嵌入式系统设计与应用

第六章 进程和操作系统（3）

西安交通大学电信学院
孙宏滨

Hongbin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

Metrics

- 我们该如何评估调度策略？
  - 能满足所有截止时限
  - **CPU**利用率**---CPU**执行有用工作所占的时间比例
  - 调度开销**---**做调度决策所需的时间

Hongbin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

3 基于优先级的调度

- 分配优先级主要有两种方法：
  - 静态优先级：在整个执行过程中优先级始终不变
  - 动态优先级：在执行过程中优先级发生变化

Hongqin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

单调速率调度

- 单调速率调度（**Rate-Monotonic Scheduling, RMS**）：首先为实时操作系统开发的调度策略之一，直至现在仍然被广泛使用。
- **RMS**属于静态调度策略。事实证明，固定优先级的做法在许多情况下都足以有效地调度进程。
- **RMS**的理论基础是单调速率分析（**Rate Monotonic Analysis, RMA**）。

Honolulu Sangiaotong University
Xi'an Jiaotong University
Internal Teaching Use Only

RMA模型

- 所有进程都周期性地运行在一个CPU上
- 上下文切换的时间可以忽略
- 进程间的数据不相关
- 进程的执行时间是恒定的
- 所有进程的截止时限都在其周期的结束点上
- 优先级最高的就绪进程一定会被选择执行

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

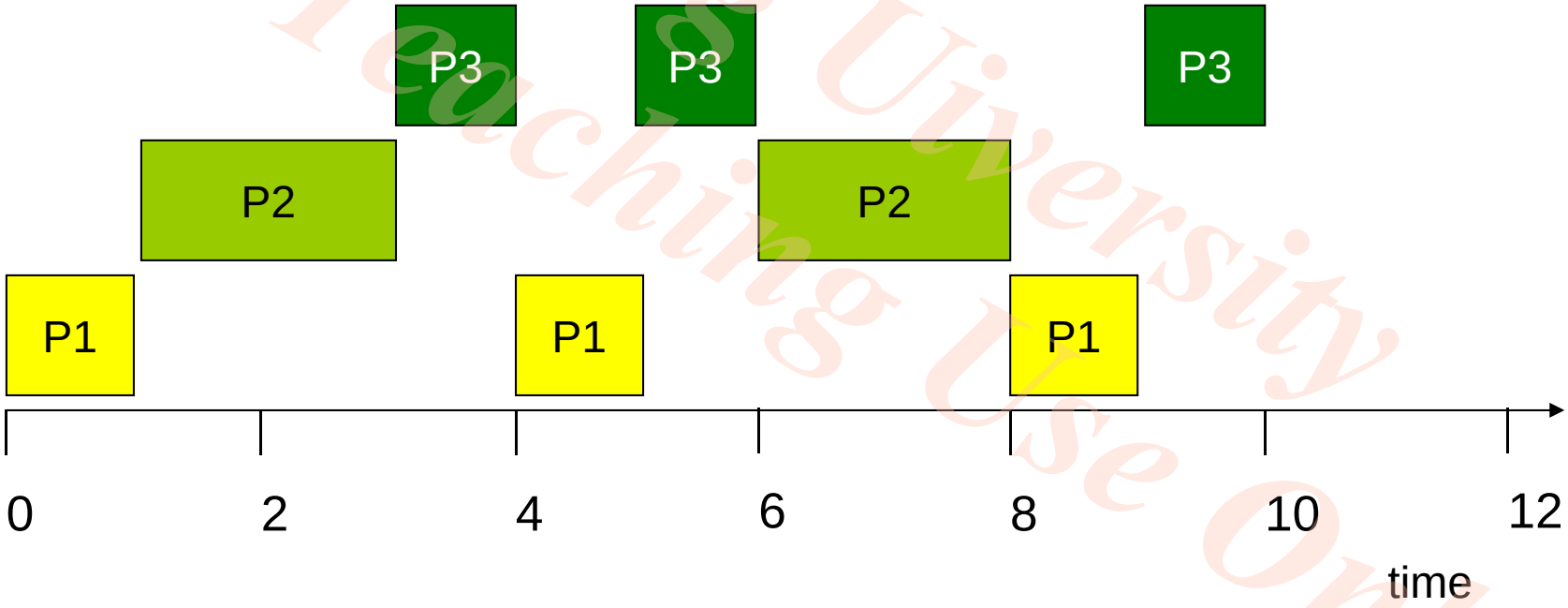
RMA的主要结论

- 相对简单的调度策略在某些特定条件下是最优的。
- 根据周期顺序进行优先级指定，周期最短的进程被指定为最高优先级。
- 这种固定优先级的调度策略对静态优先级进程来说是最适合的。在保证所有进程的截止时限的同时，还能获得最高的**CPU**利用率。

Hongjin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

单调速率调度示例1

进程	执行时间	周期
P1	1	4
P2	2	6
P3	3	12



Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

单调速率调度示例2

进程	执行时间	周期
P1	2	4
P2	3	6
P3	3	12

没有可行的优先级指定方法能保证调度。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

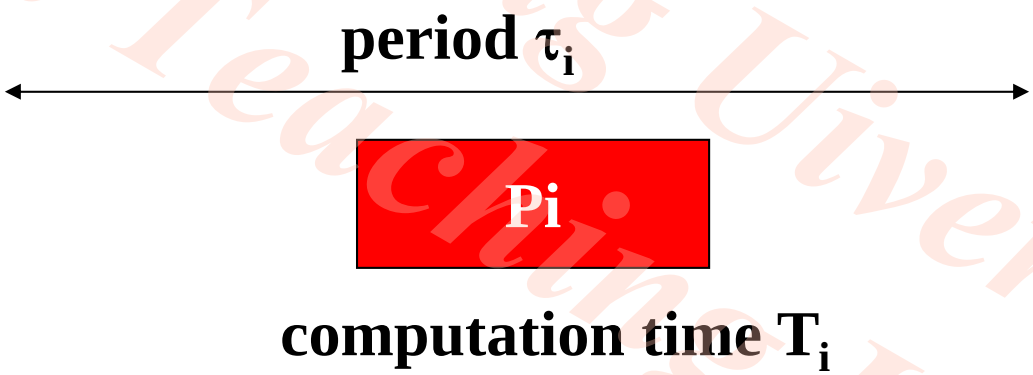
临界时刻分析

- **Liu**和**Layland**应用临界时刻分析证明了**RMA**优先级指定方法是最优的。
- 响应时间**Response time**：进程结束的时间.
- 临界时刻**Critical instant**：任务执行期间具有最大响应时间的时刻.
- 任意进程**P**的临界时刻出现在**P**已经就绪并且所有更高优先级的进程均已就绪时。
  - 如果我们让一个较高优先级进程进入等待状态，那么**P**的响应时间只会缩短。

Hongkin Sun Xi'an Jiaotong University Internal Teaching Use Only

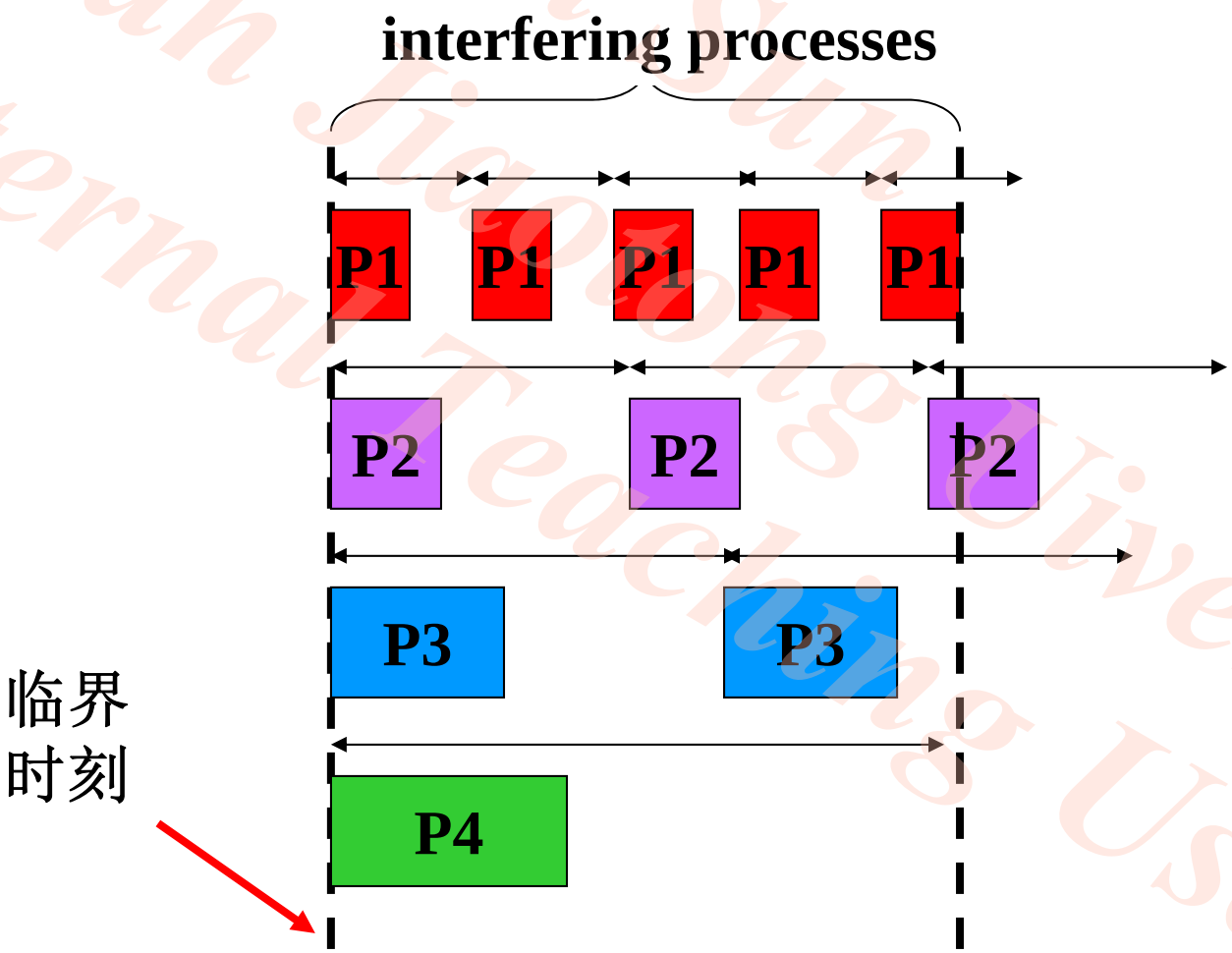
进程的参数

- T_i 表示进程i的计算时间；
- τ_i 表示进程i的周期。



Hongbin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

临界时刻的定义



临界时刻定义为执行期间任务具有最大相应时间的那一刻。

在RMA模型中，任意进程P的临界时刻出现在它已经就绪并且所有更高优先级的进程均已就绪时。

Hongwin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

临界时刻分析证明（1）

- 在任意大小的进程组中周期越短的进程应该给予越高的优先级。
- **RMS**总能够提供可行的调度，只要这样的调度存在。
- **RMS**调度是最优的静态优先级调度。

Hongshin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

临界时刻分析证明（2）

- 在RMS框架中，n个任务组成的集合的CPU利用率：

$$U = \sum_{i=1}^{n} \frac{T_i}{\tau_i}.$$

- 对于一个双任务集合，CPU的利用率*U的*最小上界$2X(2^{1/2}-1)\approx 0.83$。空闲17%的时间。
- 对于m个具有固定优先级的任务，CPU利用率的最小上界 $U = m(2^{1/m} - 1)$
  - 利用率达到100%发生在整数倍周期关系时。

注意：是充分非必要条件！

Hongjin Sun Xiangjiaotong University Internal Teaching Use Only

RMS实现

- 按照优先级顺序扫描进程列表；
- 选择具有最高优先级的就绪进程运行。

```
/* processes[] is an array of process activation records,
   stored in order of priority, with processes[0] being
   the highest-priority process */
Activation_record processes[NPROCESSES];

void RMA(int current) { /* current = currently executing
process */
  int i;
  /* turn off current process (may be turned back on) */
  processes[current].state = READY_STATE;
  /* find process to start executing */
  for (i = 0; i < NPROCESSES; i++)
      if (processes[i].state == READY_STATE) {
          /* make this the running process */
          processes[i].state == EXECUTING_STATE;
          break;
      }
}
```

Hongqin Shu
Xi'an Jiaotong University
Internal Teaching Use Only

截止时限最近者优先调度

- 截止时限最近者优先：**earliest deadline first, EDF**
- **EDF**是一种动态优先级方案，其策略很简单：
  - 根据截止时限顺序指定优先级。最高优先级的进程是时间上截止时限最近的进程，最低优先级的 进程是截止时限最远的进程。
  - 每次定时器中断，优先级都必须重新计算。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

示例6.4：截止时限最近者优先调度

进程	执行时间	周期
P1	1	3
P2	1	4
P3	2	5

超周期为60

Hongbin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

EDF分析

- Liu和Layland证明：
- EDF可以达到 100%的CPU利用率。
- 如果利用率小于等于1，那么可行调度存在。
- 如果EDF系统过载并且错过了截止时限，在截止时限前，系统将以100%的负荷运行。

Hongqin Sun Xi'an Jiaotong University Internal Teaching Use Only

EDF实现

- EDF的实现比RMS要复杂的多。
- On each timer interrupt:
  - compute time to deadline;
  - choose process closest to deadline.
- Generally considered too expensive to use in practice.

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

RMS vs. EDF

- **RMS**与**EDF**之间，哪一种调度策略更好？
  - 这取决于个人的标准。
- **EDF**可以实现更高的**CPU**利用率，但是调度决策的开销比较大，可能很难诊断出即将过载的可能性。
- **RMS**的**CPU**利用率较低，但是可以更容易保证所有进程的截止时限均能被满足。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

解决调度的问题

- 如果你的进程集不可调度，而又必须满足截止时限怎么办？
  - 换一个更高性能的**CPU**
  - 重新设计进程以降低执行时间
  - 重新修改规格以降低截止时限

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

对模型假设的进一步分析

- 在**RMS**和**EDF**分析中，我们做了很强的假设。这些假设使分析更易处理，但分析的预测在实际中可能不成立。

- 一些错误的预测可能会导致系统错过一些关键的截止时限。因此，明晰这些假设的后果很重要。

Hongqin Sun, Xi'an Jiaotong University
Internal Teaching Use Only

优先级倒置 Priority inversion

- 优先级倒置：优先级较低的进程通过占有资源会阻塞优先级较高的进程的运行。
- 不考虑进程所需资源来调度进程会导致优先级倒置。
  - 低优先级进程**P2**执行时获得总线控制权
  - 高优先级进程**P1**就绪，**P1**抢先**P2**，但**P2**仍保留总线控制权。
  - **P1**请求总线，被拒绝，因为**P2**占用总线。
  - 除非**P1**有办法从**P2**拿到总线控制权，否则会发生死锁**deadlock**。

Hongyin Shi Xiangdaotong University Internal Teaching Use Only

解决优先级倒置

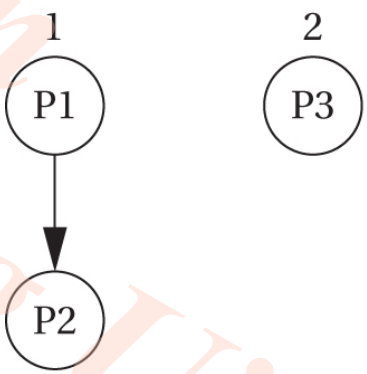
- 解决优先级倒置的最一般的方法：
  - 提高任何向操作系统请求资源的进程的优先级。进程的优先级临时变得高于任何其他进程。这样可保证该进程一旦获得资源就可持续运行，以便其能够使用该资源完成工作后，将资源返回给操作系统。

- 另一种类似方法：
  - 给系统资源分配优先级。使进程继承其所请求资源的优先级。低优先级进程继承资源的优先级后，其优先级变高。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

数据相关

- 了解数据相关性可以帮助我们更高效的使用CPU（utilization）
  - 限制某些特定的进程组合永远不会出现。
  - 临界分析：进程不会同时就绪，最坏情况下，CPU的需求



Task graph

Task	Deadline
1	10
2	8

Task rates

Process	CPU time
P1	2
P2	1
P3	4

Execution times

P1和P2不可能同时执行
P3在一次循环中不可能同时抢先P1和P2

Hongbin Sun Xi'an Jiaotong University International Exchanging Use Only

上下文切换时间

- 上下文切换所占用的时间会使任务调度的裕量更紧张。
- 上下文切换所造成的影响难准确计算---取决于上下文切换的顺序。
- 在实际中，操作系统上下文切换的开销相对于很多通常的任务周期（**ms**到**μs**级）很小（数百个时钟周期）。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

4 进程间通信机制

- 进程间经常需要互相通信。
- 进程通信机制（**interprocess communication mechanism**）由操作系统提供。
- 进程发送信息的方式有两种：
  - 阻塞：在发送了阻塞信息后，进程进入等待状态直到它接收到响应为止。
  - 非阻塞：允许进程在发送信息之后继续执行。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

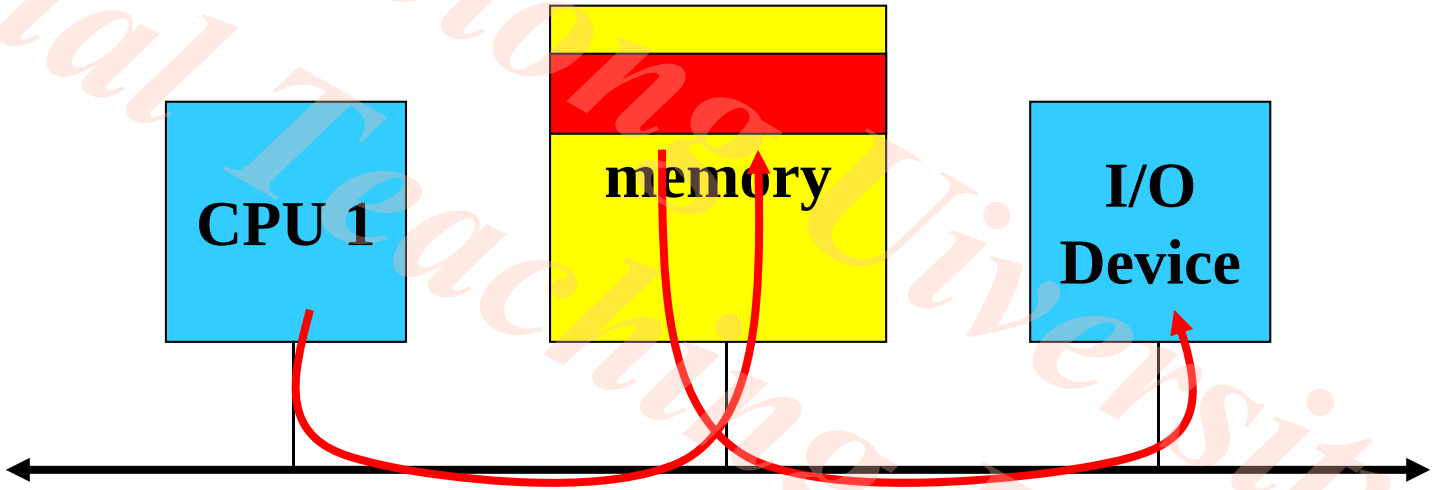
进程间通信的主要形式

- 共享内存 **Shared memory:**
  - **processes have some memory in common;**
  - **must cooperate to avoid destroying/missing messages.**
- 消息传递 **Message passing:**
  - **processes send messages along a communication channel---no common address space.**

Hongqun Sun
Xi'an Jiaotong University
Internal Teaching Use Only

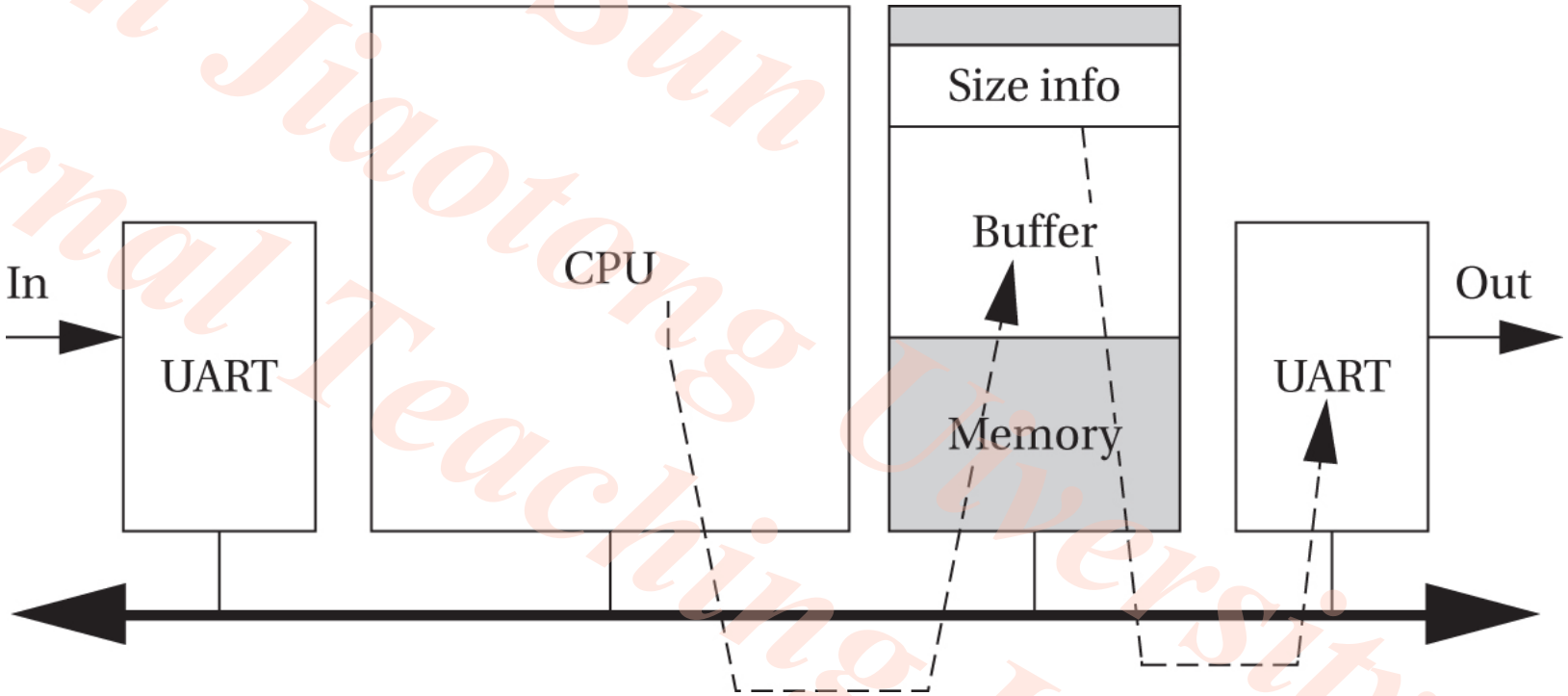
共享内存

- 基于总线系统的共享内存通信



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

示例：作为共享内存的弹性缓冲区



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

临界时间竞争

- 共享内存需要相互通知的标志（**Flag**），附加的共享数据单元。
  - **0**数据尚未准备好，**1**数据已准备好。
- 双向信号的问题:
  - **CPU 1 reads flag and sees 0.**
  - **CPU 2 reads flag and sees 0.**
  - **CPU 1 sets flag to one and writes location.**
  - **CPU 2 sets flag to one and overwrites location.**

Hongshan Sun
Xi'an Jiaotong Liversity
Internal Teaching Use Only

测试-设置原子操作

- 临界时间竞争问题可以用测试-设置原子操作来解决（ **atomic test-and-set**）：
  - 读一个单元、并将其设置，返回测试结果。
  - **False or True**
- 原子（**atomic**）操作：不能被中断的操作。
- 临界区域：一段代码不能被其他进程打断。
  - 访问共享内存
  - 访问**I/O**外设

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

信号量 Semaphores

- 测试-设置可以用来实现信号量，信号量是语言级的同步结构。
- 假设系统提供一个信号量对一块受保护的内存块进行访问。

```
/* some nonprotected operations here */
P(); /* wait for semaphore */
/* do protected work here */
V(); /* release semaphore */
```

  - **Get access to semaphore with P().**
  - **Perform critical region operations.**
  - **Release semaphore with V().**

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

ARM测试-设置操作示例

- ARM使用SWP(交换)指令来实现测试-设置操作。

  SWP Rd, Rm, Rn //Rd=Rm, Rd与[Rn]内容交换

```
                ADR r0,SEMAPHORE
                LDR r1,#1
GETFLAG         SWP r1,r1,[r0]
                BNZ GETFLAG
                HASFLAG …
```

- 测试从内存中所获得的信号量是否为0。0，则信号量可用；非0，则再次尝试获取。

Hongkin Sun Xi'an Jiaotong University Internal Teaching Use Only

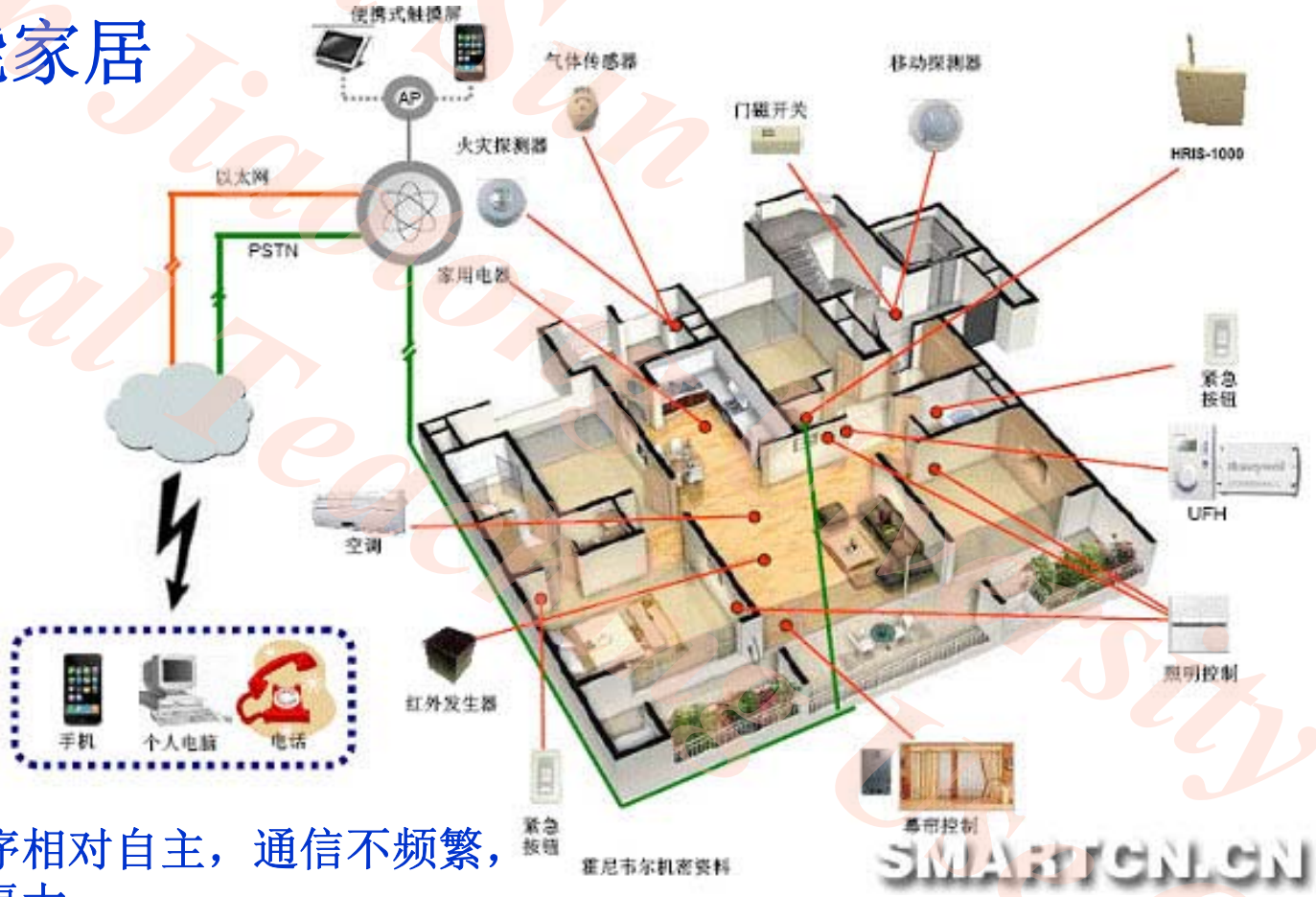
消息传递

- 消息传递通信补充内存共享模型。消息不是存储在通信链路上，而是在线路两端的发送方和接收方上。
- 共享内存模型：数据存储在通信链路/内存中。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

消息传递示例

智能家居

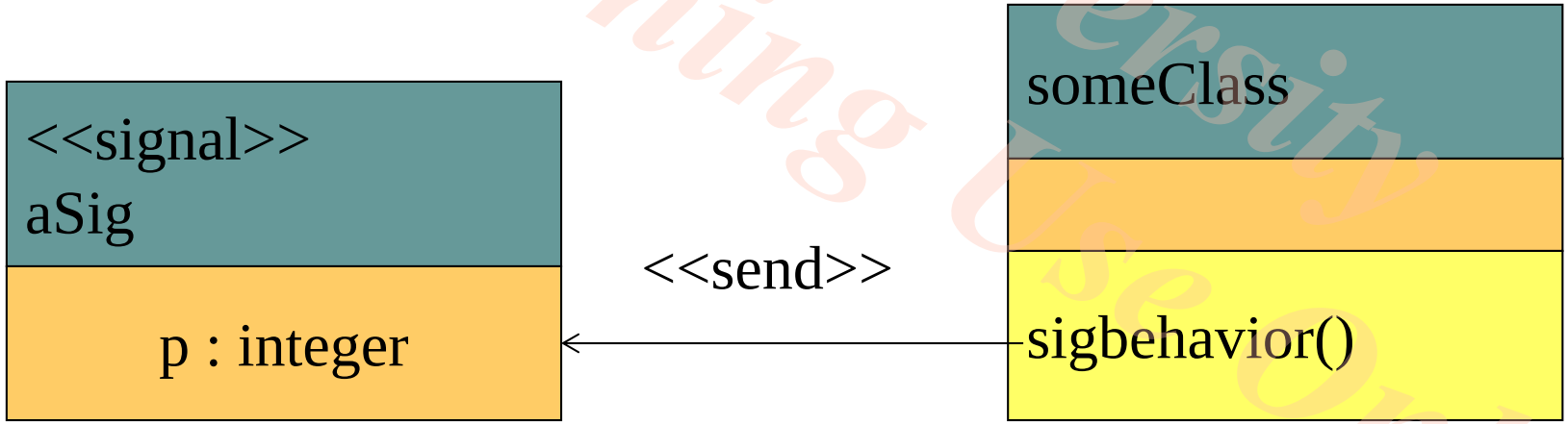


应用程序相对自主，通信不频繁，物理间隔大。

Honeywell Sun Xiang Jiaotong University Internal Teaching Use Only

信号 Signal

- 在Unix中常用到的另一种进程间通信的方式是信号。
- **More general than Unix signal---may carry arbitrary data:**



Hongbin Sun
Xi'an Jiaotong Uiversity
Internet Teaching Use Only

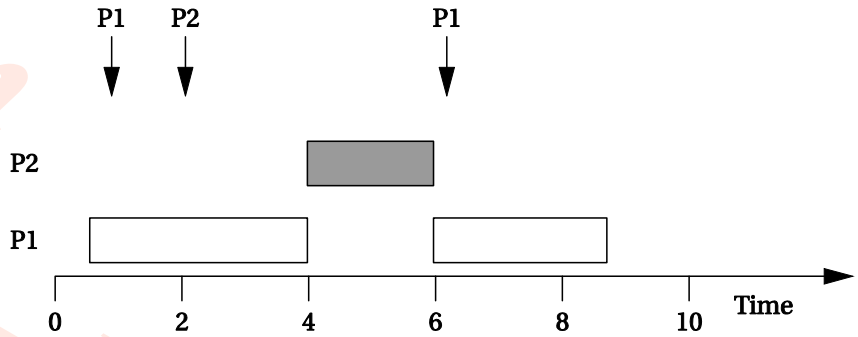
5 评估操作系统的性能

- 进行调度策略分析时，使用了简化的假设：
  - 上下文切换的时间为0。
    - 由于上下文切换时间比进程的执行时间要小得多，忽略它是合理的。但在某些情况会明显增加时延。
  - 已知进程的执行时间。
    - 程序时间不是一个的数字，而是由最差执行时间和最好执行时间来界定的。
  - 孤立的确定单个进程的最差和最好执行时间。
    - 进程在高速缓存中是相互影响的，进程间的高速缓存冲突可能会显著降低进程的执行时间。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

调度与上下文切换的开销

进程	执行时间	截止时限
P1	3	5
P2	3	10



如果假设初始化进程的总时间（包括上下文切换和调度策略评估）为1个时间单元，则没有可行的调度。

2TP1 + TP2 = 2*(1+3)+(1_3)=11

零耗时的上下文切换通常是合理的近似，问题多数出在一些最高速率的进程中。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

进程执行时间

- 进程执行时间不是恒定的。数据相关的特性和高速缓存影响都会导致运行时间的很大变化。
- 确定最差执行时间→未使用的**CPU**时间。
- 多余的**CPU**时间可能是好事。
  - 进程执行更快，更容易满足截止时限。
- 多余的**CPU**时间也可能是坏事:
  - 下一个进程早启动，可能会导致新的抢占。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

进程与高速缓存

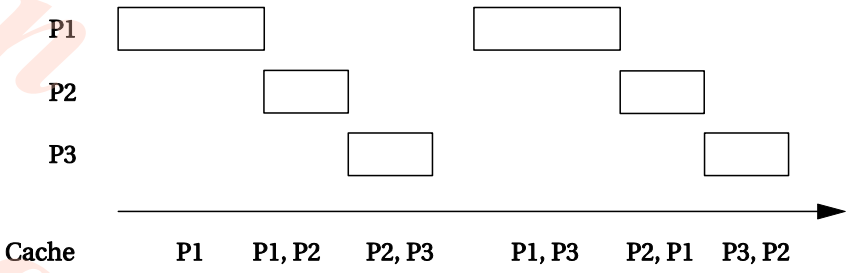
- **Cache**会导致一个程序的执行影响其他程序的执行时间。当多个程序在同一**cache**中时，界定单个程序基于**cache**性能的技术就不起作用了。
- 很多实时嵌入式操作系统是基于无**cache**设计的，由于缺少分析**cache**行为的工具。
- 后果：过度设计的硬件，远超所需的计算能力。
- 通过高速缓存效果的分析，我们可以更好使用可用硬件。

Honsoth Sun Jiaotong University
Xi'an Jiaotong University
Internal Teaching Use Only

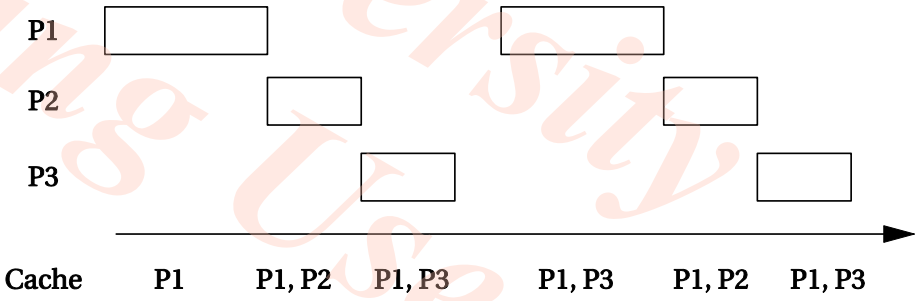
高速缓存上的调度效果

Process	WCET	Avg. CPU time
P1	8	6
P2	4	3
P3	4	3

Schedule 1 (LRU cache):



Schedule 2 (half of cache reserved for P1):



Honglin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

6 进程的功耗管理与优化

- 电源管理策略Power management policy:确定何时执行何种电源管理操作的策略。
- 操作系统可以像管理时间一样管理功耗。
- 操作系统通过关闭一些系统部件来降低功耗.
  - 也可以有部分关闭的功能。
- 电源管理策略需要基于静态和动态电源管理机制的特性来设计。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

电源管理与性能

- 电源管理与性能通常是对立的
- 进入掉电模式将消耗
  - **energy,**
  - **time.**
- 退出掉电模式将消耗
  - **energy,**
  - **time.**
- 应该小心切换模式，决定时需要对系统进行全面的分析。

Hongqin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

简单的电源管理策略

- 需求驱动**Request-driven**: 当收到请求时系统加电。增加响应延迟，是否可接受？
- 预测关闭**Predictive shutdown**: 预测下一个请求何时到达，并在此之前启动系统，以为请求者节省启动时间。
  - 请求者可能不得不等待活动周期到来。在最坏情况下，会错过截止时限。
  - 系统可能会在没有临近活动的情况下自动重启。结果，系统会浪费能耗。

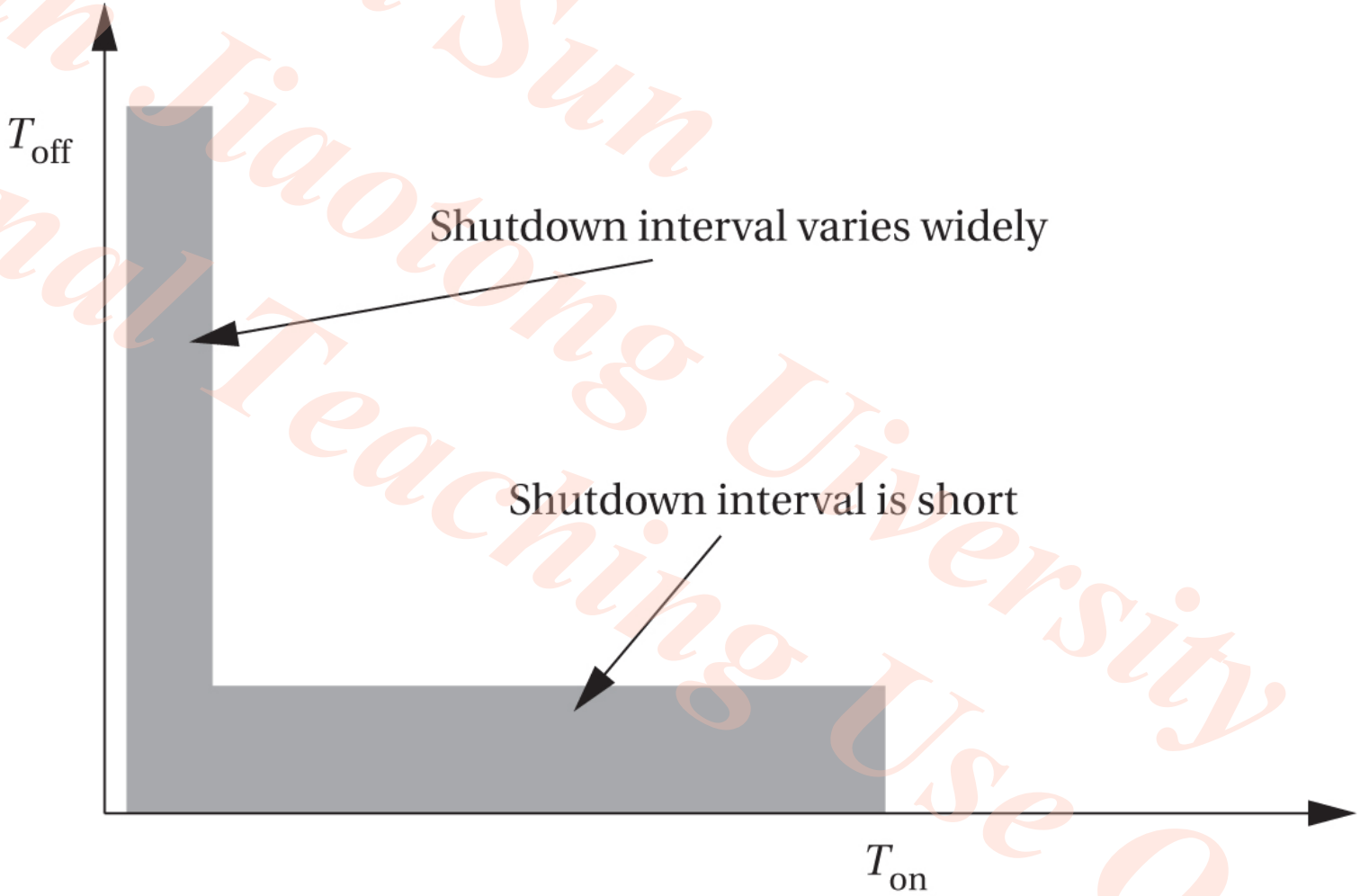
Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

Probabilistic shutdown

- 假设服务请求是符合概然时间的。基于预期行为的概率模型来猜测系统的活动行为。
- 选择好的概率模型非常重要，策略机制也不能过于复杂，因为做决策也消耗功耗。
- 优化的目标包括:
  - **power consumption;**
  - **response time.**
- 最简单的技术就是固定时间: **shut down after time T_{on}, turn back on after waiting for T_{off}.**

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

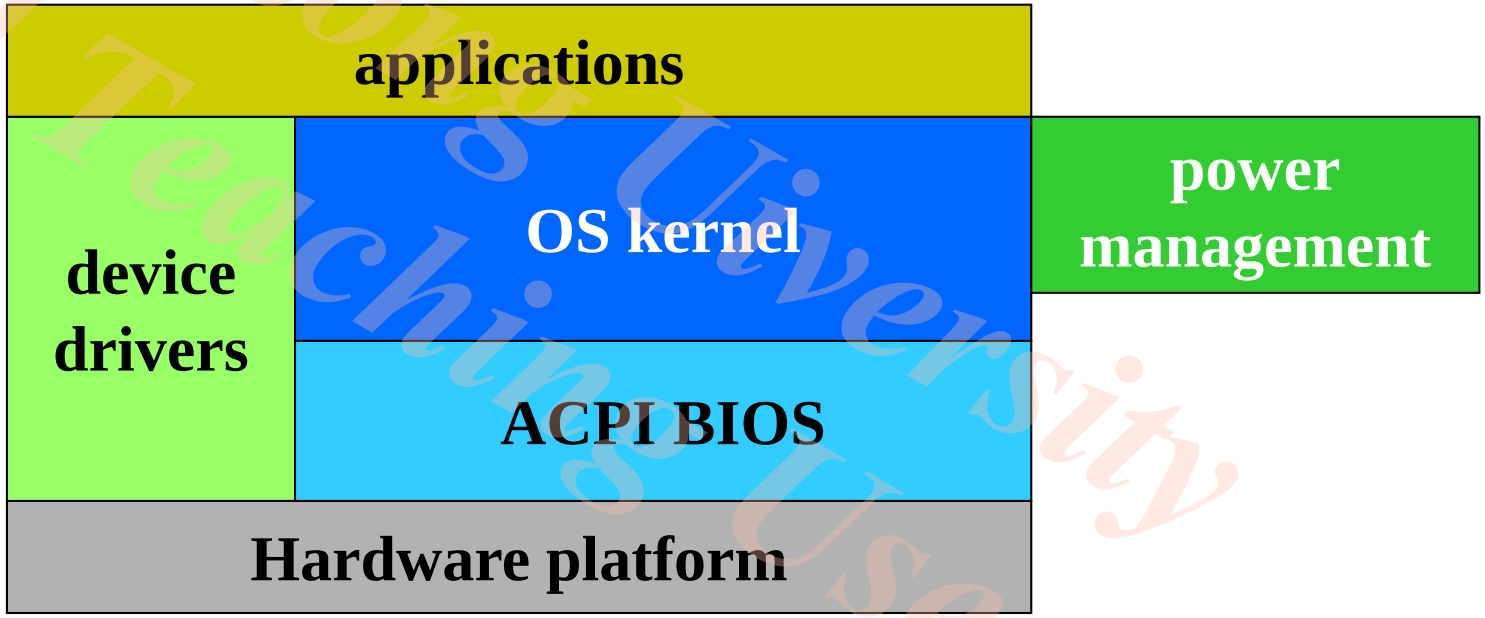
图形终端的概率分布



Hongbin Sun
Xi'an Jiaotong Uiversity
Internal Teaching Use Only

高级配置和电源接口

- **ACPI(Advanced Configuration and Power Interface)**是一个电源管理服务的开放工业标准。



Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

ACPI的全局电源状态

- **G3:** 机械关闭状态，系统无功率消耗。
- **G2:** 软关闭状态，需要整个操作系统重启来恢复到工作状态。
  - **S1:** 无系统上下文丢失的低唤醒延迟状态
  - **S2:** 丢失**CPU**和系统高速缓存状态的低唤醒延迟状态
  - **S3:** 除主存外其他所有系统状态都会丢失的低唤醒延迟状态
  - **S4:** 低功耗休眠状态，所有设备均被关闭
- **G1:** 休眠状态，系统看上去是关闭的，其重回工作状态的时间与功耗成反比。
- **G0:** 工作状态，系统处于完全可用状态

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only

典型的电源管理程序

- 观察器：通过**ACPI**接口接收描述系统行为的消息。

- 判定模型：基于观察结果确定电源管理行为。

Hongbin Sun Xi'an Jiaotong University Internal Teaching Use Only