第一章:

**嵌入式系统定义：**根据IEEE（国际电气和电子工程师协会）的定义：嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置”。

一般定义：以应用为中心、以计算机技术为基础、软件硬件可裁剪、功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

微机学会的定义：嵌入式系统是以嵌入式应用为目的的计算机系统。

**嵌入式系统含义：**“嵌入式系统”一般指非PC系统，有计算机功能但又不称之为计算机的设备或器材。它是以应用为中心，软硬件可缩扩的，适应应用系统对功能、可靠性、成本、体积、功耗等综合性严格要求的专用计算机系统。

**构成：**主要由嵌入式处理器、相关支撑硬件、嵌入式操作系统及应用软件系统等组成

**与个人计算机区别：** 嵌入式系统一般是专用系统，而PC是通用计算平台；嵌入式系统一般是专用系统，而PC是通用计算平台；嵌入式系统软件故障带来的后果比PC机大得多；嵌入式系统一般采用实时操作系统；嵌入式系统大都有成本、功耗的要求；嵌入式系统得到多种微处理体系的支持；嵌入式系统需要专用的开发工具。

**硬实时含义：**是指系统要确保在最坏情况下的服务时间，对于事件响应时间的截止期限必须得到满足。

**软实时含义：**其时限是一个柔性灵活的，可以容忍偶然的超时错误。失败造成的后果并不严重。

第二章:

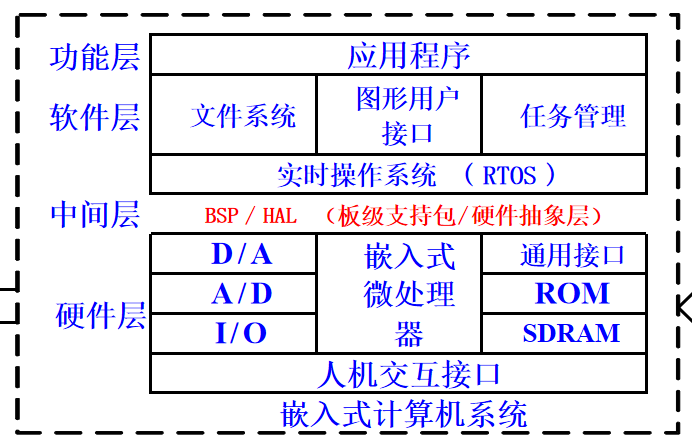
**嵌入式系统的总体结构：**主要包括：硬件层、中间层、系统软件层和应用层4个部分

嵌入式硬件主要包括提供嵌入式计算机正常运行的最小系统（如电源、系统时钟、复位电路、存储器等）、通用I/O口和一些外设及其它设备。

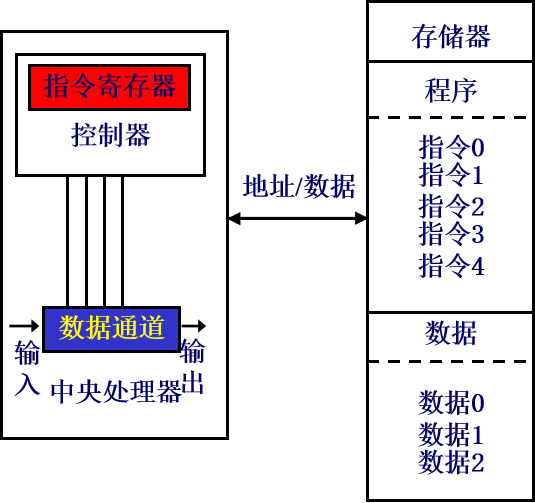
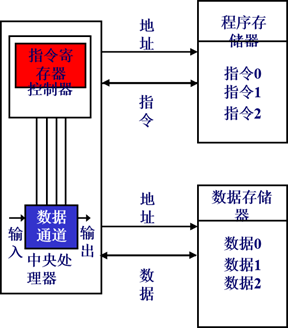
嵌入式系统中间层又称嵌入式硬件抽象层，如硬件驱动程序、系统启动软件等。是以嵌入式处理器为核心的

嵌入式系统软件层为应用层提供系统服务，如操作系统、文件系统、图形用户接口等。

功能层主要是用户应用程序。是嵌入式软件的核心部分

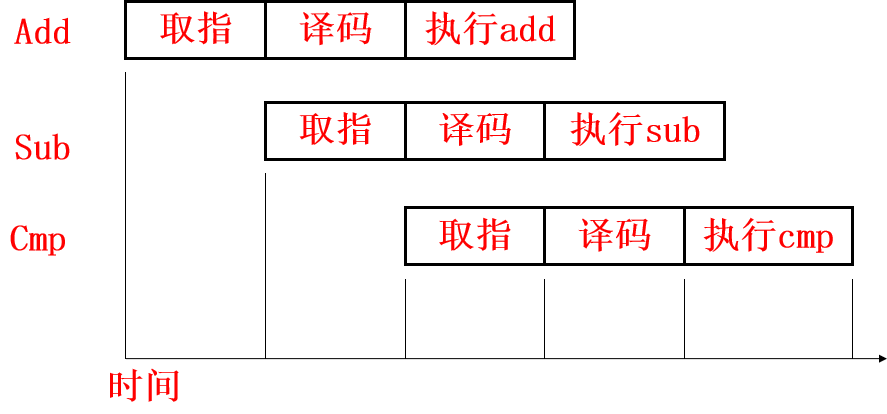


**冯诺伊曼体系结构：**冯·诺伊曼结构也称普林斯顿结构，是一种将程序指令存储器和数据存储器合并在一起的存储器结构。处理器，经由同一个总线传输来访问程序和数据存储器，程序指令和数据的宽度相同。

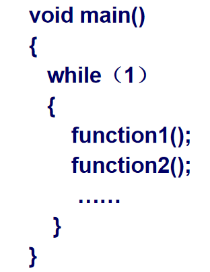
****

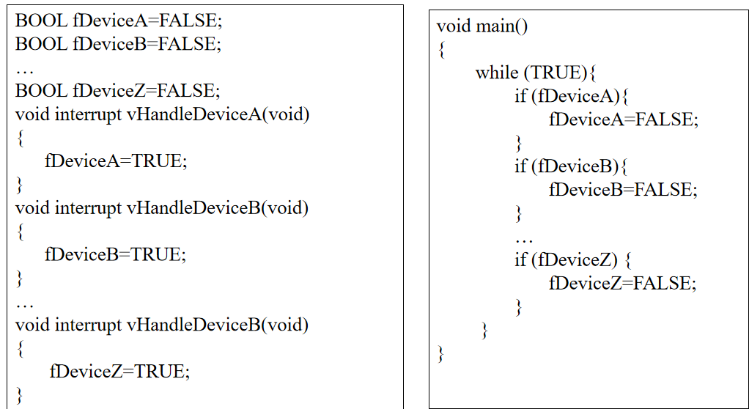
**哈佛体系结构：**哈佛结构是一种将程序指令存储和数据存储分开的存储器结构，目的是为了减轻程序运行时的访存瓶颈。哈佛结构的微处理器通常具有较高的执行效率。

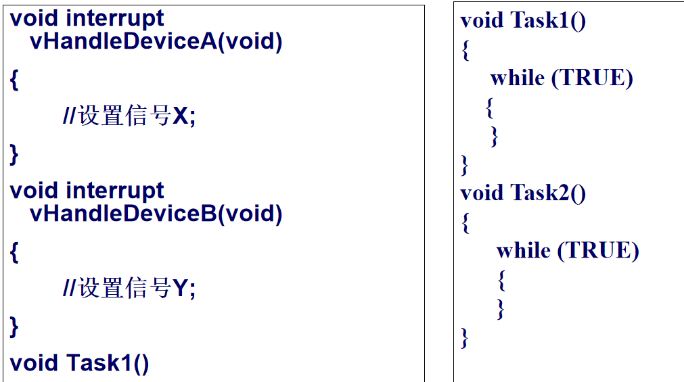
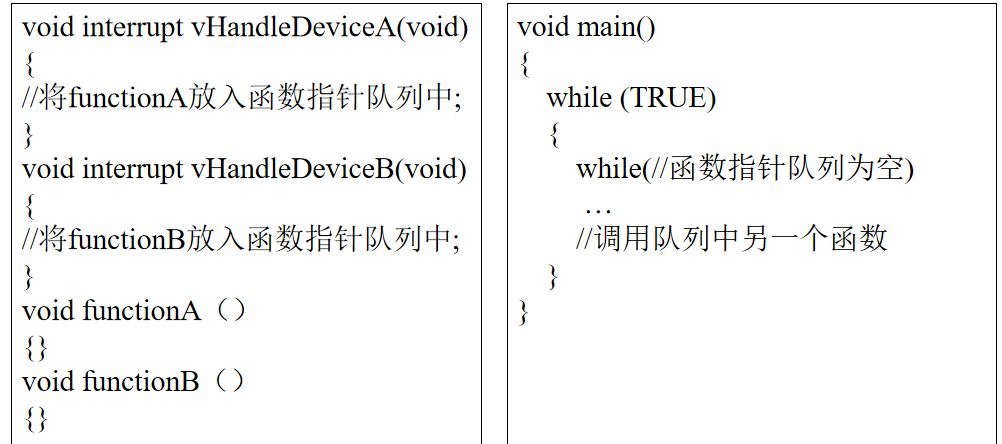
**CISC：**复杂指令集（Complex Instruction Set Computer），具有大量的指令和寻址方式，那么就需要更多的解释器。8/2原则：80%的程序只使用20%的指令。大多数程序只使用少量的指令就能够运行。

**** **特点：**指令格式不固定，指令长度不一致，操作数可多可少；寻址方式复杂多样，以利于程序的编写；采用微程序结构，执行每条指令均需完成一个微指令序列；每条指令需要若干个机器周期才能完成，指令越复杂，花费的机器周期越多。**RISC：**精简指令集（Reduced Instruction Set Computer)，指令数目少，在通道中只包含最有用的指令；执行时间短，确保数据通道快速执行每一条指令；使CPU硬件结构设计变得更为简单；每条指令都采用标准字长**。CISC与RISC的区别：**  从硬件角度来看CISC处理的是不等长指令集，它必须对不等长指令进行分割，因此在执行单一指令的时候需要进行较多的处理工作。而RISC执行的是等长精简指令集，CPU在执行指令的时候速度较快且性能稳定。 从软件角度来看，大型操作系统较适合运行在支持CISC的处理器上。是我们所熟识的DOS、Windows操作系统。而实时操作系统大多运行在支持RISC的处理器上。 **流水线的指令执行方式：**

**四种嵌入式系统软件的体系结构：**四种软件结构：轮转结构（round-robin）、带中断的轮转结构、函数队列调度（function-queue scheduling）结构和实时操作系统（real-time operating system）结构

 **轮转结构（round-robin）：**是能想象得到的、最简单的一种结构。该结构中不存在中断，主循环只是简单地依次执行函数调用。 **优点：**简单，它没有中断，没有共享数据，无须考虑延迟时间 **缺点：**如果一个设备需要比微处理器在最坏情况下完成一个循环的时间更短的响应时间，那么这个系统将无法工作。即使所要求的响应时间不是绝对的截止时间，当有冗长的处理时系统也会工作得不好。这种结构很脆弱。 轮转结构可能仅仅适用于非常简单的装置，如数字手表和微波炉等。

 **带中断的轮转结构：**中断程序处理硬件特别紧急的需求，然后设置标志，主循环轮询这些标志，然后根据这些需求进行后续的处理。 **优点：**与轮询结构相比这种结构可对优先级进行更多的控制。中断程序可以获得很快的响应，因为硬件的中断信号会使位处理器停止正在main函数中执行的任何操作，而转去执行中断程序。 **缺点：**中断程序与主程序中的数据共享问题，当正在执行的主程序正在处理共享数据时，被中断程序中断，进而处理中断程序，在中断程序中有可能又对共享数据进行了相应的操作，从而导致回到主程序时，共享数据的值已经发生了改变，导致意想不到的结果。

  **函数队列调度结构：**中断程序在一个函数指针中添加一个函数指针，以供main函数调用。主程序仅需要从该队列中读取相应的指针并且调用相关的函数。 **优点:**main可以根据任何可以达到目标的优先级方案来调用函数，这样任何需要更快响应的任务代码都可以被更早执行。为了做到这一点，只需要在对函数指针进行排队的程序中对代码进行一点技巧性设计。 **缺点：**具有较低优先级任务代码的函数可能会有更差的响应。如果某个较低优先级任务的代码函数过长，就有可能影响较高优先级函数的响应时间。 **实时操作系统结构：**是在函数队列调度结构基础上发展而来的一种结构。 **和以前那些结构的不同之处在于：**中断程序和任务代码之间的必要信号发送是由实时操作系统处理的，而并不需要使用共享变量来达到这个目标。在代码中并没有用循环来决定下一步要做什么。实时操作系统内部的代码可以决定什么任务代码函数可以运行。实时操作系统可以在一个任务代码程序运行期间将其挂起，以便运行另一个程序。 前两点主要针对编程的方便性；而最后一点是实质性的：使用实时操作系统结构的系统不仅可以控制任务代码的响应时间，还可以控制中断程序的响应时间。

**不同软件结构的特点：**在轮询机构和函数队列调度结构中，一个任务代码函数的响应时间，取决于包括低优先级任务子程序在内的各个任务代码子程序的长度。当改变任意一个子程序的时候，就有可能改变了整个系统的响应时间。而在实时操作系统结构中，对于较低优先级函数的改变通常不会影响较高优先级函数的响应时间。

**软件结构的选择：**选择可以满足响应时间需求的最简单的结构。如果系统对于响应时间的要求很高使得一个实时操作系统称为必须的，那就应该使用实时操作系统结构。如果对一个系统有意义的话，可以将这几种结构结合起来使用。第四章：

**arm编号含义：** Advanced RISC Machines的缩写。A是高端应用芯片，R系列一般用在高实时性的高端工业，医疗器械，路由交换机等。M系列就一般用在一般得工业。【以前,ARM使用一种基于数字的命名法。在早期(20世纪90年代),还在数字后面添字母后缀,用来进一步明确该处理器支持的特性。】

**arm指令集：**是一个32位精简指令集，是指计算机ARM操作指令系统

**Thumb 指令集:** 32位ARM指令集的子集，按16位指令重新编码。Thumb是ARM体系结构中一种16位的指令集。Thumb指令集可以看作是ARM指令压缩形式的子集，它是为减小代码量而提出，具有16bit的代码密度。

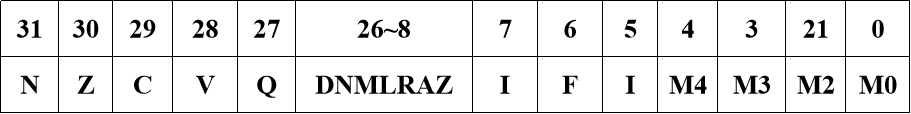
特点：THUMB代码所需空间为ARM代码的70%；THUMB代码所使用的指令数比ARM代码多40%；用32位存储器，ARM代码比THUMB代码快40%；用16位存储器，THUMB代码比ARM代码快45%；使用THUMB代码，外部存储器功耗比ARM代码少30%

在Thumb模式下，较小的指令码有更少的功能性。

Thumb指令集没有协处理器指令、信号量指令、以及访问CPSR或SPSR的指令，没有乘加指令及64位乘法指令等，且指令的第二操作数受到限制。

大多数的Thumb数据处理指令采用2地址格式。

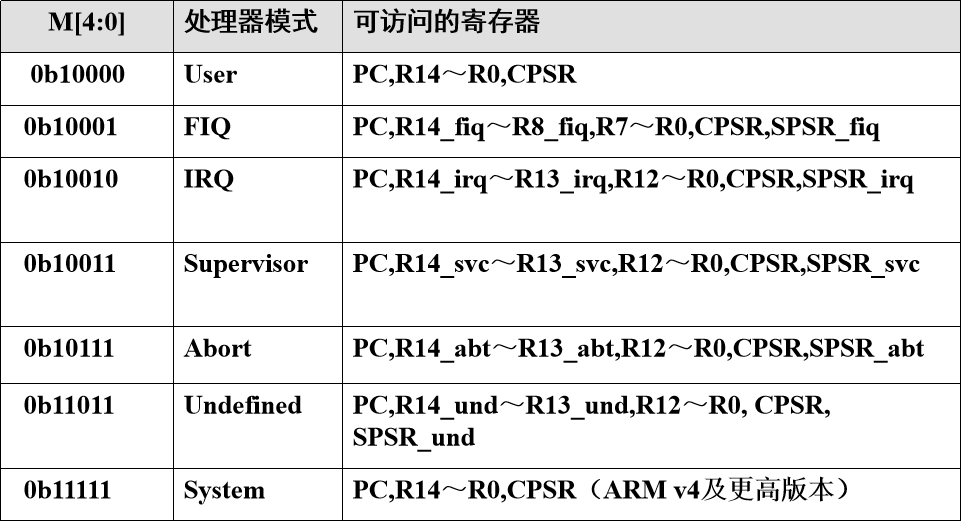
**Thumb指令与ARM指令的区别：**分支指令:跳转的范围小,除B指令外,都是无条件跳转；数据处理指令:Thumb指令只有2个操作数,而ARM指令是3个操作数；单寄存器加载存储指令:Thumb指令只能访问R0~R7；多寄存器加载存储指令:Thumb指令只能访问R0~R7的子集；Thumb特有指令:PUSH和POP作用于R13。**7种工作模式：**用户模式（User,usr），快速中断模式（FIQ,fiq），外部中断模式（IRQ,irq），特权模式（Supervisor,sve），数据访问中止模式（Abort,abt），未定义指令中止模式（Undefined,und），系统模式（System,sys）。 除了用户模式以外，其他6种处理器模式可以称为特权模式，在这些模式下，程序可以访问所有的系统资源，也可以任意地进行处理器模式的切换。其中除了系统模式外的其他5种特权模式又称为异常模式。

**状态寄存器（PSR）的每一位的含义：**

条件标志位：N（Negative）：当两个补码表示的有符号整数运算时，N＝1表示运算的结果为负数，N＝0 表示结果为正数或零。 Z（Zero）：Z＝1表示运算结果是0，Z＝0表示运算结果不是零，对于CMP指令，Z＝1表示进行比较的两个数大小相等。 C（Carry）：在加法指令中（包括比较指令CMN），结果产生进位了，则C＝1，表示无符号数运算发生上溢出，其他情况下C＝0；在减法指令中（包括比较指令CMP），结果产生借位了，则C＝0，表示无符号数运算发生下溢出，其他情况下C＝1；对于包含移位操作的非加/减法运算指令，C中包含最后一次被溢出的位的数值，对于其他非加/减法运算指令，C位的值通常不受影响。 V（oVerflow）：对于加/减法运算指令，当操作数和运算结果为二进制的补码表示的带符号数时，V＝1表示符号位溢出,其他的指令通常不影响V位。 Q标志位：CPSR（当前程序状态寄存器）的bit[27]称为Q标志位，主要用于指示增强的DSP指令是否发生了溢出，同样的，SPSR（备份程序状态寄存器）的bit[27]也称为Q标志位，用于在异常中断发生时保存和恢复CPSR中的Q标志位。

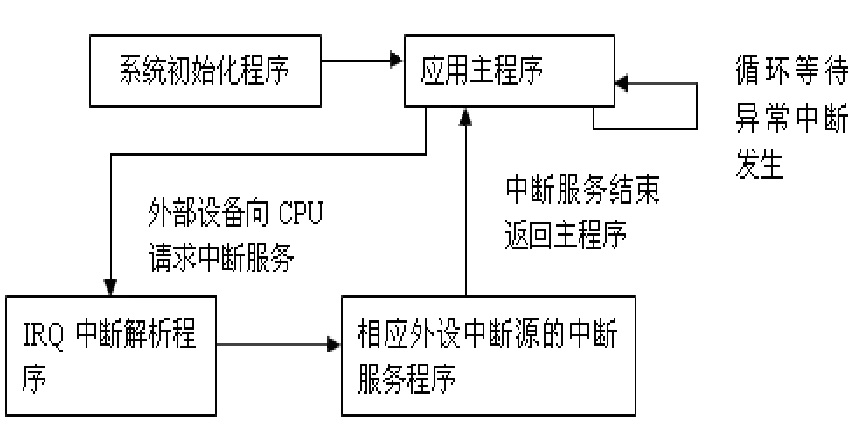
CPSR中的控制位：CPSR的低8位I、F、T及M[4：0]统称为控制位，当异常中断发生时这些位发生变化。在特权级的处理器模式下，软件可以修改这些控制位。 中断禁止位：当I＝1时禁止IRQ中断；当F＝1时禁止FIQ中断；通常一旦进入中断服务程序可以通过置位I和F来禁止中断，但是在本中断服务程序退出前必须恢复原来I、F位的值。 T控制位：用来控制指令执行的状态，即说明本指令是ARM指令还是Thumb指令。对于不同版本的ARM处理器，T控制位的含义是有些不同的。对于ARM v4及更高版本的T系列处理器，T控制位含义如下： 当T＝0，表示执行ARM指令；当T＝1，表示执行Thumb指令。对于ARM v5及更高版本的非T系列处理器：当T＝0，表示执行ARM指令；当T＝1，表示强制下一条执行的指令产生未定义指令中断。 M 控制位：M[4:0]称为处理器模式标识位；

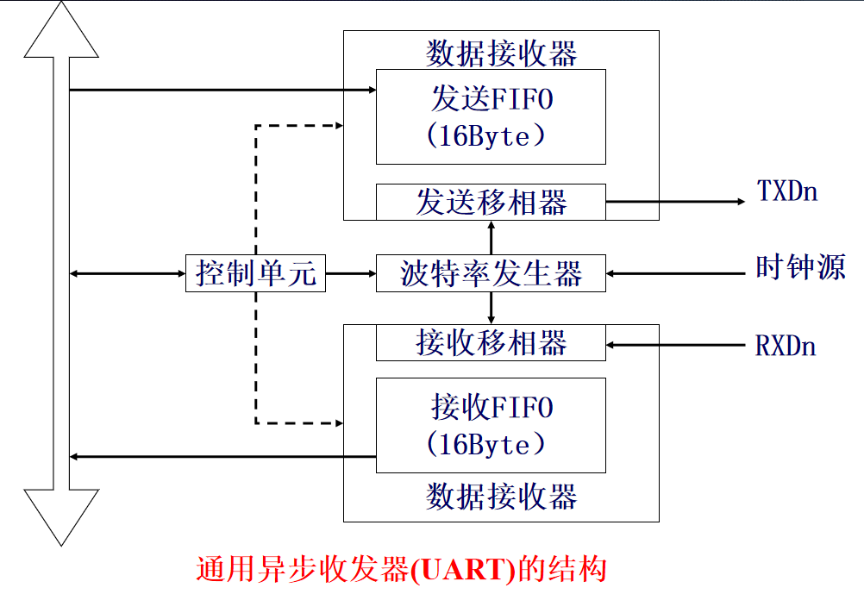
CPSR的其他位应用于将来ARM版本的扩展，程序可以先不操作这些位。



**arm处理器响应过程：**ARM微处理器在进入异常处理程序之前会执行以下几步操作：（这些是中断发生时，处理器硬件自动处理的）1、将下一条指令的地址存入相应连接寄存器LR，以便程序在处理异常返回时能从正确的位置重新开始执行。 2、将CPSR复制到相应的SPSR中。 3、根据异常类型，强制设置CPSR的运行模式位。 4、强制PC从相关的异常向量地址取下一条指令执行，从而跳转到相应的异常处理程序处。 **返回过程：**1、将链接寄存器LR（R14\_irq）的值减去相应的偏移量后送到PC中。 SUBS PC，LR\_irq，#4 2、将SPSR（SPSR\_irq）复制回CPSR中。 3、若在进入异常处理时设置了中断禁止位，要在此清除。 **（P76）**

**arm中断编程机制：**

****

**bootloader 的两种模式：**启动加载（Boot loading）模式：这种模式也称为“自主”（Autonomous）模式。也即 Boot Loader 从目标机上的某个固态存储设备上将操作系统加载到 RAM 中运行，整个过程并没有用户的介入。应用场合：系统正常工作模式。

下载（Downloading）模式：目标机上的 Boot Loader 将通过串口连接或网络连接等通信手段从主机（Host）下载文件。 应用场合： 第一次安装内核与根文件系统时被使用；系统更新或升级。**arm启动的一般步骤：（包含两种不同的操作模式：“启动加载”模式和“下载”模式。）**设置中断向量表；初始化存储设备；初始化堆栈；初始化用户执行环境；呼叫主应用程序。第五章:

**异步串行通信：** 异步串行方式是将传输数据的每个字符一位接一位(例如先低位、后高位)地传送。

数据的各不同位可以分时使用同一传输通道，因此串行I/O可以减少信号连线，最少用一对线即可进行。

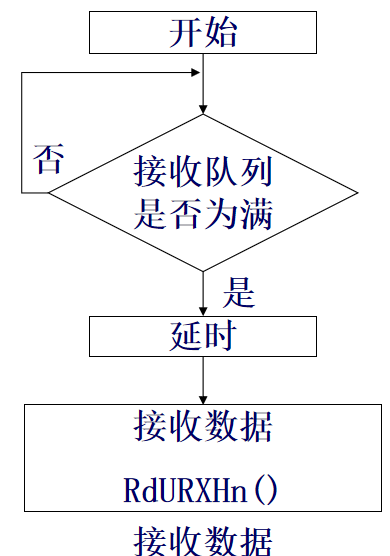
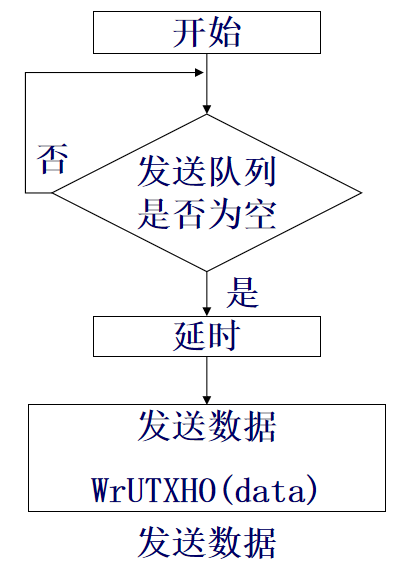
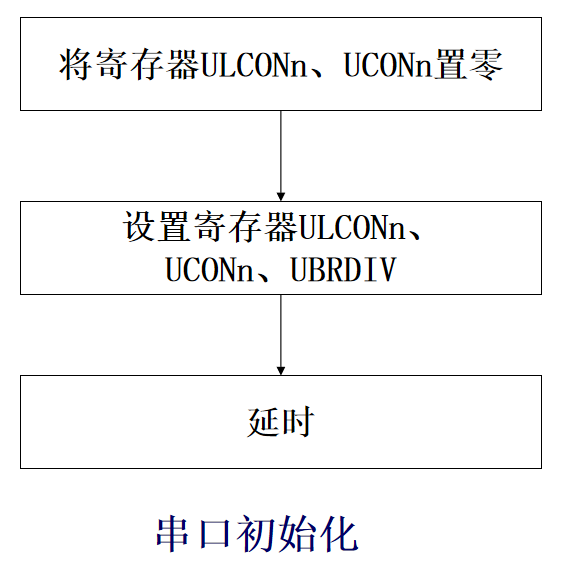
微机异步串行通信中，常用的波特率为50，95，110，150，300，600，1200，2400，4800，9600 ,115200等。

接收方按约定的格式接收数据，并进行检查，一般可以查出以下三种错误：奇偶错，帧格式错，溢出错

**（P101-P106）**

volatile关键字是一种类型修饰符，用它声明的类型变量表示可以被某些编译器未知的因素更改，比如：操作系统、硬件或者其它线程等。

volatile用在如下的几个地方： 中断服务程序中修改的供其它程序检测的变量需 要加volatile；多任务环境下各任务间共享的标志应该加volatile；存储器映射的硬件寄存器通常也要加volatile说明，因为每次对它的读写都可能由不同意义； 串口初始化包括：设置波特率；设置起始位、奇偶校验位等；设置中断等 发送数据主要是将需发送的数据放入发送数据寄存器。

接收数据主要是将已到达的数据从接受寄存器中取出

**IIC（I2C ，Inter-IC)总线：**Philips 公司20多年前开发的两线制串行总线；外部设备IC芯片可以使用简单的通信硬件进行相互通信；数据传送率可达100kbits/s，且提供7位寻址位；在快速模式下，可达3.4 Mbits/s 以及10位寻址；许多IC芯片支持该总线接口。 **（P138-P147）**

第六章:

**操作系统的结构：**驱动程序；内核；接口库；外围

**操作系统的五大管理功能：**进程管理、文件管理、存储管理、设备管理和作业管理。

**操作系统分类：**分时操作系统（特点：多路性，独立性，交互性，及时性）；

实时系统（硬实时系统主要用于工业生产的过程控制、较大系统的跟踪和控制、武器的制导等响应速度十分快、工作极其安全可靠的场合。

软实时系统主要应用于对响应的速度要求不象硬实时系统那么高，且时限要求也不那么严密的信息咨询和事务处理领域。）；

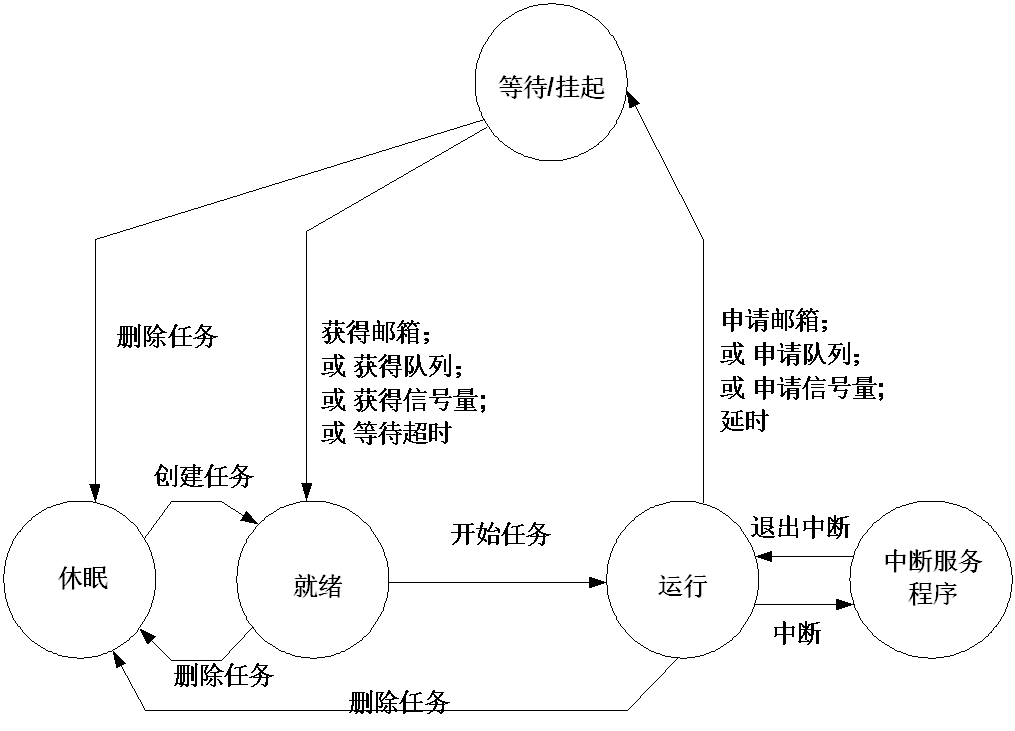
网络操作系统（常见的网络操作系统主要有 UNIX、NOVELL、WINDOWS NT、Netware等）；

分布式操作系统(Distributed Operating System，简称DOS)（分布式操作系统有：Amoeba、Mach、Chorus、DEC等。）；

多处理器操作系统（多处理机操作系统一般分为主从式和对称式。）**嵌入式操作系统特点：**实时性；小内核；可剪裁、可配置；易移植；高可靠性；低功耗。

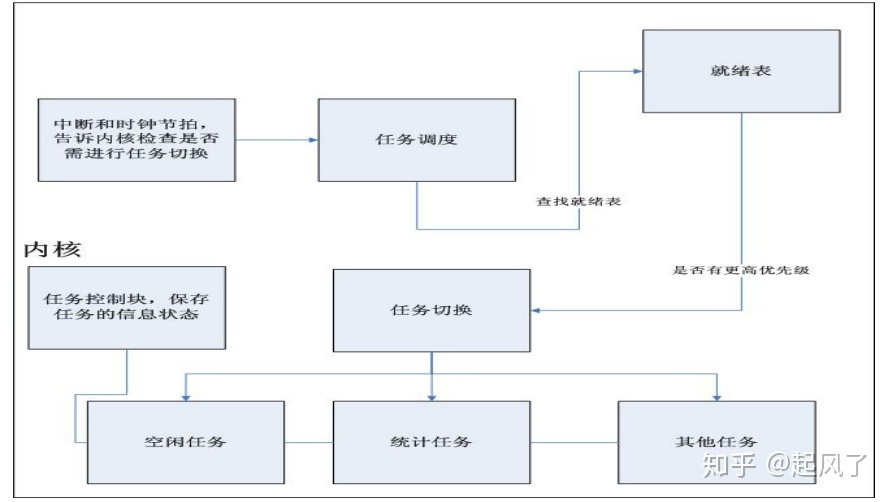
**任务包含的几种状态：**休眠（DORMANT）；就绪(READY)；挂起或等待(WAITING)；运行(RUNNING)

**任务转换、状态之间转换关系：**

****

第七章:

**uC/OS - i i内核:** 内核的主要功能是任务的调度和切换。任务调度，任务切换，中断与时钟节拍，临界段代码的处理，统计任务和空闲任务。



**任务控制块TCB:** （Task Control Block）是用来实现任务管理的数据结构。

在内核中每个任务对应一个TCB，所有任务的TCB构成一个空闲任务控制块列表。

**（P170）**

任务控制块 OS\_TCB是一个数据结构，保存该任务的 相关参数，包括任务堆栈指针，状态，优先级，任务 表位置，任务链表指针等。

一旦任务建立了，任务控制块OS\_TCBs将被赋值。

所有的任务控制块分为两条链表，空闲链表和使用链表。**优先级：**OSTCBPrio任务优先级。高级优先任务的OSTCBPrio值小。

uC/OS-II采用的是优先级抢占式规则，即在系统中创建任务时要按照其任务的重要性分配一个唯一的优先级级别。是一个实时、多任务的操作系统。

在uC/OS-II中每个任务有唯一的优先级，因此任务的优先级也是任务的唯一编号(ID)，可以作为任务的唯一标识。内核可用控制块优先级表OSTCBPrioTbl[]由任务的优先级查到任务控制块的地址。

uC/OS-II主要就是利用任务控制快OS\_TCB、就绪表(ready list)和控制块优先级表OSTCBPrioTbl[]来进行任务调度的。任务的优先级别最多有64级。

在uC/OS-II的任务个数和最低优先级都可以在配置文件OS\_CFG.H中用OS\_LOWEST\_PRIO来配置，任务的总数也不能超过OS\_LOWEST\_PRIO+1个。系统总是把最低优先级别OS\_LOWEST\_PRIO自动赋给空闲任务；如果我们还使能了统计任务，则系统还会把优先级别OS\_LOWEST\_PRIO-1自动赋给统计任务。0~OS\_LOWEST\_PRIO-2优先级可以赋给用户任务。 由于在uC/OS-II中每个任务都具有唯一的优先级别，因此这个优先级别也可以作为任务在系统中的标识identification

**存储变量：**局部变量—— 栈区；局部静态变量——静态区；全局变量——静态区的常量区；全局静态变量——静态区

变量在内存中存储方式影响变量在内存中的生命周期。

全局变量和局部变量的区别：生命周期不同：局部变量是所在的函数结束之后释放，全局变量是所在的程序结束之后释放；存储空间不同；全局变量的初始值为0，局部变量的初始值为随机赋予的；当全局变量和局部变量重名的时候，局部变量起作用。