第15章　Windows进程和线程管理

本章主要内容

1. 进程和线程
2. 线程调度
3. 对称多处理机上的线程调度
4. 线程优先级提升
5. 进程同步

15.1 进程和线程

1. 进程的特点
   ① 是一个可执行程序，有代码和数据。
   ② 有一个独立地址空间。
   ③ 可有多个线程。
   ④ 进程之间不具有父子关系。
2. 线程是进程内的执行实体。没有线程，进程的程序无法执行。

15.1.1　进程对象

1．执行体进程块(EPROCESS)

- 对象管理器创建和删除进程对象。
- 执行体进程块：进程控制块。Win32子系统提供检索和改变进程属性的系统服务。

在系统内部，一个进程的KPROCESS 对象的地址和EPROCESS对象的地址相同。

```
struct EPROCESS{
    KPROCESS  Pcb;
    PHANDLE_TABLE  ObjectTable;
    HARDWARE_PTE  PageDirectoryPte;
    LIST_ENTRY  ThreadListHead;
    MM_AVL_TABLE  VadRoot;
    ……
}
```

进程对象的基本属性

对象类型

进程

对象体属性

内核进程块（KPROCESS）
进程ID
访问令牌
配额限制
内存管理信息
句柄表
页目录表
异常/调试程序端口
进程环境块
映像文件名和地址
下一个进程块

a. 内核进程块（KPROCESS）

它是调度程序对象。包括：基本优先级、默认时间片、进程的转锁、进程所在处理机簇、进程状态、进程中的线程的内核总时间和用户总时间、进程页目录指针、属于该进程的所有线程的内核线程块队列指针等。

进程页目录的物理地址被保存在KPROCESS块中。

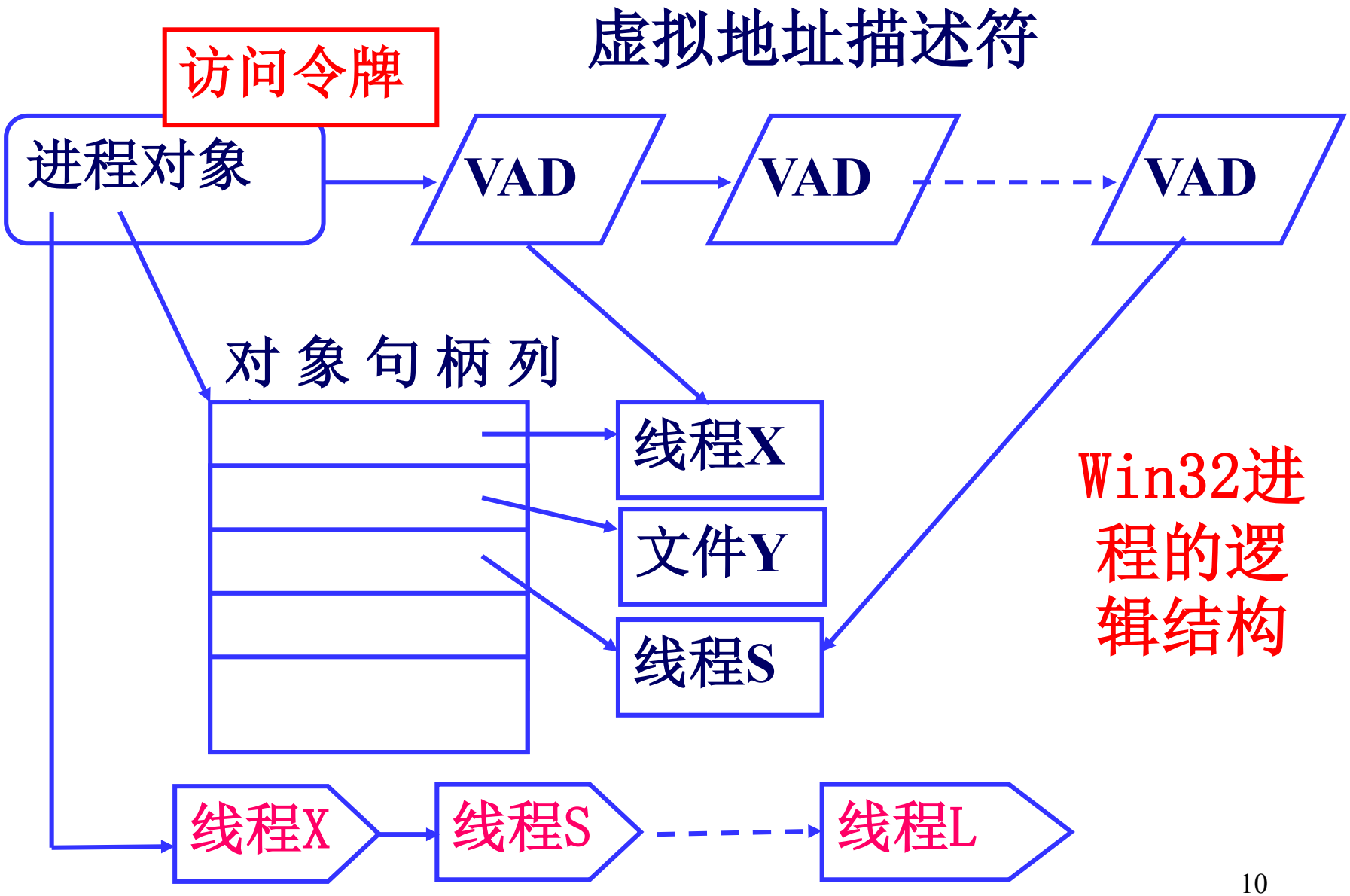
b.访问令牌：用户登录时由系统直接连到进程的安全认证。其内容：用户名及安全标识。

c.存储器管理信息：记录进程使用的一组虚地址空间的域和工作集信息，用一系列虚拟地址描述符VAD和指向工作集列表的指针描述。

d.对象句柄表：记录进程创建和打开的所有对象的列表。

e. 进程环境块PEB。位于进程私有地址空间，包含一些需要由用户代码来修改的信息。如，进程映像信息（基地址、版本号、模块列表），供线程使用的进程堆的数量和大小，亲合掩码等。

f. 异常/调试程序端口：是进程的线程出现异常或进行调试时，进程管理器发送消息的内部通信通道。


虚拟地址描述符
访问令牌
进程对象
VAD
VAD
VAD
对 象 句 柄 列
线程X
文件Y
线程S
Win32进程的逻辑结构
线程X
线程S
线程L

2．进程对象的服务

Win32子系统的主要进程服务：创建和打开进程、进程退出、进程终止、获得和设置进程的各种信息。

- CreateProcess创建新进程及其主线程，执行指定的程序。
- ExitProcess 和 TerminateProcess 终止调用者进程内的所有线程。

CreateProcess主要流程

① 打开将在进程中执行的映像文件(.exe)，创建一个区域对象，建立映像与主存之间的映射关系。

② 创建执行体进程对象，包括申请并初始化执行体进程控制块，创建并初始化进程地址空间、内核进程块和进程环境块等。

③ 创建一个主线程。

④ 通知Win32子系统，对新进程和线程进行一系列初始化。

⑤ 完成地址空间的初始化，开始执行程序。

ExitProcess和TerminateProcess

- 终止调用者进程内的所有线程。
  - a. 当进程中的一个线程调用ExitProcess时，将终止进程和进程中所有线程的执行，是正常完成采用的退出方式。
  - b. TerminateProcess终止指定的进程和它的所有线程。可终止自己，可终止其他进程。通常用于异常情况下使用进程句柄终止进程。

15.1.2　线程对象

1．执行体线程块（ETHREAD）

线程是内核支持的线程，是处理器调度的对象。每个线程都有一个执行体线程块。

在系统内部，一个进程的KTHREAD 对象的地址和ETHREAD对象的地址相同。

```
struct ETHREAD{
   KTHREAD  Tcb;
   PVOID  StardAddress; 线程的启动地址
   ……
}
```

线程对象的基本属性

对象类型

线程

对象体属性

核心线程块（KTHREAD）
进程ID
创建和退出时间信息
LPC端口信息
启动地址
访问令牌和线程类别
I/O信息
指向所在进程的EPROCESS块

内核线程块（KTHREAD）

- 其中的许多域与线程调度机制有关。
  1. 核心栈的栈指针和大小
  2. 指向系统服务表的指针
  3. 与调度和同步有关的信息（优先级、时间片、处理机簇、当前的状态、总的时间、等待信息、等待块列表等）
  4. 与本进程有关的APC列表
  5. 线程环境块。在进程私有地址空间。

2. 线程对象的服务

Win32子系统的线程服务主要有：

① CreateThread创建线程

② ExitThread线程退出

③ TerminateThread终止某个线程

④ SetThreadPriority 改变线程优先级

15.2 线程调度

- 核心支持线程，处理器调度的对象是线程。它是基于优先级的抢先式的多处理器调度系统。优先级相同时按时间片轮转。
- 通常线程可在任何一个可用处理器上运行，线程的亲合处理器集合允许用户线程选择偏好的处理器。

- 处理器调度对象是线程，进程仅作为资源对象和线程的运行环境的提供者。内核中完成线程调度功能的一些函数统称为内核调度器。

- 处理器调度是严格针对线程队列进行的，并不考虑被调度线程属于哪个进程。

  例如，进程P有5个可运行的线程，进程Q有2个可运行的线程，如果这7个线程的优先级相同，则每个线程将得到1/7的处理器时间。

进程和线程优先级

（1）进程优先级

系统支持四种优先级：空闲/普通/高/实时

a. 只有进程处于普通优先级时，才能升高或降低其优先级。
b. 空闲优先级是为那些在系统处于空闲状态时执行的线程使用的。如屏幕保护程序。
c. 高优先级只是在需要时才使用。Task manager 以高优先级运行。
d. 实时优先级主要用于核心态的系统线程。

进程优先级类别	CreateProcess标志	级别
空闲(Idle)	IDLE_PRIORITY_CLASS	4
普通(Normal)	NORMAL_PRIORITY_CLASS	7/9
高(High)	HIGH_PRIORITY_CLASS	13
实时(Real-Time)	REALTIME_PRIORITY_CLASS	24

（2） 线程优先级

- 一旦线程被创建，其优先级就是所属进程的优先级。
- 进程仅有基本优先级。线程有基本优先级和当前优先级。系统根据当前优先级调度线程。

线程的优先级被分成以下三部分：

① 16个实时线程优先级（16~31）

② 15个可变线程优先级（1~15）

③ 空闲优先级（0）

- 从不调整在实时范围(16-31)内的线程优先级
- 空闲优先级(0)仅用于系统的零页线程，实现对系统中的空闲物理页面进行清零的操作。

（3）线程的时间配额

- 相当于时间片的长短。
- 时间配额是一个称为配额单位（Quantum Unit）的整数值（6、12、18、36）。
- 每次时钟中断，时钟中断服务例程从线程的时间配额中减少一个固定值（3）。如果时间配额用完，系统将选择另一个线程进入运行状态。

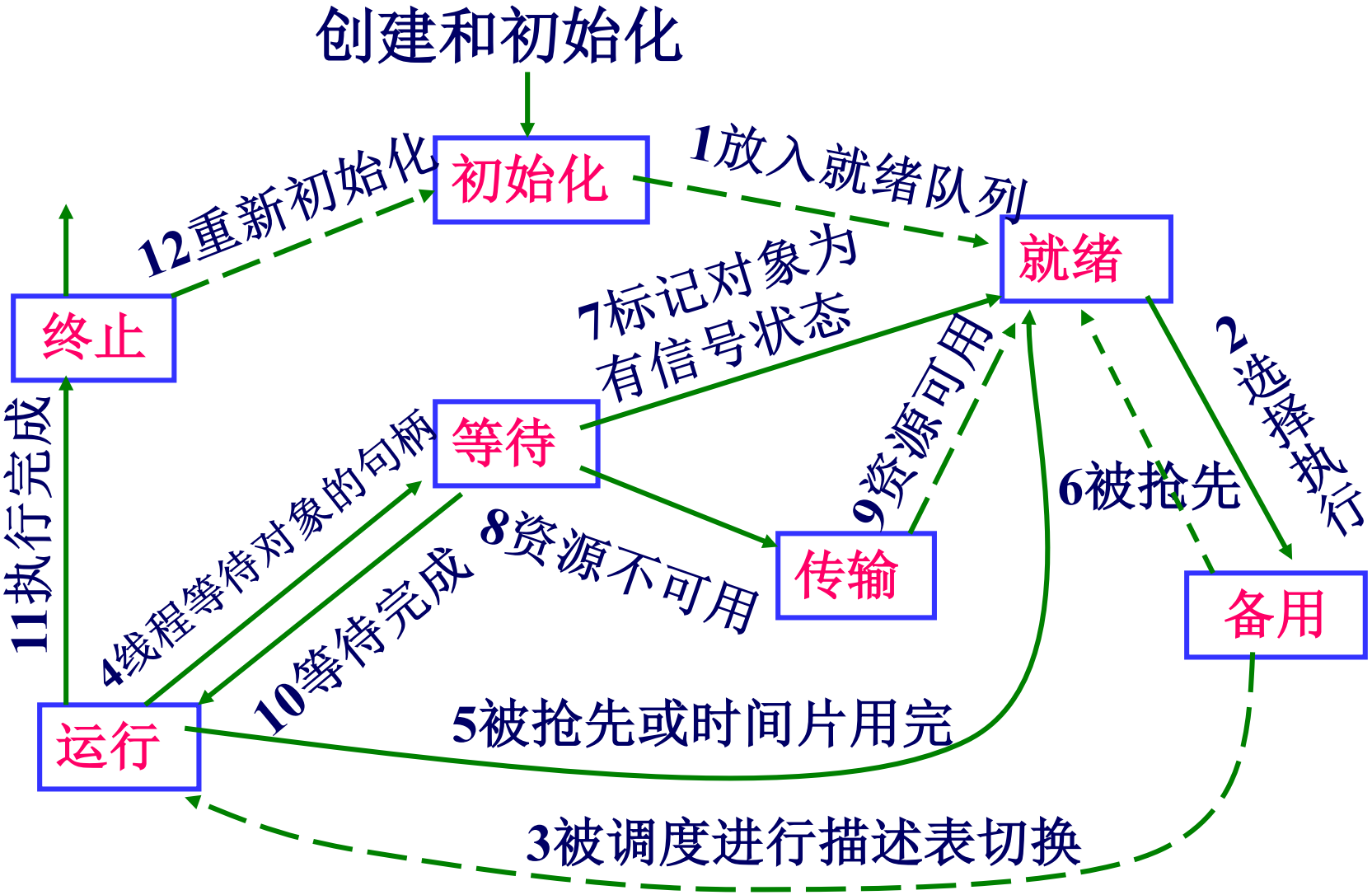
线程调度的时机

单处理机系统中的线程调度时机。

1. 主动切换。线程主动放弃处理器。
2. 抢先。被剥夺处理机。
3. 时间配额用完。
4. 运行结束。

线程的状态

1. 就绪状态(ready)
2. 备用状态(standby)。已选好处理机，正等待描述表切换，以便进入运行状态。
3. 运行状态(Running)
4. 等待状态(waiting)
5. 传输状态(transition)。核心栈被调到外存的就绪态。
6. 终止状态(terminated)
7. 初始化状态(Initialized)。正在创建过程中。


创建和初始化
初始化
1放入就绪队列
就绪
12重新初始化
终止
7标记对象为
有信号状态
2选择执行
等待
9资源可用
6被抢先
4线程等待对象的句柄
8资源不可用
传输
备用
11执行完成
10等待完成
运行
5被抢先或时间片用完
3被调度进行描述表切换

15.3 对称多处理机系统上的线程调度

15.3.1 几个与调度有关的概念：

1. 亲合关系（Affinity）。每个线程都有一个亲合掩码，描述该线程可在哪些处理机上运行。默认时，为所有可用处理机。
2. 线程的首选处理器和第二处理器

15.3.2 线程调度程序的数据结构

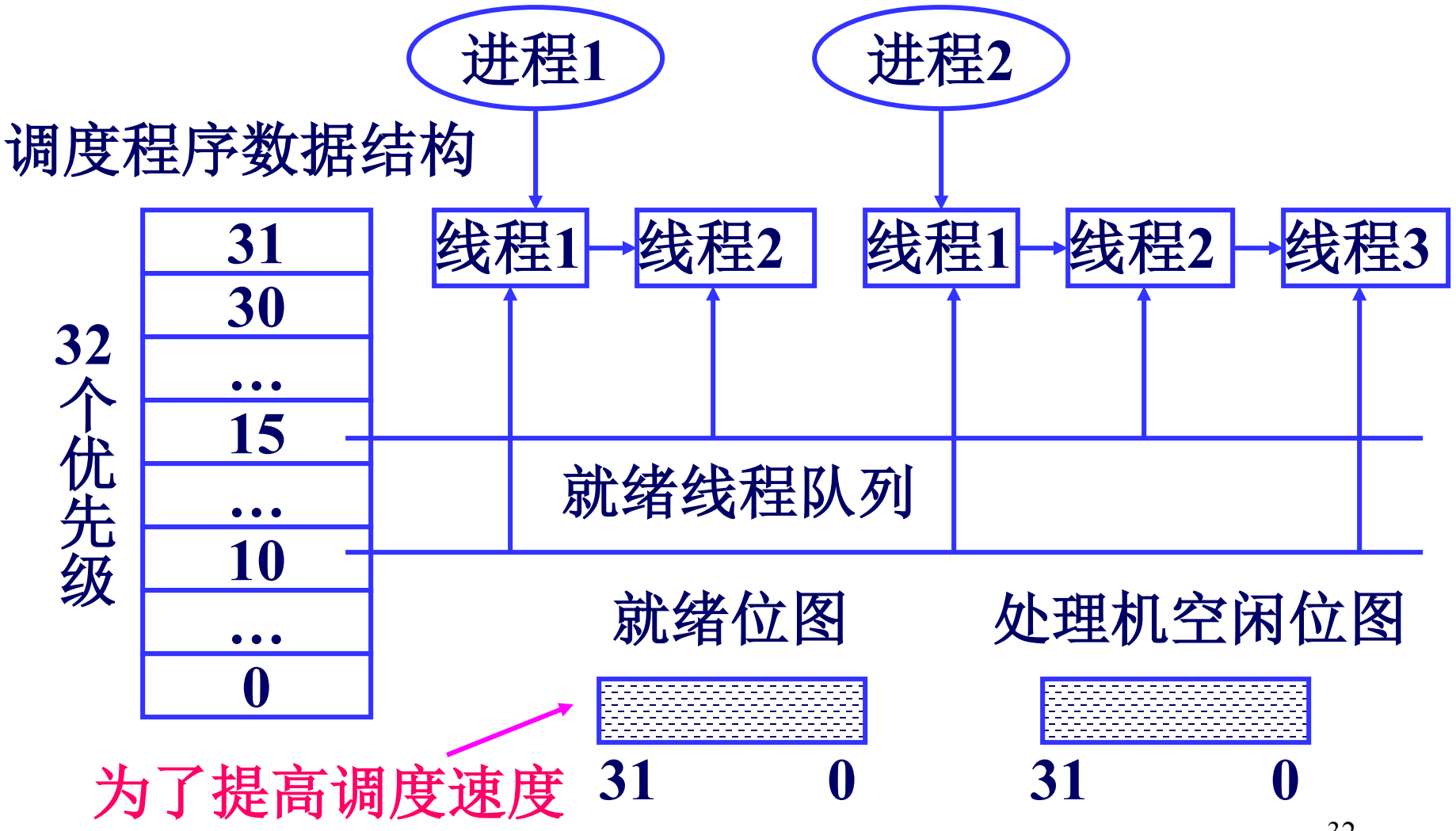
负责记录各线程的状态。

① 32个就绪队列。每个优先级对应一个。

② 32位掩码的就绪位图。每一位指示一个调度优先级的就绪队列中是否有线程等待运行。

③ 32位掩码的空闲位图。每一位指示一个处理机是否处于空闲状态。

调度程序的数据结构


进程1
进程2
调度程序数据结构
31
30
…
15
…
10
…
0
32个优先级
线程1
线程2
线程1
线程2
线程3
就绪线程队列
就绪位图
处理机空闲位图
31
0
31
0
为了提高调度速度

- 为了防止调度程序代码与线程在访问调度程序数据结构时发生冲突，线程调度出现在DPC/Dispatcher中断优先级。
- 在多处理机系统中，在修改调度程序数据结构之前，必须获得内核调度器自旋锁，实现多处理机互斥访问。

15.3.3 多处理机的线程调度算法

1. 有空闲处理机
2. 没有空闲处理机
3. 为特定的处理器选择线程
4. 优先级高的就绪线程可能不处于运行状态。在多处理机系统中，由于线程的亲合关系，系统并不总是选择优先级高的线程抢先优先级低的线程所占用的处理机。

15.4 线程优先级提升

- 系统会提升线程的优先级，以改善性能。

1. I/O操作完成后的线程。
2. 信号量或事件等待结束的线程。
3. 前台进程中的线程完成一个等待操作。
4. 由于窗口活动而唤醒图形用户接口线程。
5. 线程处于就绪状态超过一定时间，仍未能进入运行状态(处理器饥饿)。

15.5 进程同步

- 实现进程和线程之间互斥和同步的机制有：事件对象、互斥体对象、信号量对象以及相应的系统服务。
- 线程等待与这三种对象同步时，可使用两个系统服务来实现。
  - ✓WaitForSingleObject()
  - ✓WaitForMultipleObjects()
- 同步对象是线程可与之同步执行的对象

15.5.1 同步对象

1. 事件(Event)对象：是同步对象中最简单的形式。它有有信号和无信号两个状态。它相当于一个“触发器”，用于通知线程某个事件是否出现.
2. 互斥体对象(Mutex)：用来控制共享资源的互斥访问。
3. 信号量对象：就是资源信号量，初始值可在0到指定最大值之间设置，用于限制并发访问的线程数。

15.5.2 同步对象的应用示例

1. 互斥量对象的示例

两个线程通过共享一个计数器对各自获得的系统时间进行记录。因此，对计数器的操作，两者必须互斥。

2. 综合示例

- 一个线程负责读文件数据至共享缓冲区。另两个线程负责处理文件数据，分别用于统计文件中的字数和行数。
- 设两个事件对象：一个是人工重置事件，用于通知两个处理线程；另一个是自动重置事件，用于通知读文件数据线程。
- 再设一个互斥信号量，互斥访问共享缓冲区。