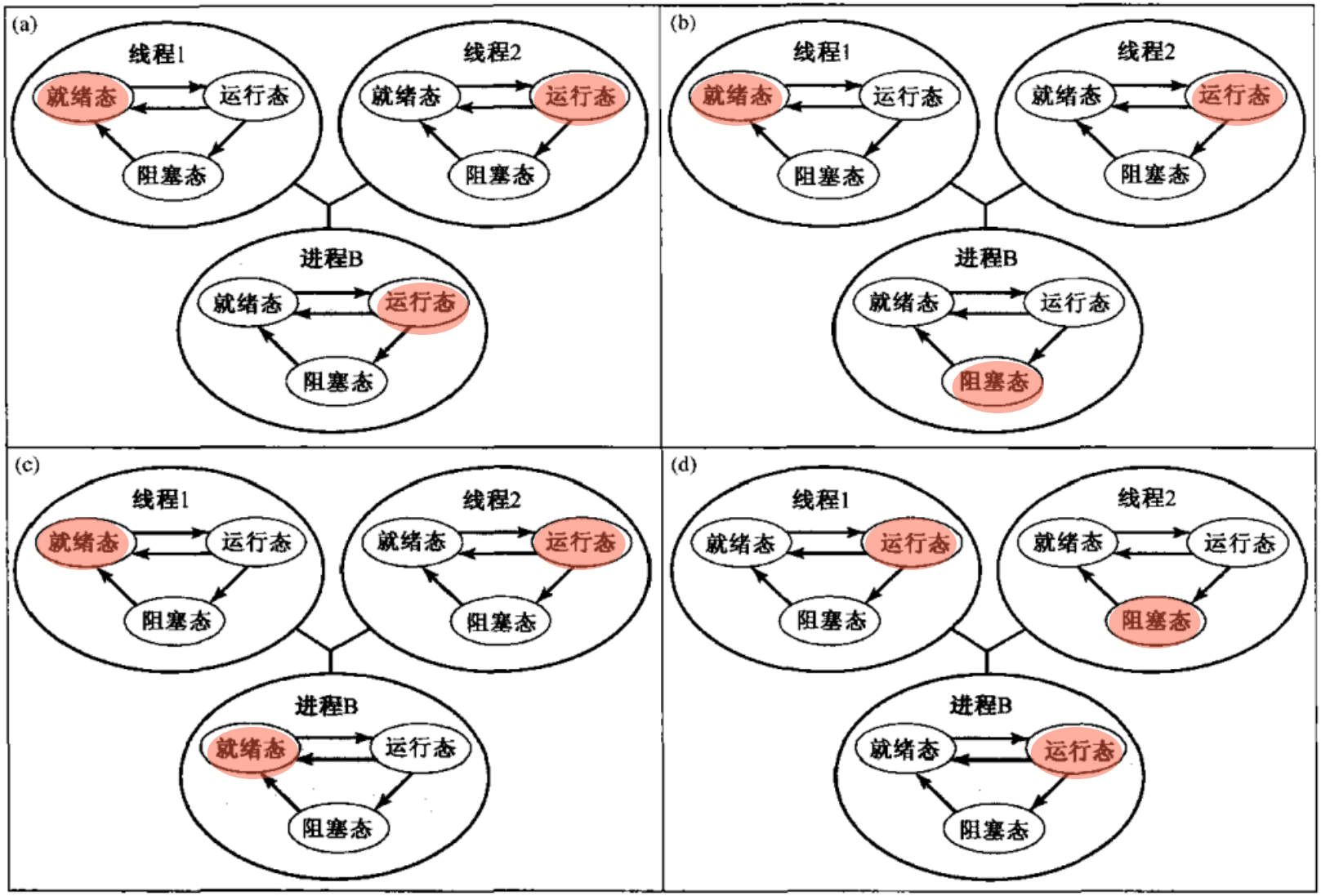

Process
Thread
User space
Kernel space
Kernel
Run-time system
Thread table
Process table

UNIX

1、用户级线程

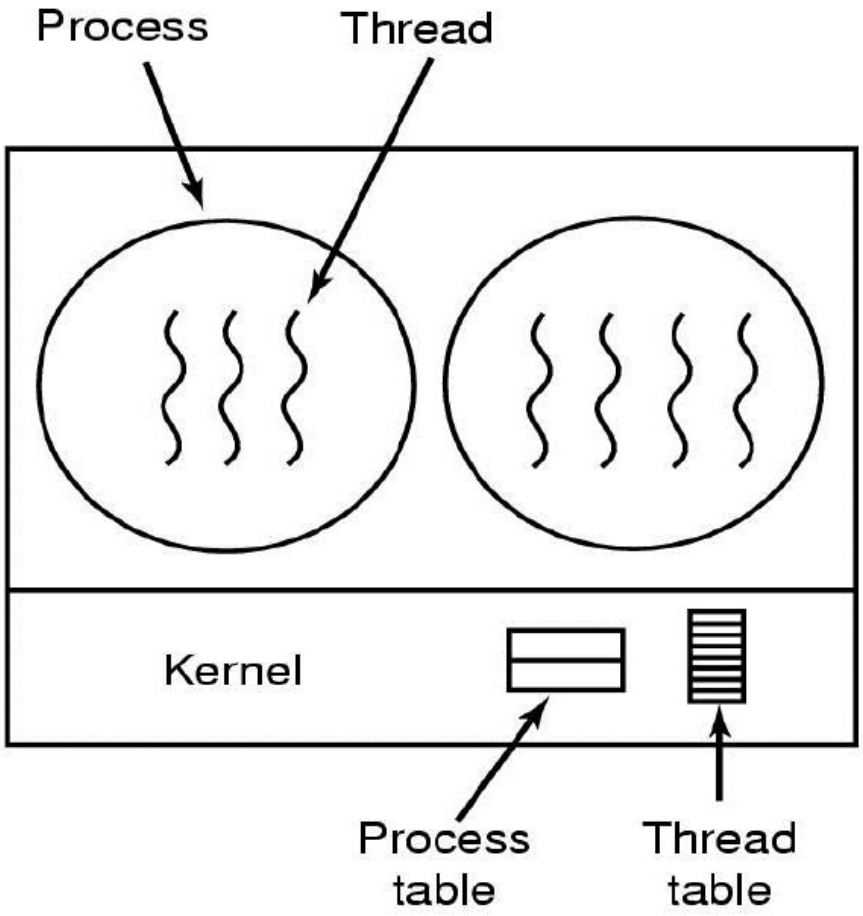
- 有关线程管理的所有工作都由应用程序完成，内核意识不到线程的存在。
- 优点：
  - 线程切换不需要内核态特权；
  - 调度可以是应用程序相关的；
  - 用户级线程可以在任何操作系统中运行，不需要对底层内核进行修改以支持用户级线程。
- 缺点：
  - 当执行一个系统调用时，会阻塞进程中所有线程；
  - 无法利用多处理技术。

4.2 线程执行模式



用户级线程状态和进程级状态之间关系的例子

2、内核级线程

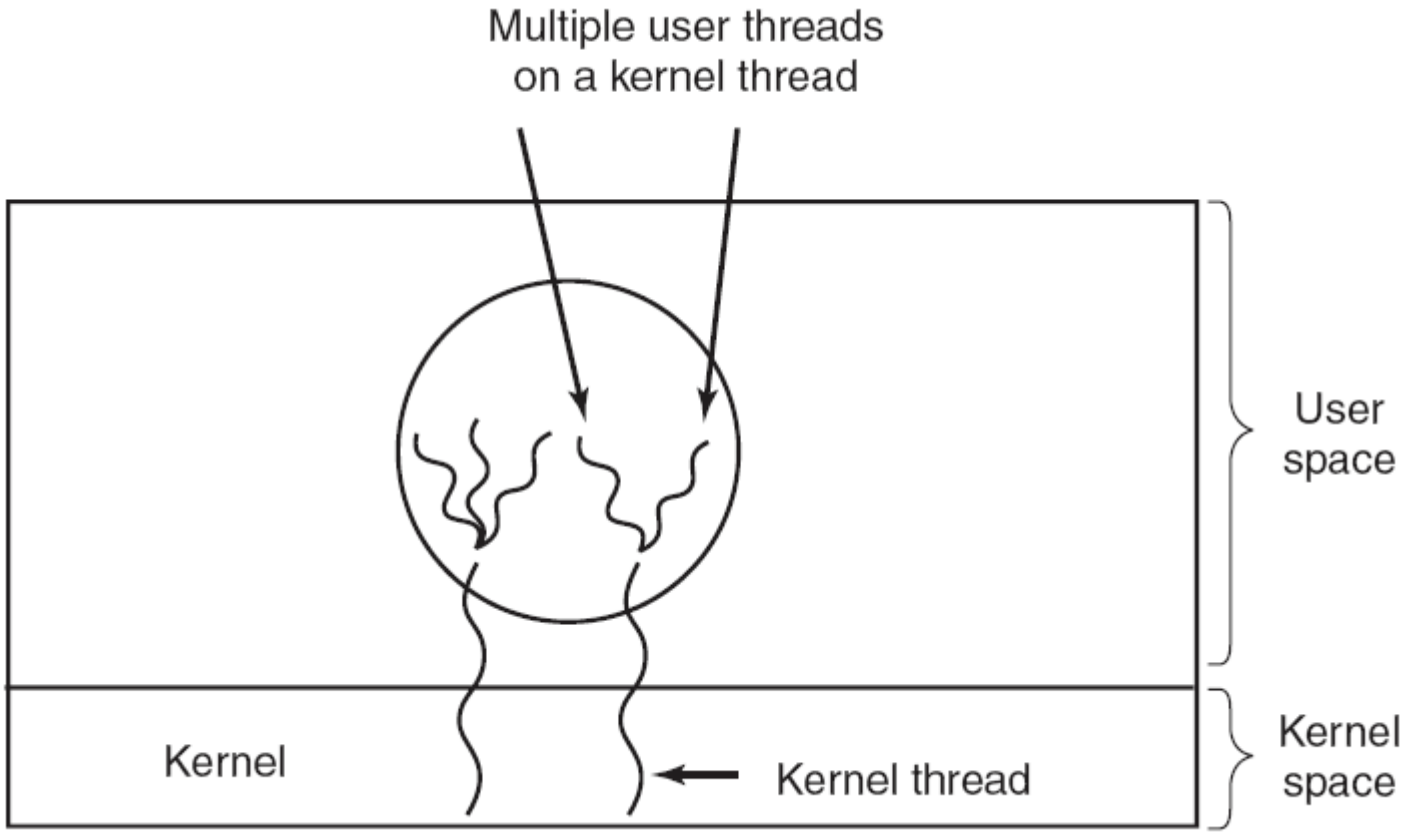


WINDOWS

2、内核级线程

- 有关线程管理的工作由内核完成，应用程序只有一个到内核线程设施的应用程序编程接口。
- 优点：
  - 内核可同时把同一进程中的多个线程调度到多个处理器中；
  - 若进程中的一个线程被阻塞，内核可以调度同一进程中的另一个线程；
  - 内核例程本身也可以使用多线程。
- 缺点：
  - 在把控制从一个线程传送到同一个进程内的另一个线程时，需要内核的状态切换。

3、混合方法



SOLARIS

3、混合方法

- 线程创建、调度和同步在应用程序中进行，一个应用程序中的多个用户级线程被映射到一些内核级线程上。
- 结合纯粹用户级线程方法和内核级线程方法的优点，并克服它们的缺点。

4.2.2 其他方案 ★★★★★

- 线程和进程间的关系

线程:进程	描述	实例系统
1:1	执行的每个线程是一个唯一的进程，有它自己的地址空间和资源	传统UNIX
M:1	一个进程定义了一个地址空间和动态资源所有权，可以在该进程中创建和执行多个线程	WindowsNT、Solaris
1:M	一个线程可以从一个进程环境迁移到另一个进程环境，允许线程可以很容易地在不同系统中移动	RS、Emerald
M:N	结合了M:1和1:M的属性	TRIX

EXAMPLE

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define NUMBER_OF_THREADS     10

void *print_hello_world(void *tid)
{
        /* This function prints the thread's identifier and then exits. */
        printf("Hello World. Greetings from thread %d\n", tid);
        pthread_exit(NULL);
}

int main(int argc, char *argv[])
{
        /* The main program creates 10 threads and then exits. */
        pthread_t threads[NUMBER_OF_THREADS];
        int status, i;

        for(i=0; i < NUMBER_OF_THREADS; i++) {
                printf("Main here. Creating thread %d\n", i);
                status = pthread_create(&threads[i], NULL, print_hello_world, (void *)i);

                if (status != 0) {
                        printf("Oops. pthread_create returned error code %d\n", status);
                        exit(-1);
                }
        }
        exit(NULL);
}
```

4.3 多核和线程 ★★★★★

- 使用多核系统支持多线程应用程序会带来一些性能和应用程序设计上的问题。
- 多核架构带来的潜在性能提升取决于一个应用程序有效使用可用并行资源的能力。

提问

4.判断题：在多线程环境中，如果某个线程A执行一个系统调用后被阻塞，那么与线程A同属一进程的其他线程也会被阻塞。

A.正确　　B.错误

5.判断题：在多线程环境中，线程的切换会涉及用户模式与内核模式的切换。

A.正确　　B.错误

6.判断题：在多线程环境中，线程的切换会涉及用户模式与内核模式的切换。

A.正确　　B.错误

4.4 Windows 线程

- Windows进程作为对象实现，使用两类与进程相关的对象：
  - 进程
  - 线程
- Windows线程有六种状态：
  - 就绪
  - 备用
  - 运行
  - 等待
  - 过渡
  - 终止

4.5 Solaris 线程

- Solaris采用三层线程架构
  - 用户级线程：通过线程库在进程地址空间中实现，对操作系统是不可见的，进程内一个用户创建的执行单元。
  - 轻量级进程：用户级线程和内核线程间的映射，每个轻量级进程支持一个或多个用户级线程，并映射到一个内核线程。
  - 内核线程：可调度和分派到系统处理器上运行的基本实体。
- 辅助操作系统的线程管理，并向应用程序提供清晰的接口。

4.6 Linux 线程

- Linux中进程和线程没有区别。
  - 当两个进程共享相同虚存时，可被当做是一个进程中的线程；
  - 没有为线程单独定义数据结构。

作业

- 复习题4. 5， 4. 6,