嵌入式系统设计与应用

第六章 进程和操作系统（1）

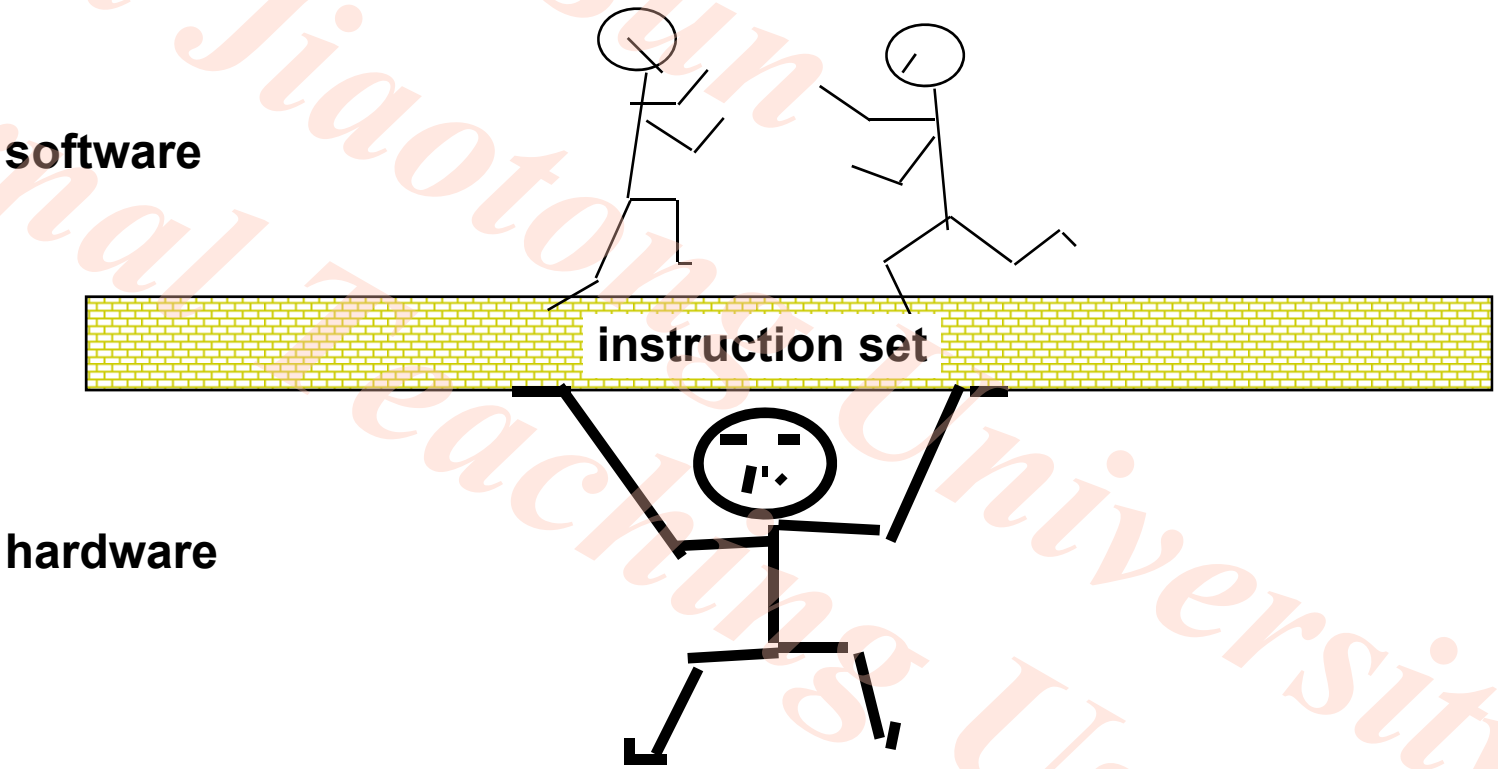
西安交通大学电信学院
孙宏滨

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

什么是操作系统？

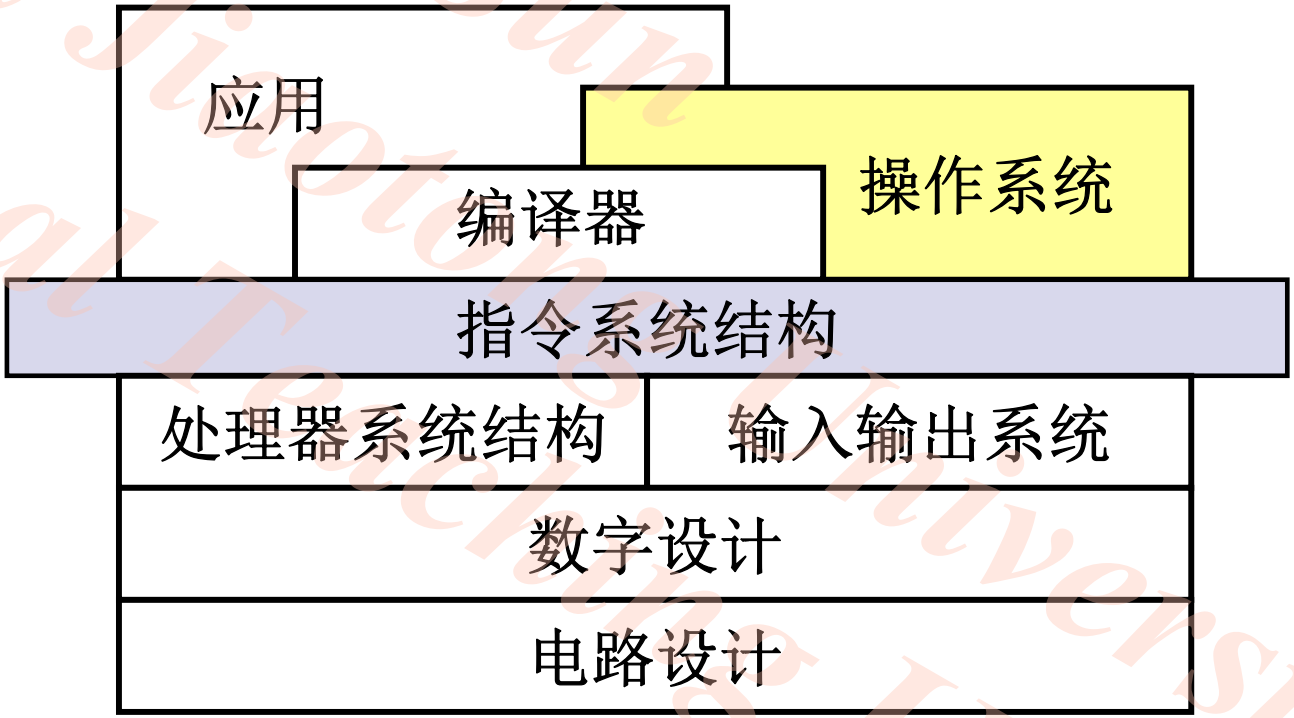
Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

什么是操作系统？



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

什么是操作系统？



A program that manages a computer's hardware, provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

操作系统的功能（wikipedia）

- 管理计算机硬件与软件资源的计算机程序，同时也是计算机系统的内核与基石。

Honorin Science
Xi'an Jiaotong University
Internal Teaching Use Only

操作系统的功能（wikipedia）

- 进程管理：决定系统资源供需的优先次序
- 内存管理：**Memory management**
- 文件系统：**Storage management**
- 网络通信：**Networking**
- 安全机制：**Protection and Security**
- **I/O**设备管理：**I/O**设备驱动
- 用户界面：有些操作系统集成了图形用户界面（**GUI**），而有些仅使用命令行界面（**CLI**），而将**GUI**视为一种非必要的应用程序。

Honorary Sun Xiaoyang Sun Yat-sen University International Teaching Use Only

嵌入式操作系统？

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

嵌入式操作系统

- 嵌入式操作系统是嵌入式系统的操作系统。它们通常被设计非常紧凑有效，抛弃了运行在它们之上的特定的应用程序所不需要的各种功能。嵌入式操作系统多数也是实时操作系统。

--- wikipedia

Honeyin Software
Xi'an Jiaotong University
Internal Teaching Use Only

实时操作系统？

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

实时操作系统

- 提供满足实时需求的操作系统。
  - 用考虑了实施需求的算法来分配系统资源
  - 通用操作系统多采用公平性原则分配资源，即将CPU平均分配给所有的进程，这样的资源分配没有考虑时间因素，容易导致进程错过其执行的截止时限。

- **WinCE、VxWorks、μC/OS-Ⅱ、RTLinux**

Honolulu Saint Louis University
Xiamen University
Internet Teaching Use Only

移动操作系统

- **Mobile operation system**：是指在移动设备上运作的操作系统。
- 移动操作系统近似在台式机上运行的操作系统，但是它们通常较为简单，而且提供了无线通信的功能。
- 使用移动操作系统的设备有智能手机、**PDA**、平板电脑等，另外也包括嵌入式系统、移动通信设备、无线设备等。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

主流的移动操作系统

- iOS：苹果，封闭，13.4%
- Android：谷歌，开源， 81.3%
- Window Phone：微软，封闭，4.1%

- BlackBerry 10：1%
- Tizen：已放弃

市场需求并不足以支撑三种以上操作系统

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

1 任务与进程 Task and Process

- 任务是对一组相关联的操作的功能描述。
- 任务也可以为理解为一组进程。
- 进程是程序的一次**unique**的执行。
  - Several copies of a program may run simultaneously or at different times.
- 每个进程都拥有自己的状态(寄存器和存储器)和地址空间(线程)
- 操作系统负责管理进程

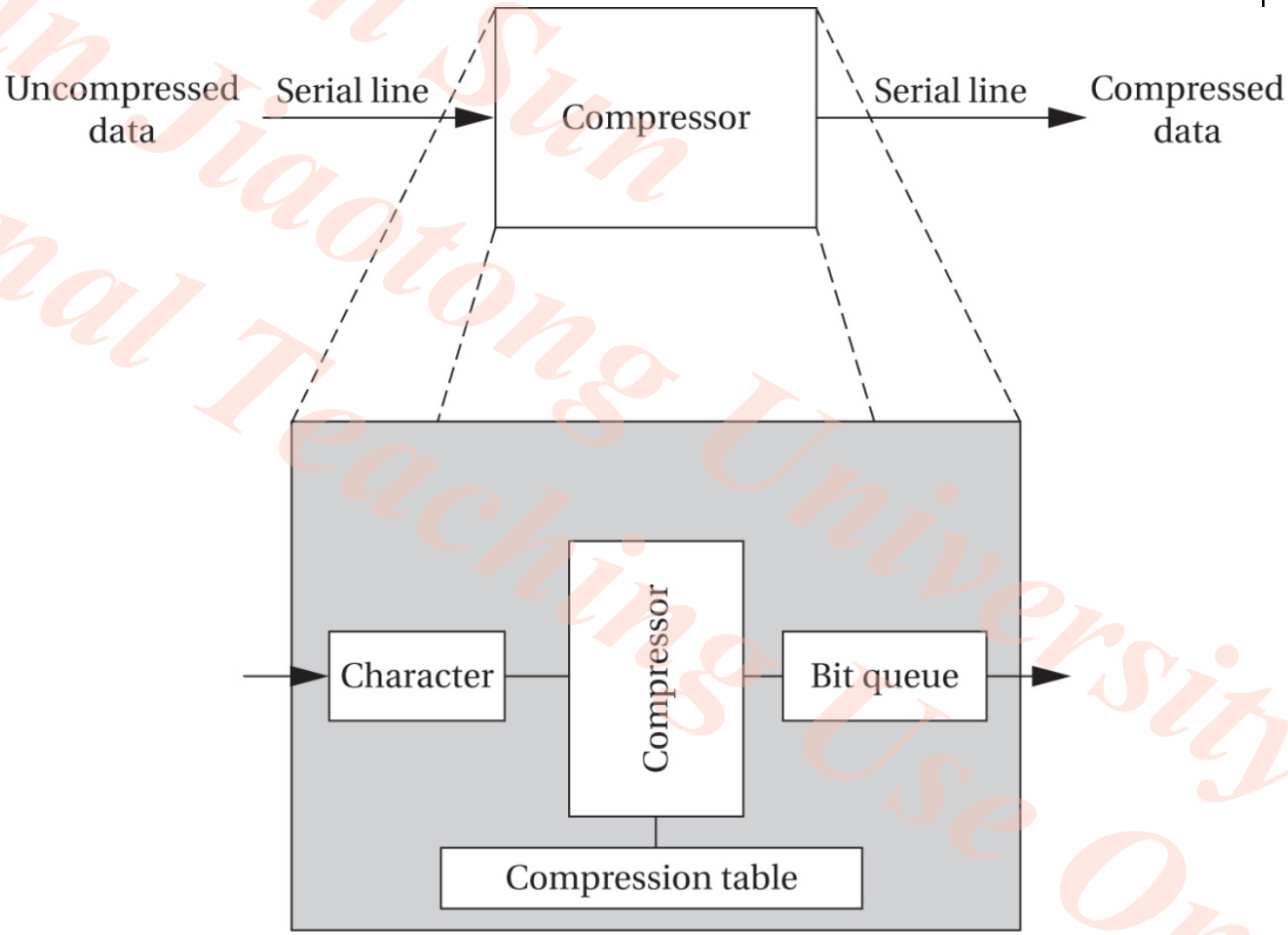
Hongjun Sun
Xi'an Jiaotong University
Internal Teaching Use Only

为何需要多进程？

- 多任务本身意味着多进程。
- 多进程有助于处理时序复杂的问题：
  - 多速率 **multiple rates**
    - 多媒体 **multimedia**
    - 汽车 **automotive**
  - 异步输入 **asynchronous input**
    - 用户接口 **user interfaces**
    - 通信系统 **communication systems**

Hongqin Shi
Xi'an Jiaotong University
Internal Teaching Use Only

示例：即时数据压缩



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

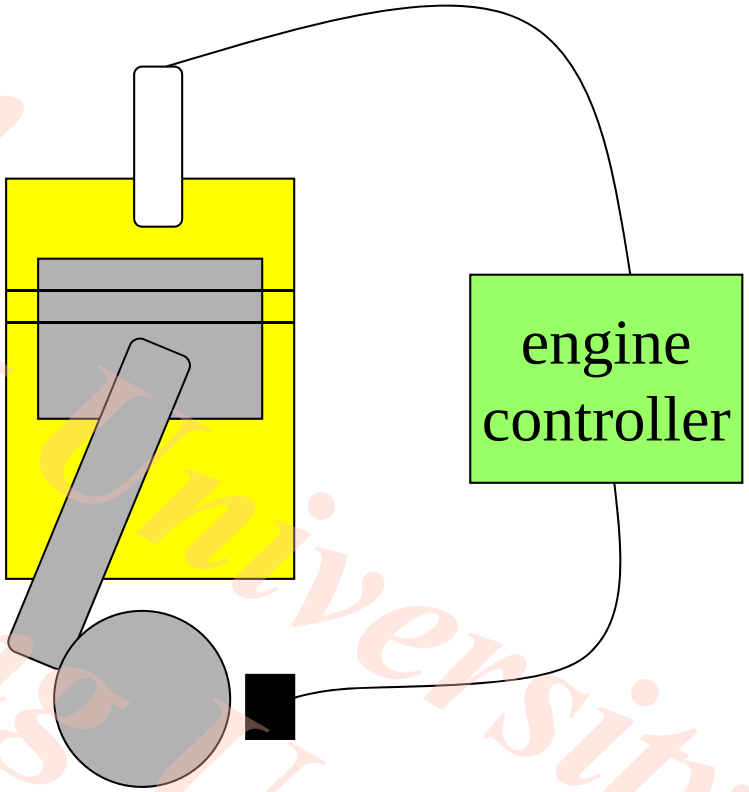
多速率系统 Multi-rate systems

- 任务可以是同步的，也可以是异步的。
- 同步任务的发生速率可能会不同。
- 进程根据任务的计算需求差异运行在不同的速率上。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

示例：汽车引擎控制

- 任务：
  - spark control
  - crankshaft sensing
  - fuel/air mixture
  - oxygen sensor
  - Kalman filter



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

引擎控制器中的典型速率

Variable	Full range time (ms)	Update period (ms)
Engine spark timing	300	2
Throttle	40	2
Air flow	30	4
Battery voltage	80	4
Fuel flow	250	10
Recycled exhaust gas	500	25
Status switches	100	20
Air temperature	Seconds	400
Barometric pressure	Seconds	1000
Spark (dwell)	10	1
Fuel adjustment	80	8
Carburetor	500	25
Mode actuators	100	100

Hongyun Sun
Xi'an Jiaotong University
Internal Teaching Use Only

进程的时限要求

- 进程的时限要求对调度方案的选择影响很大。

- 释放时间**Release time**：进程处于准备执行状态的时刻，此时进程尚没有取得**CPU**的控制权，也没有开始运行。

- 截止时限**Deadline**: 进程必须结束的时刻。

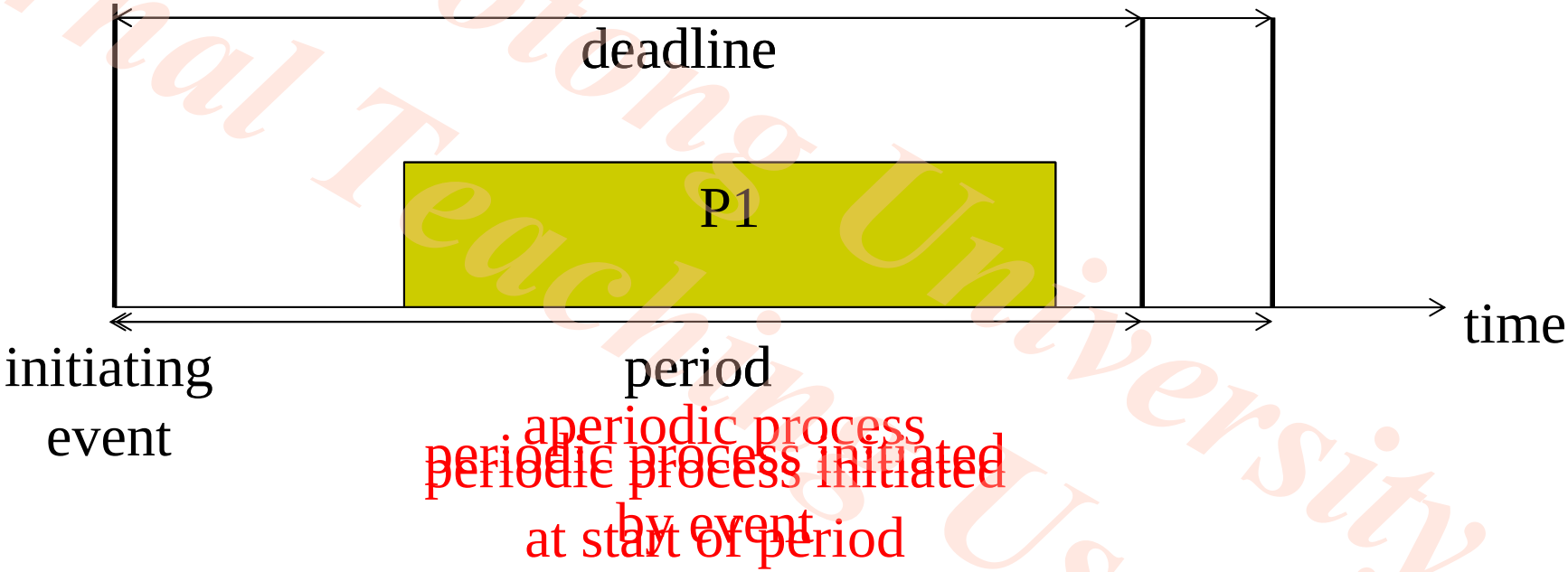
Honoun Sun Xi'an Jiaotong University International Teaching Use Only

实时（Real-time）系统

- 按照外部时序约束进行计算
- 截止时限的频率:
  - 周期性 **Periodic**
  - 非周期性 **Aperiodic**
- 截止时限的类型:
  - **Hard: failure to meet deadline causes system failure.**
  - **Soft: failure to meet deadline causes degraded response.**
  - **Firm: late response is useless but some late responses can be tolerated.**

Hongbin Sun
Xi'an Jiaotong University
Internal Use Only

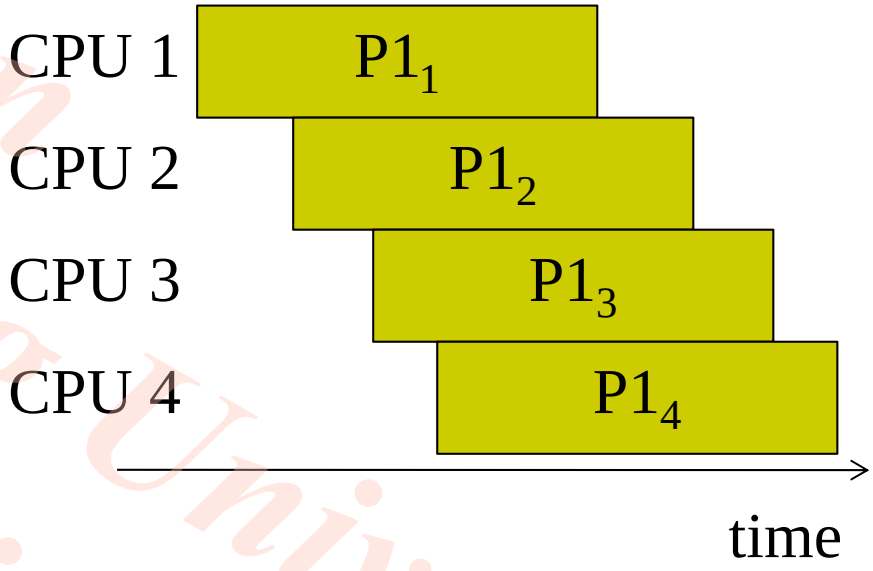
释放实现与截止时限



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

进程的速率需求

- 速率需求指明进程必须以多快的速度被启动。
- 周期**Period**：连续两次执行之间的时间。
- 速率**Rate**：周期的倒数.
- 进程启动时间间隔可以小于其周期：进程的流水线执行。



Hongqin Sun
Xi'an Jiaotong University
International Teaching Use Only

违反时限限制

- 如果进程错过了截止时限会发生什么？
  - 硬截止时限**Hard deadline: system fails if missed.**
    - 车辆控制系统：灾难性的后果
  - 软截止时限**Soft deadline: user may notice, but system doesn't necessarily fail.**
    - 多媒体娱乐系统：声音或视频的毛刺

Hongsun Sah
Xi'an Jiaotong University
Internal Teaching Use Only

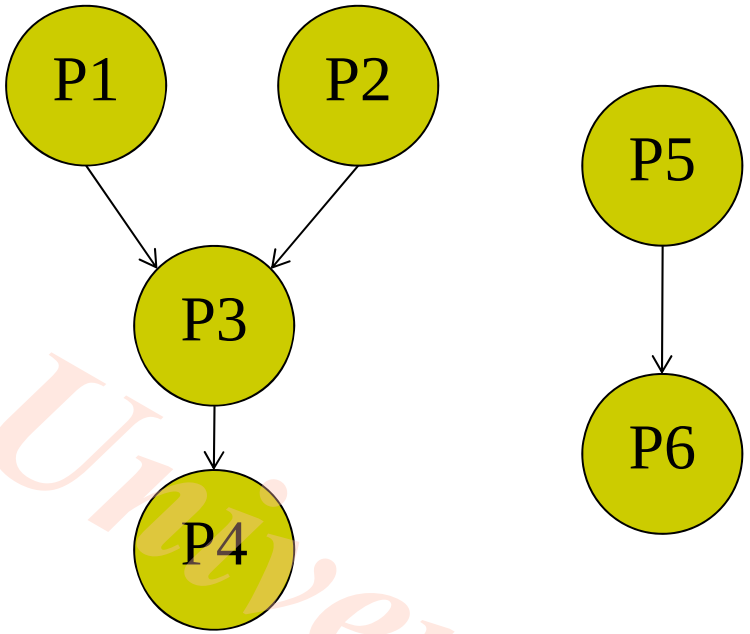
示例：一个航天飞机的软件错误

- **Space Shuttle's first launch was delayed by a software timing error:**
  - **Primary control system PASS and backup system BFS.**
  - **BFS failed to synchronize with PASS.**
  - **Change to one routine added delay that threw off start time calculation.**
  - **1 in 67 chance of timing problem.**

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

任务图

- 进程间可能存在数据相关性---执行顺序受限于进程间的数据传递。
- 任务图（Task graph）：具有数据相关性的一组进程.
- 任务：一组由数据相关连接的节点
- 任务集：多个任务，整个图是一个任务.



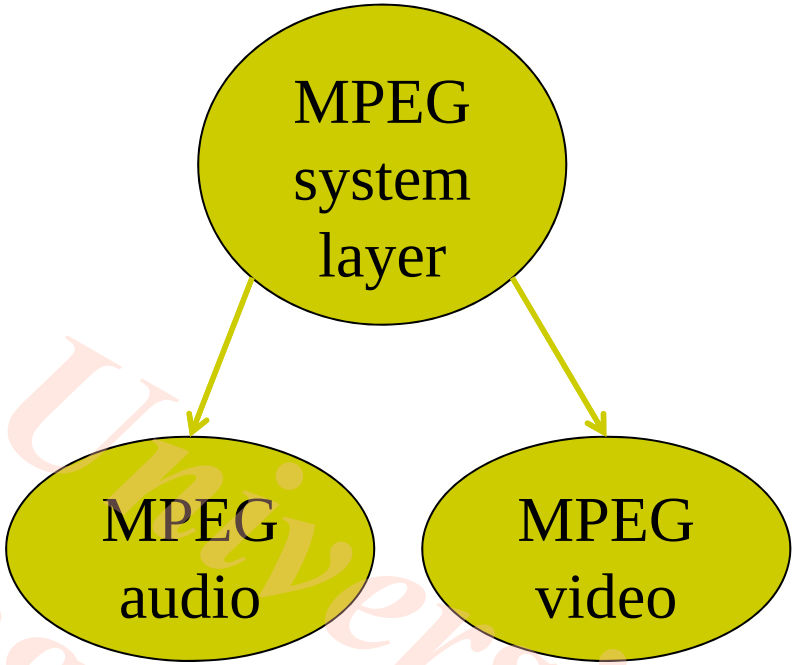
任务 1　　任务 2

任务集

Hongbin Sun
Xi'an Jiaotong University
Intelligent Learning Use Only

任务间通信 – 不同速率的进程

- 任务图假设每个任务中的所有进程运行速率相同，任务之间不需要通信。
- 现实中，不同速率的进程之间的通信非常普遍。
  - It’s hard to require immediate response for multi-rate communication.



音视频同步

不同速率的进程间的通信

Hongwen Sun
Xi'an Jiaotong University
Internal Teaching Use Only

CPU度量

- 进程执行时间 T_i.
  - 未被抢占的执行时间：CPU时间≠启动时间 – 结束时间
  - 可能的时间单位: seconds, clock cycles.
  - Worst-case, best-case execution time may be useful in some cases.
- 执行时间波动来源:
  - Data dependencies.
  - Memory system.
  - CPU pipeline.

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

利用率

- **CPU利用率:**
  - **CPU**执行有用计算的时间占总时间的比率**%**。
  - 通常假设没有调度耗费。

- 利用率:
  - **U = (CPU time for useful work)/ (total available CPU time)**

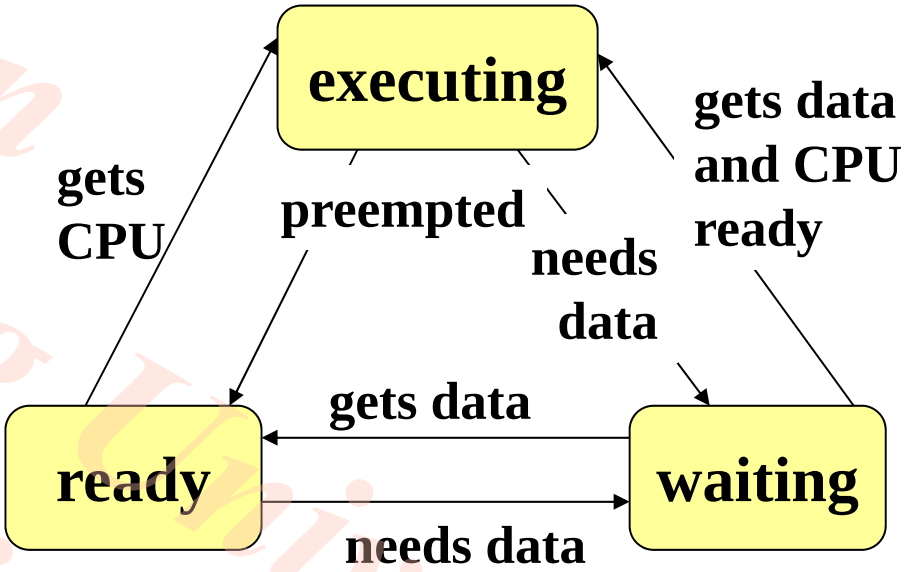
    $= [\Sigma_{t1 \le t \le t2} T(t)] / [t2 - t1]$

    $= T/t$

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

进程的状态

- 进程可以是以下三种基本状态中的一种：
  - 执行：executing on the CPU;
  - 就绪：ready to run;
  - 等待：waiting for data.



操作系统最重要的工作就是决定后续要运行的进程。选择运行进程的顺序的过程被称为调度 **scheduling.**

Hongyun Sun
Xi'an Jiaotong University
Internal Teaching Use Only

调度策略 scheduling policy

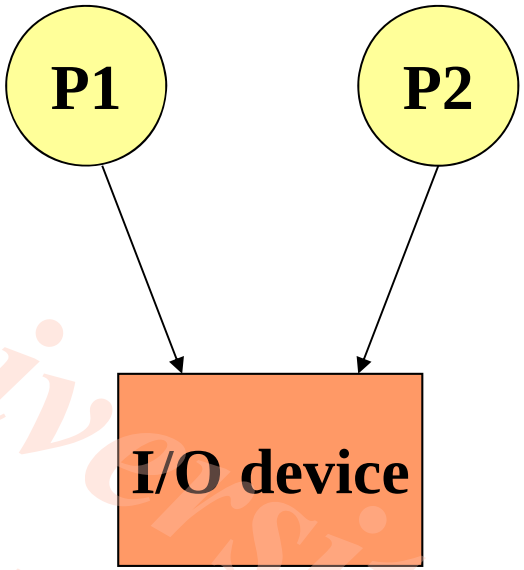
- 调度策略定义了如何选择进程从就绪态提升到运行态。
  - 我们能满足所有的截止时限吗? 必须在所有情况下满足截止时限。
  - 满足截止时限我们需要CPU有多大马力?
- 合理的调度策略不仅能确保满足所有截止时限，对完成任务所需的CPU马力也有重要影响。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

可调度性

- 可调度性表明是否存在一种调度执行使系统的所有进程能满足所有截止时限。

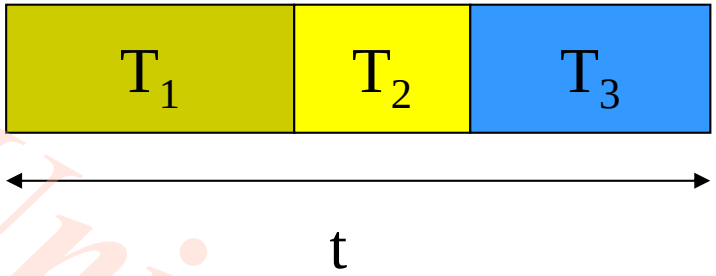
- 资源约束使可调度性分析成为**NP-hard**问题。
  - 对于所有的资源请求时序都能满足截止时限。



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

简单的可调度性分析 – CPU利用率

- 假设:
  - 没有资源冲突.
  - 固定的进程执行时间.



- 要求:
  - $t \geq \Sigma_i T_i$
  - CPU的利用率不可能超过100%.

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

超周期 Hyperperiod

- 在有限时间内评估CPU利用率。
- 对于周期性进程，使用超周期：所有进程周期的最小公倍数。

- 如果进程周期选择不当，超周期可能会非常长。

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

超周期示例

- 长超周期:
  - P1 7 ms.
  - P2 11 ms.
  - P3 15 ms.
  - LCM = 1155 ms.
- 短超周期:
  - P1 8 ms.
  - P2 12 ms.
  - P3 16 ms.
  - LCM = 96 ms.

least-common multiple, LCM
最小公倍数

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

简单的CPU利用率计算

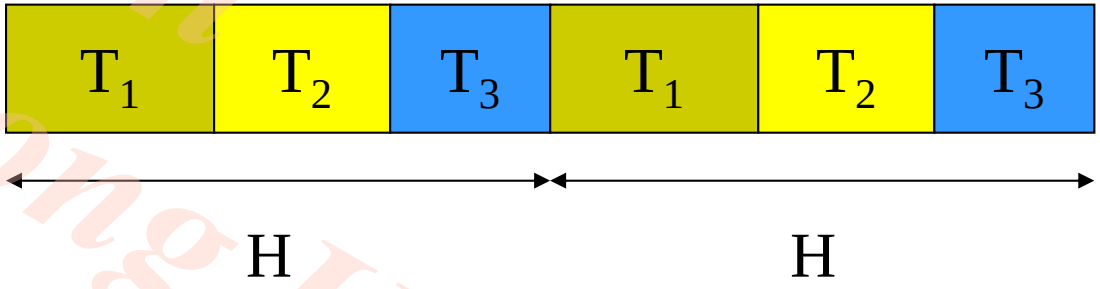
- P1 period 1 ms, CPU time 0.1 ms.
- P2 period 1 ms, CPU time 0.2 ms.
- P3 period 5 ms, CPU time 0.3 ms.

LCM		5.00E-03	
	peirod	CPU time	CPU time/LCM
P1	1.00E-03	1.00E-04	5.00E-04
P2	1.00E-03	2.00E-04	1.00E-03
P3	5.00E-03	3.00E-04	3.00E-04
	total CPU/LCM		1.80E-03
	utilization		3.60E-01

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

静态循环/时分多路调度

- 静态循环调度：以相等间隔将超周期**H**划分为相同大小的时间片。

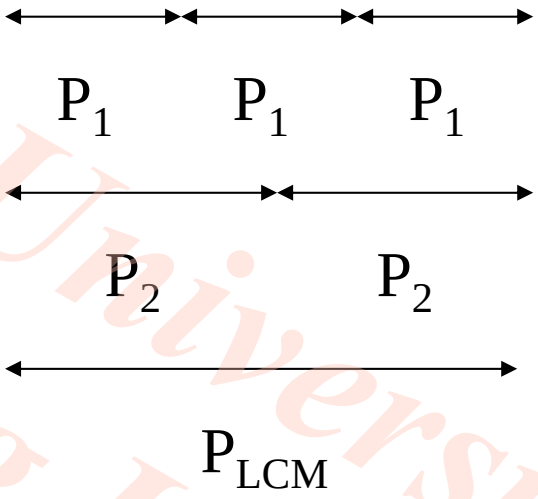


- 进程总是在相同的时间片内运行。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

时分多路的假设

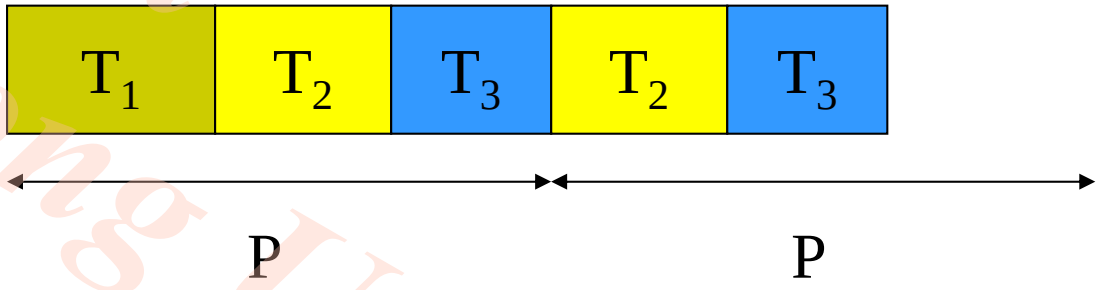
- 调度基于进程周期的最小公倍数.
- 调度决策简单 -> 调度开销很小



Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

轮询 Round-robin

- 只调度处于就绪状态的进程
  - 评估进程的顺序总是相同的



- 为空的时间片可以用来执行临时的、无截止时限的非周期性任务。

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

轮询的假设

- 基于所有进程周期的最小公倍数调度
- 进程的时间片大小相同
- 调度策略简单 -> 低调度开销
  - 常被用于诸如总线等硬件中

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

调度开销

- 调度开销**scheduling overhead**：选择下一个执行进程所需要的时间。
- 通常，调度策略越复杂，需要的调度时间越长。复杂的调度策略通常可以获得较高的理论利用率，但是开销也很大。

- 最终决定调度策略必须同时考虑理论利用率与实际的调度开销。

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

运行周期性进程

- 需要以不同的速率运行周期性进程
- 为了简化实现的描述，我们假设：
  - **process = subroutine.**

Honolulu Sun Xiaoshan University Internal Teaching Use Only

while循环实现

- 最简单实现是使用循环。

```
while (TRUE) {
    p1 ( );
    p2 ( );
}
```

用循环实现的问题：

1. 无法控制时间：尽可能快的执行速度、不可预测
2. 所有进程以相同的速率运行

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

定时循环实现

- 使用定时器来产生周期性中断，在中断服务程序中调用**pall**，实现进程的周期执行

```
void pall(){
    p1();
    p2();
}
```

但是，如果其中一个进程运行时间过长。。。？定时器中断将使**CPU**中断管理屏蔽它的下次中断，直到**pall()**程序返回。结果，下一次循环就会延迟启动。

此外，如何控制多速率进程？

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

多个定时器实现

- 将具有相同速率的进程合并成一个函数。
- 每个函数对应一个定时器，每个定时器的速率设置不同。
- 可能会没有足够的定时器来实现所有速率。

```
void pA (){ /* rate A */
    p1 ( );
    p3 ( );
}

void pB (){ /* rate B */
    p2 ( );
    p4 ( );
    p5 ( );
}
```

Hongyun Sun
Xi'an Jiaotong University
Interest Teaching Use Only

定时器+计数器实现

- 用软件计数器进一步分（divide）定时器。
- 进程的速率需要是定时器频率的整数倍。否则调度将变得复杂且易出错。

```
int p2count = 0;
void pall(){
    p1();
    if (p2count >= 2) {
        p2();
        p2count = 0;
        }
    else p2count++;
    p3();
}
```

Hongbin Sun
Xi'an Jiaotong University
Internal Teaching Use Only

实现进程

- 所有这些实现方法都是不充分的。
- 需要更好的时限控制，需要更高的可靠性。
- 需要比子程序更好的机制。

- 需要实时操作系统（**Real-Time Operating System, RTOS**）

Hongqin Sun
Xi'an Jiaotong University
Internal Teaching Use Only