2010 年 9 月
第 29 卷　第 9 期

实时多任务机制在单片机系统中的研究

王瑜瑜　范蟠果　秦　旭
（西北工业大学自动化学院　西安　710129）

摘　要：从操作系统实现多任务机制的原理入手，首先介绍了单片机实时多任务机制的实现方法，并以 MCS-51 系列单片机为例提出了一种新型的程序结构。然后提出了该机制的 2 种实现策略：基于定时中断和外部中断，详细阐述 2 种实现策略的实现原理和方法并对二者进行了比较。最后通过实验设计一个 MCS-51 单片机的应用系统验证了这种结构模块清晰、调度灵活。

关键词：多任务机制；单片机系统；中断；任务调度
中图分类号：TP368.1　　**文献标识码**：A

Research of real-time multi-task mechanism in the single-chip microcomputer system

Wang Yuyu　Fan Panguo　Qin Xu
(School of Automation, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: From the principle of the operating system realizing multi-task mechanism, this paper firstly introduces the realizing method of microcontroller multi-task mechanism and presents a new type of program structure taking MCS-51 series microcontroller for example. Then it provides two kinds of strategies for this mechanism: based on timing interrupt and external interrupt, expounds the principles and implementation methods of the two realizing strategies and compares the both. Finally test shows that this structure module is clear and scheduling is flexible.

Keywords: multi-task mechanism; single-chip microcomputer system; interrupt; task scheduling

0　引　言

传统的单片机程序一般采用单任务机制，单任务系统具有简单直观、易于控制的优点。然而由于程序只能按顺序依次执行，缺乏灵活性，只能使用中断函数实时地处理一些较短的任务，在较复杂的应用中使用极为不便。所以将实时多任务机制引入单片机系统设计显得十分必要。多任务机制是现代操作系统才具有的突出优点。将多任务机制引入单片机系统，可以大大提高现有单片机系统的工作效率，满足多任务要求。在这种机制下，CPU 的运行时间被划分为许多小的时间片，由某种调度算法按不同优先级别分配给不同的应用程序。多个应用程序分别在自已的时间片内访问 CPU，从而造成微观上轮流运行，宏观上并发运行的多任务效果。时间片的划分借助于中断机制来实现，而通用的单片机中都允许使用中断，因此在单片机系统中实现多任务机制是切实可行的，但单片机的运行能力决定了调度算法必须简单明了。多任务机制是现代操作系统才具有的突出优点。在这种机制下，CPU 时间按某种调度算法分配给不同的任务。常见的单片机上的多任务机制一般采用时间片轮换的方式，根据系统时钟将 CPU 运行时间划分为若干时间片，由系统调度程序分给各个任务模块。这种方式比较适合于实时性要求比较高的场合。而多任务恰恰是现代操作系统的突出特点，将多任务机制引入单片机系统，可以大大提高现有单片机系统的工作效率，满足多任务要求。种机制下，CPU 时间按某种调度算法分配给不同的任务。

1　多任务机制在单片机上的实现原理

所谓实时多任务，计算机并不是真的在给定时间段内运行多个程序，而是按时间片在各个程序间飞快地切换，各个程序轮流占用 CPU，由于切换时间太短，所以就有了在同一时间内运行多个程序的错觉[1]。就操作系统的特点来看，多任务系统不外乎 2 种方式：协同式和抢占式。在一个协同式多任务系统中要求每个任务的具有不同的优先级，当有比当前任务优先级高的任务就绪时一定会产生任务切换，最大限度提升了系统的实时性，多用于实时

作者简介：王瑜瑜，硕士研究生，主要研究方向为计算机控制和自动化理论。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

性较高的测控系统。在抢占式多任务系统中，操作系统必须具有从任何一个运行的程序上取走控制权和使另一个程序获得控制权的能力。一个抢占式系统下运行的应用程序不用担心独占系统，因为系统会合理地给每个运行的任务分配时间片。将这种系统称做抢占式系统是因为它不管在所给时间块内会发生什么事，都抢占运行程序的控制权。这样做使得任务的独立性较好，但是实时性就会相应下降，常用在实时性不高但能满足要求的测控系统。单片机本身是单任务工作的，即程序只能按单一的线索顺序执行，一个任务完成之后（除非中断），才能进行下一个任务。但是，由于现代单片机绝大多数都提供了 2 个甚至多个定时/计数器，而且都具备定时中断功能，如果定时器、系统晶振和分频系数配合得当，那么单个计时器可以进行最短不到 1 μs，最长几十 ms 的定时中断。如果把定时中断用作时间片的分配，就可以实现多个任务分时轮换执行。同时，现代很多单片机的通用 I/O 口都具有中断的功能，以 51 单片机为例，其内提供了 T0 与 T1 两个 16 位的定时/计数器，若系统晶振为 12 MHz，那么计时器最长可以计时 65.536 ms，最短为 1 μs。也即通过对定时器工作模式进行设置，可以进行 1 μs ~ 65.536 ms 的定时中断。如果把定时中断用作时间片的分配，完全可以实现多个任务分时轮换执行[2]。同时，单片机还提供 INT0 与 INT1 两个外部中断（可通过 8259 扩充为多级中断）和 P0、P1、P2 三个可编程的 I/O 端口，与定时器配合下，完全可以实现具有优先权判别的多任务控制。

2　多任务机制的实现方法

利用单片机具有的内部时钟中断，可以方便地实现多任务机制。下面以常用的 MCS-51 系列单片机为例，说明其实现方法。

2.1　划分时间片

由多任务机制的特点，必须提供一个系统时钟，根据系统时钟将 CPU 的运行时间划分为若干时间片，系统调度程序将它分给各个子任务。可由 51 系统单片机内带的定时器完成这一工作，定时器频率应设为各个子任务要求执行频率的整数倍[3]。

2.2　构造任务调度表

为了实现多任务，还需要构造任务调度表并维护一个系统任务指针。该表的表项数等于需要调度的子任务数，每个表项占 4 个字节，描述个任务模块的入口地址（2 个字节）和访问计数器（2 个字节）。任务调度表反映了各任务的执行顺序和优先级别。系统任务指针为 1 个字节数，指向系统当前正在执行的任务。由任务指针和任务调度表表头可以找到相应的任务程序入口地址。

2.3　任务调度程序

多任务机制的具体实现由任务调度程序来完成。调度程序在每个时钟片的开始取得控制，它将系统任务指针加 4，并指向下一任务模块，根据系统任务指针查阅任务调度表，将任务的访问计数器减 1，判断其是否为 0。不为零则直接返回，为零则将当前任务的入口地址读入 DPTR 寄存器，用 JMP @ A + DPTR 将指令转而执行相应的任务模块。任务执行完毕将访问计数器置为初值，供下次使用。任务指针到达调度表末尾后，调度程序将重新指向调度表的开头，以开始下一调度周期。

3　多任务机制的实现策略

3.1　基于定时器分配时间片的实现策略

基于定时中断的实现机制，是现代单片机实时多任务机制的主要实现方式，现行大部分在单片机上运行的商业性实时多任务操作系统内核都是基于这一原理。基于时间片轮换的原理是用定时器中断产生时间片，然后为每一个任务分配一定数量的时间片，每次定时中断产生时，将正在执行的任务时间片数量减 1，当任务的时间片数量为 0 时，将 CPU 的控制权切换给其他任务，同时根据任务状态重新给上一次任务分配一定数量的时间片。系统一开始就初始化定时器，并为每一个任务分配一定数量的时间片和合适的堆栈区，一个时间片的长度定时器就会产生中断。假设为每个任务分配的时间片数为 $n_0, n_1, n_2, \cdots, n_i$，定时器的定时中断时间为 t_0，即时间片的长度，由于中断和初始化设置等需要消耗单片机时间，所以单纯使用定时器产生的时间片比理论值短，但这对于单片机内部的任务轮换来说已经足够了，所以第 i 个任务每次获得 CPU 的控制时间为 $n_i \times t_0$。其核心的程序实现过程包括：(1) 主程序，负责系统启动时各种软硬件的初始化工作；(2) 定时中断子程序，负责管理各个任务当前拥有的时间片数量，判断是否需要进行任务切换；(3) 任务切换子程序，保存当前运行任务的各种信息，轮询各个任务的状态，根据上一个任务的状态，决定是否为之分配时间片，并为下一个任务执行做好准备，最后将任务切换到下一个任务[4]。假设多个任务控制多个设备，可让单片机的 I/O 连接相应的设备。每个任务分配一定的时间片，在时间片内占用 CPU 进行运算和控制外部设备，其程序主流程如图 1 所示。

3.2　基于外部中断的实现策略

如果单片机外接较多的传感器，任务的切换需要考虑外部传感器的状态时，就需要通过单片机响应外部中断的方式切换任务。由于外部中断响应速度快，任务具有很高的实时性，但是，中断程序在其退出之前一直占据 CPU 的使用权，其他任务就得不到执行，所以不能实现多任务并发执行，影响整个系统的实时性。在此必须注意以下 2 点：(1) 外部中断优先级的处理。由于实际的微型控制系统中，单片机的外部设备比较固定，也即外部设备中断的优先级比较固定，完全可以用较简单的优先级表法实现优先级的确定。只要在内存中预置一片数据区，对应表示外部设备的中断优先级，那么就可以通过查表的方式获取优先级别，从而判断任务的切换方向。进一步可以通过程序

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

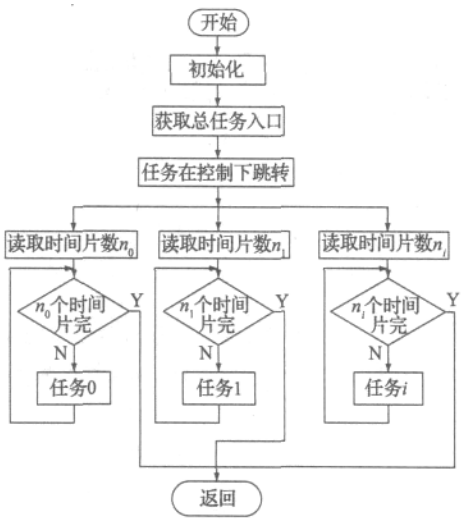

图 1　主程序流程

动态的修改长期等待和长期运行的任务对应的优先级，使所有的任务都有机会获得 CPU[5-6]。(2) 外部中断口的扩展。通常在传统的单片机系统中，特别是以 51 单片机为核心的测控系统中，只有 2 个外部中断，在实际智能测控系统中往往不够用，需要中断扩展芯片 8259A 来扩展外部中断。这样势必增加电路的成本和复杂度，考虑到很多单片机的 I/O 口并未全部利用，可以将剩余的端口作为外部传感器的状态输入端，同时将传感器的输出端通过反向器统一接在单片机的外部中断请求引脚，任何一个传感器都可以向单片机发中断请求。

3.3　2 种策略的比较

由于应用对象的复杂性和多变性，往往单一的实现机制不能满足测控系统的要求。在实际应用中，往往是多种机制的综合运用，针对不同的任务，采取不同的机制。基于定时中断的实现策略，各任务在任务调度程序控制下有条不紊的执行，每个任务在给定时间片内完全占有 CPU，可以完成既定子任务，同时又在时间片结束时让出 CPU，以便其他任务执行。比较适合于批处理系统和任务既定的控制系统。基于外部中断的实现策略，实时性很好，可以实现系统与外界交互，及时地调度相应的任务。但是，随着任务数量的增多，软件设计变得十分复杂。所以在实际的系统设计中，应充分考虑任务本身的特性，对于占用 CPU 时间较短，实时性要求特别高的任务，尽量使用外部中断来完成，对于任务数量较多，较复杂的系统尽量采用基于定时中断的策略。

4　具体应用实例

采用 51 单片机设计一个应用系统，该系统有 3 个通道的数据采集输入，一个通道每 10 ms 采集一次，另 2 个通道每 2 ms 一次。显示采用多端数码管动态显示方式，同时配有键盘接受用户命令[7]。

由上述实现方案，可如下设计各模块：

3 个输入模块，分别记为 N1、N2、N3 模块，其执行频率分别为 1 次/10 毫秒，1 次/2 毫秒和 1 次/2 毫秒。

显示模块记为 D 模块，每次显示一位，每两位显示之间需间隔一定时间，定为 2 ms，即显示模块的执行频率为 1 次/2 毫秒。

键盘管理模块记为 K 模块，键盘扫描的去抖时间定为 50 ms，键盘管理模块的执行频率为 1 次/50 毫秒。

此外系统还应有其他功能模块，由于它们对时间要求并不严格，同时对完成时间也不可预料，因此可放入主程序中去完成。考虑到 A/D 转换时间和输出执行时间，可以将基本的时间片定为 500 μs，统计各任务的调度情况及任务调度表如表 1 所示。

表 1　任务调度表

模块名	执行频率	模块访问计数器值(执行周期)		入口地址
		10 进制计数值	16 进制计数值	
N1	1 次/10 毫秒	200014H		0014H
N2	1 次/2 毫秒	40004H		0004H
N3	1 次/2 毫秒	40004H		0004H
D	1 次/2 毫秒	40004H		0004H
K	1 次/50 毫秒	1000064H		0064H

5　结束语

本文在单片机系统中引入了现代操作系统所具有的多任务机制，提出了在不采用操作系统的情况下实现单片机实时多任务的两种实现策略和一种新型的实现方法[8]。在现有的单片机系统中实现多任务机制，仅仅在升级软件和增加、更改少许硬件电路的条件下，就可以使控制系统支持多任务处理和控制，不仅升级成本低，而且大大延续系统的寿命和扩充系统的功能，具有较高的实用价值。

参 考 文 献

[1]　罗江，户永清. 51 单片机多任务机制的实现策略研究[J]. 四川文理学院学报：自然科学版，2008，18(2)：82-84.
[2]　陈劲松，程新明，魏忠. 时间片轮转算法在单片机程序设计中的应用[J]. 电子技术应用，2003(3)：18-19.
[3]　王辉堂，颜自勇，陈文. 一种基于 C51 的多任务机制及应用[J]. 电子设计应用，2006(8)：139-141.
[4]　汪滢，辛晓宁，袁德成. 实时多任务机制在低功耗单片机系统中的最小实现[J]. 化工自动化及仪表，2002，29(3)：70-72.
[5]　余进，黄继武. 多任务机制在单片机系统中的应用[J]. 电子技术应用，1999(3)：14-16.
[6]　田野. 单片机数据采集卡及其实施多任务机制的实现[J]. 自动化仪表，2002，23(6)：33-36.
[7]　张琪，刘莉，张训文. 基于单片机的多参数控制仪[J]. 仪器仪表学报，2006，27(6)：1917-1919.
[8]　潘永雄. 基于过程的单片机程序结构及实现方法[J]. 电子工程师，2005，31(3)：5-7.

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net