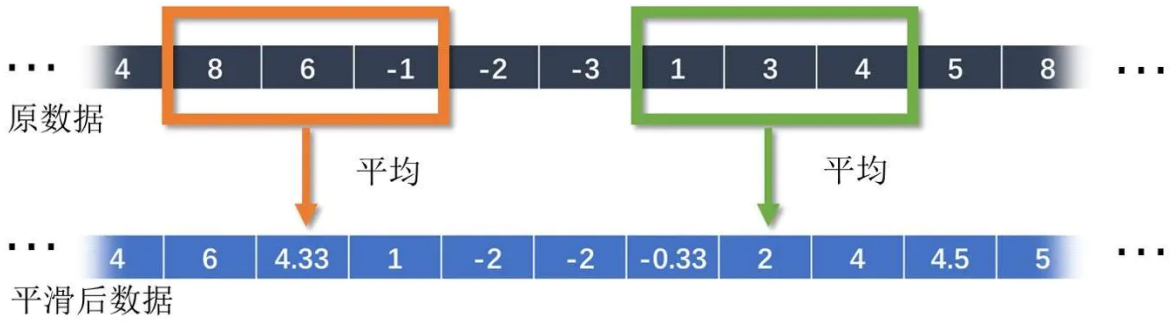
滑动平均滤波

原	
Stakeholders	
Status	
Type	
Created	@December 22, 2022 2:19 PM
Last Edited Time	@December 23, 2022 6:46 PM
Last Edited By	

滑动平均滤波法是一种FIR滤波法,其原理是对一组数据yt(n)中的每个点分别进行N - 1 点移动平均后得到滤波结果fyt(n),其对应的噪声减少率NRR=1/N .其滤波算法模型描述如下：

$$fyt[n] = \frac{1}{N}\sum_{i=0}^{N-1} yt[n-i], (n \geq N-1) \qquad (1)$$

方法：把连续取N个采样值看成一个队列，队列的长度固定为N，每次采样到一个新数据放入队尾，并扔掉原来队首的一次数据.(先进先出原则)把队列中的N个数据进行算术平均运算，就可获得新的滤波结果N值的选取



优点：拥有保低频滤高频的特点，而且对于特点频率的滤波具有良好的效果。对周期性干扰有良好的抑制作用，平滑度高，适用于高频振荡的系统

缺点：所有频率分量的信号都会有不同程度衰减，灵敏度低，对偶然出现的脉冲性干扰的抑制作用较差，不易消除由于脉冲干扰所引起的采样值偏差，不适用于脉冲干扰比较严重的场合，比较浪费RAM

滑动平均过程，本质上等同于一个卷积运算，其卷积核为一个简单的矩形脉冲，卷积核的长度，可以减少随机白噪声，同时保持最清晰的阶跃响应。随着滑动窗口选择得越长，随机噪声抑制效果越好，但对应的边缘清晰度（阶跃响应的过渡速度）也越差。

但是，从频域看，滑动平均滤波器是对频域编码信号最不友好的滤波器，它几乎没有能力将一个频带与另一个频带分开。由滑动平均滤波器发展来的滤波器还包括高斯滤波，布莱克曼滤波和多次滑动平均滤波，这些改进的滑动平均滤波器在频域中具有稍好的性能，但以增加的计算时间为代价。

实时运算（Real-time computing）是计算机科学中对受到“实时约束”的计算机硬件和计算机软件系统的研究，实时约束像是从事件发生到系统回应之间的最长时间限制。实时程序必须保证在严格的时间限制内响应。

实时系统是指在系统工作时，能在特定的时间内完成特定的任务，其各种资源可以根据需要进行动态的分配，因此其处理事务的能力强，速度快。通常，实时操作系统分为两大类:

1.事件驱动型。当一个高优先级的任务需要执行时，系统会自动切换到这个任务。这种根据优先级调度任务的方式称为抢占式任务处理。
2.时间触发型。每个任务在各自设定好的的时间间隔内重复、轮流调度。
时间触发型设计往往比较严格地调度任务，具有更好的多任务处理能力。多个任务被不停地轮流调度，在宏观上，就相当于一个CPU同时执行多个任务。

1）高精度计时系统

计时精度是影响实时性的一个重要因素。在实时应用系统中，经常需要精确确定实时地操作某个设备或执行某个任务，或精确的计算一个时间函数。这些不仅依赖于一些硬件提供的时钟精度，也依赖于实时操作系统实现的高精度计时功能。

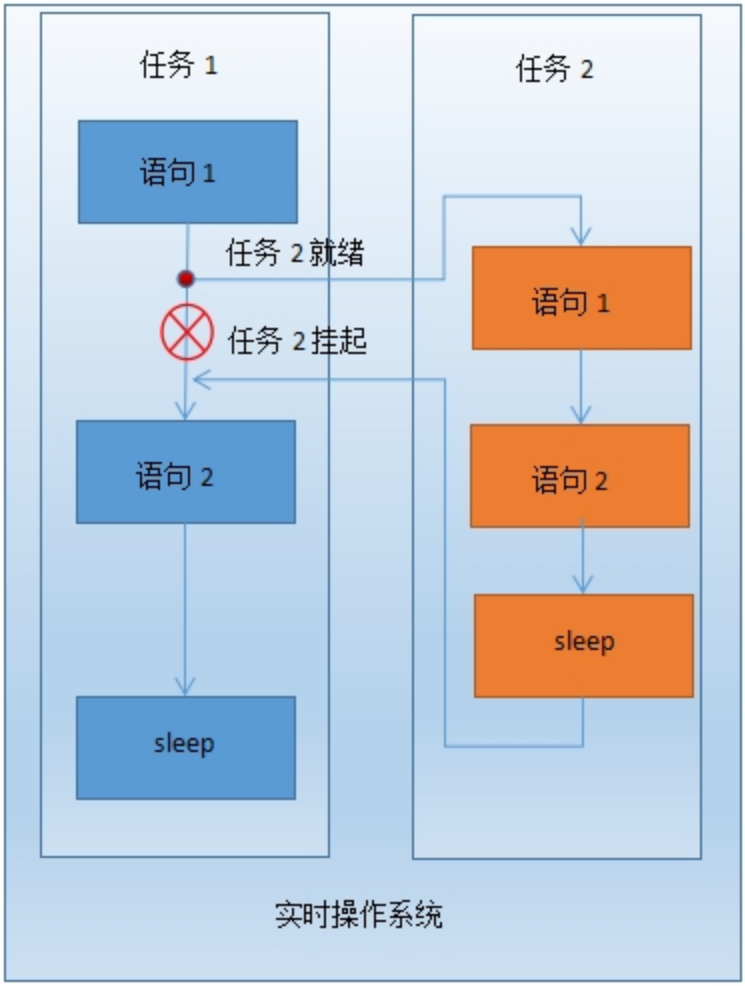
2）多级中断机制

一个实时应用系统通常需要处理多种外部信息或事件，但处理的紧迫程度有轻重缓急之分。有的必须立即作出反应，有的则可以延后处理。因此，需要建立多级中断嵌套处理机

制，以确保对紧迫程度较高的实时事件进行及时响应和处理。

3）实时调度机制

实时操作系统不仅要及时响应实时事件中断，同时也要及时调度运行实时任务。但是，处理机调度并不能随心所欲的进行，因为涉及到两个进程之间的切换，只能在确保“安全切换”的时间点上进行，实时调度机制包括两个方面，一是在调度策略和算法上保证优先调度实时任务；二是建立更多“安全切换”时间点，保证及时调度实时任务

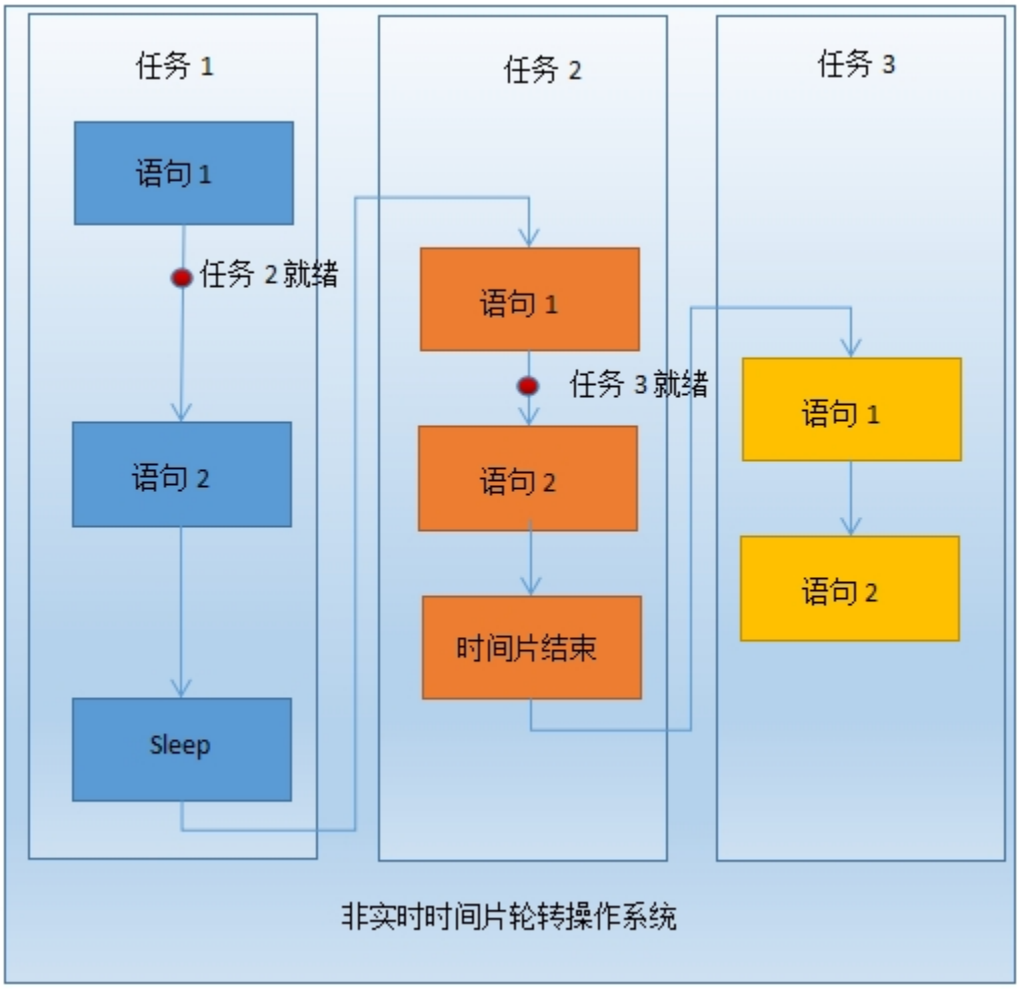


非实时系统：

早期的嵌入式系统中通常把嵌入式程序分成两部分，即前台程序和后台程序。前台程序通过中段来处理事件，其结构一般为无限循环；后台程序则掌管整个嵌入式系统软、硬件资源的分配、管理以及任务的调度，是一个系统管理调度程序。即前后台系统。一 般情况

下，后台程序也叫任务级程序，前台程序也叫事件处理级程序。在程序运行时，后台程序检查每个任务是否具备运行条件，通过一定的调度算法来完成相应的操作。对于实时性要求特别严格的操作通常由中断来完成，仅在中断服务程序中标记事件的发生，不再做任何工作就退出中断，经过后台程序的调度，转由前台程序 完成事件的处理，这样就不会造成在中断服务程序中处理费时的事件而影响后续和其他中断。

实际上，前后台系统的实时性比预计的要差。这是因为前后台系统认为所有的任务具有相同的优先级别，即是平等的，而且任务的执行又是通过FIFO队列排队，因而对那些实时性要求高的任务不可能立刻得到处理。另外，由于前台程序是一个无限循环的结构，一旦在这个循环体中正在处理的任务崩溃，使得整个任务队列中的其他任务得不到机会被处理，从而造成整个系统的崩溃。由于这类系统结构简单，几乎不需要RAM/ROM的额外开销，因而在简单的嵌入式应用被广泛使用。



实时系统与非实时系统：

1. 任务调度原则不同：

   前提Task2 优先级大于Task1且Task1先准备就绪并且已经开始运行
   非实时系统：当Task2准备就绪时，不会马上切换到Task2，要等待Task1的时间片结束或者Task1主动挂起后，Task2才开始运行，然后一直运行，直到结束，系统才会再次给低优先级的Task1分配时间片。
   实时系统：当Task2准备就绪的那一刻开始，Task1直接就被内核挂起，Task2开始执行，直到结束，系统才会再次给低优先级的Task1分配时间片

2. 任务调度的时间不同：

   非实时系统：任务调度不是严格实时的，如linux调度时间的最小单位为10ms，windows系统时间片也只是ms级别。

   实时系统：调度时间是us级的，而且一般小于10us。

3. 任务调度算法不同：

   实时系统：包含专有的任务调度算法，而且这也是实时系统的核心所在。

   非实时系统：无论是linux还是windows，都缺乏有效的实时任务的调度机制和调度算法。

实时任务与非实时任务

实时任务	非实时任务
实时任务是有时间限制的任务。	非实时任务与时间限制无关。
实时任务可以表示为时间的定量表达。	非实时任务不能用时间的函数来表达。
实时任务有两种类型——硬的和软的。	非实时任务没有进一步分类。
实时任务的截止日期以秒为单位。	非实时任务的截止日期可能是几分钟、几小时甚至几天。
大多数交互式任务是实时任务。	非实时任务包括几十年前使用的一些旧工作。
实时任务广泛用于计算机系统。	非实时任务现在不在计算机系统中使用。
实时任务由实时系统计算。	非实时任务是由传统系统计算的。
示例：卫星跟踪、视频会议等。	示例：批处理作业、旧电子邮件服务等。