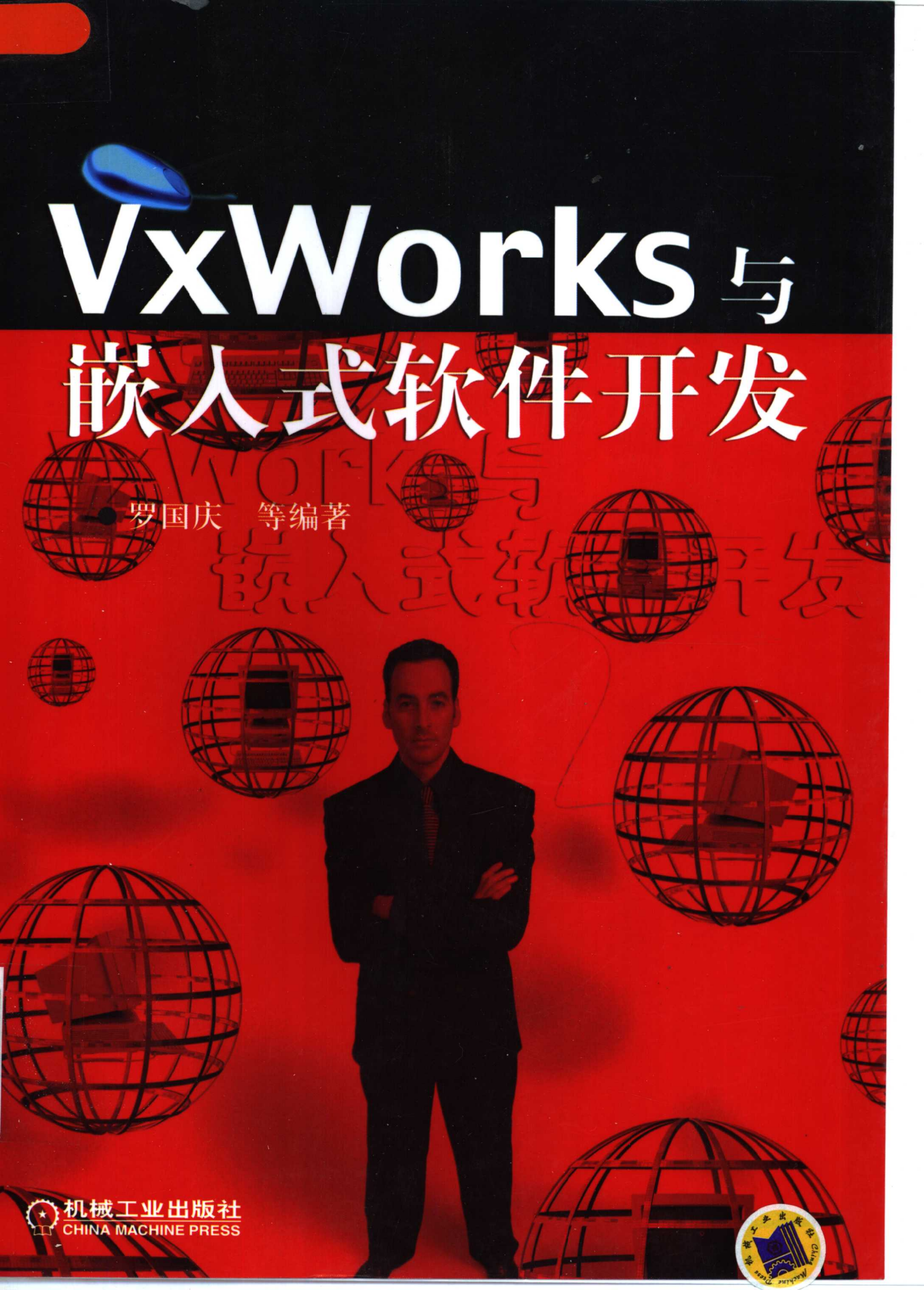
VxWorks与嵌入式软件开发

罗国庆　等编著



机械工业出版社
CHINA MACHINE PRESS

VxWorks 与嵌入式软件开发

罗国庆　等编著

机械工业出版社

本书主要介绍了 VxWorks 操作系统核心技术、Tornado 开发环境的使用和嵌入式实时软件的程序设计等内容。全书共有 9 章，主要内容包括嵌入式实时操作系统（RTOS）；VxWorks 与 Tornado 介绍；实时多任务软件的开发方法；VxWorks 的开发方法；Tornado 交叉开发环境；VxWorks 操作系统环境下的编程；Tornado 的调试方法；BSP 开发与实例；嵌入式软件测试等。

本书内容翔实、技术实用，是根据有关文献结合编者的工程开发经验编写而成的，有很强的实用和参考价值。本书适合嵌入式系统的开发、设计人员阅读，也可供从事嵌入式产品开发的广大工程技术人员学习与参考。

图书在版编目（CIP）数据

VxWorks 与嵌入式软件开发/罗国庆等编著. —北京：机械工业出版社，2003.9
ISBN 7-111-12969-5

Ⅰ.V… Ⅱ.罗… Ⅲ.实时操作系统—软件开发 Ⅳ.TP316.2

中国版本图书馆 CIP 数据核字（2003）第 075885 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：吉 玲
封面设计：陈 沛 责任印制：闫 焱

北京京丰印刷厂印刷·新华书店北京发行所发行

2003 年 9 月第 1 版第 1 次印刷
787mm×1092mm 1/16·19 印张·465 千字
0 001—4 000 册
定价：30.00 元

编辑信箱：jiling@mail.machineinfo.gov.cn

前　言

VxWorks 是专门为实时嵌入式系统设计开发的操作系统，为程序员提供了高效的实时多任务调度、中断管理、实时的系统资源以及实时的任务间通信。在各种 CPU 平台上提供了统一的编程接口和一致的运行特性，尽可能地屏蔽了不同 CPU 之间的底层差异。应用程序员可以将尽可能多的精力放在应用程序本身，而不必再去关心系统资源的管理。基于 VxWorks 操作系统的应用程序可以在不同 CPU 平台上轻松移植。

VxWorks 是美国 Wind River System 公司（即 WRS 公司）推出的一个实时操作系统。WRS 公司组建于 1981 年，是一个专门从事实时操作系统开发与生产的软件公司，该公司在实时操作系统领域被世界公认为是最具有领导地位的公司。VxWorks 是一个运行在目标机上的高性能、可裁减的嵌入式实时操作系统，以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中。

VxWorks 是一种功能强大而且比较复杂的操作系统，包括了进程管理、存储管理、设备管理、文件系统管理、网络协议及系统应用等几个部分。VxWorks 只占用很小的存储空间，并可高度裁减，保证了系统能以较高的效率运行。所以，仅仅依靠人工编程调试，很难发挥它的功能并设计出可靠、高效的嵌入式系统，必须要有与之相适应的开发工具。

Tornado 就是为开发 VxWorks 应用系统提供地集成开发环境，Tornado 中包含的工程管理软件，可以将用户自己的代码与 VxWorks 的核心有效地组合起来，可以按用户的需要裁剪配置 VxWorks 内核；VxSim 原型仿真器可以让程序员在不用目标机的情况下，直接开发系统原型，作出系统评估；功能强大的 CrossWind 调试器可以提供任务级和系统级的调试模式，可以进行多目标机的联调；优化分析工具可以帮助程序员以多种方式真正地观察、跟踪系统的运行、排除错误、优化性能。Tornado 集成环境提供了高效明晰的图形化的实时应用开发平台，它包括一套完整的面向嵌入式系统的开发和调测工具。Tornado 环境采用主机－目标机交叉开发模型，应用程序在主机的 Windows 环境下编译链接生成可执行文件，下载到目标机，通过主机上的目标服务器（Target Server）与目标机上的目标代理（Target Agent）的通信完成对应用程序的调试、分析。Tornado 是集成了编辑器、编译器、调试器于一体的高度集成的窗口环境，同样也可以在 Shell 窗口下发命令和浏览。

本书详细论述了 VxWorks 环境下开发嵌入式软件的各个方面，全书内容共分 9 章。

第 1 章介绍了嵌入式实时操作系统（RTOS）的各方面。第 2 章主要讲述了 VxWorks 与 Tornado 开发环境的基本结构。第 3 章介绍实时多任务软件的开发方法，详细讲述了实时多任务软件的各个开发步骤，最后给出了实时多任务软件的设计开发的三个实例。第 4 章论述 VxWorks 开发方法，讲述 VxWorks 开发环境的建立细节及 VxWorks 初始化流程的实现细节。第 5 章介绍 Tornado 交叉开发环境，包括 Tornado 开发工具包的介绍，最后通过一个实例来介绍 Tornado 的使用方法。第 6 章讲述 VxWorks 操作系统环境下的

编程，主要介绍了 VxWorks 操作系统环境下的编程中关于任务的创建、调度、通信方式及其中断服务机制，最后给出了通信软件开发时应该遵守的一些编程规范。第 7 章讲述 Tornado 的调试方法，详细介绍了 Tornado 调试工具的使用，并介绍了 Tornado 调试中一些常见问题及解决办法。第 8 章详细介绍 BSP 开发方法，包括 BSP 各文件的组成和作用、初始化过程和设备驱动程序等，最后给出了两个 BSP 设计的实例。第 9 章介绍嵌入式软件的测试，首先分析了在嵌入式系统开发中软件开发的重要性，接着分析了传统测试方式的缺点，并在此基础上介绍了 CodeTEST 嵌入式软件分析与测试的方法。

本书引用的部分资料及图片是书中所讲述内容所需，无侵权意图，特此声明。

本书面向广大从事电信工作，特别是嵌入式系统开发、设计人员，也可供从事嵌入式产品开发的广大工程技术人员学习与参考，本书也可作为高等院校相关专业或从事相关课题研究的本科生、研究生的参考资料。

参加本书编写及制作的人员有罗仁舟、黄定爱、陈萍、黄伟东、杜海树、冯涛、张辉、万力、王来运、陈非、曹会明、李进、方勇、苏培华、邹俊杰、韩笑、朱兴彤、刘振华等人，感谢他（她）们将自己平时点点滴滴的经验积累贡献于本书。另外，特别感谢袁占亭、蒋明伏、莫庆富等导师多年的指导。

由于编者水平有限，书中错误在所难免，请各界同仁不吝赐教。

编者

目　录

第 1 章　嵌入式实时操作系统（RTOS）

1.1　实时系统

实时系统一定是多任务的，但多任务不一定是实时的。

首先比较一下连续程序结构体、并发程序结构体、实时程序结构体，通过一个监控系统的例子，分别用伪算法写出连续程序结构体、并发程序结构体的实现程序代码。

- 连续程序结构体的特点：对传统的程序结构体，易于理解代码，程序按一定的可预知链执行；适用于执行结果不依赖于执行速度的系统，例如工资单。
- 并发程序结构体的特点：由多个连续程序结构体组成。有真并行：程序在多个 CPU 上执行；有伪并行：程序在单 CPU 上运行，有宏观上并行、微观上串行之分；适用于执行结果依赖于相对执行速度、不依赖于绝对执行速度的系统，例如 Unix，Unix 虽然是多任务的，但其算法的核心是为分时做的，是以大的任务吞吐量为设计目标的，任务之间的耦合比较松散。
- 实时程序结构体的特点：正确结构必须在时间耗尽线之前被提交，理解调试复杂，错误代价大，分析设计要慎重；例如大的嵌入式系统，多任务之间有大量的耦合关系，所以实时程序结构体是一个三维系统。

连续程序结构体、并行程序结构体处理的是与执行速度无关的数学领域的问题，实时程序结构体处理的是物理领域的问题，实时程序结构体并不提高 CPU 指令的速度，而是其程序体反映了客观世界的优先级，例如 Digital Camera 的前端由十几路 DMA 将图像读入内存，此时内存的读写速度都已经成为瓶颈。

实时，表示“立即”、“及时”。关于实时性，人们往往有不尽相同的理解和解释。一般将联机系统视作实时系统，也有人把人-机交互性的系统称为实时系统。当然，它们都是计算机发展到一定阶段的产物。

实时系统是对外来事件在限定时间内能做出反应的系统。限定时间的范围很广，可以从微秒级（如信号处理）到分级（如联机查询系统）。

实时控制系统和实时信息处理系统统称为实时系统。在实时控制系统中，计算机通过特定的外围设备（以下简称为外设）与被控对象发生联系，被控对象的信息经加工后，通过显示屏幕向控制人员显示或通过外设向被控对象发出指示，实现对被控对象的控制；在实时信息处理系统中，用户通过终端设备向系统提出服务请求，系统完成服务后通过终端回答给用户。

在实时系统中，主要有三个指标来衡量系统的实时性：响应时间（Response Time）、生存时间（Survival Time）、吞吐量（Throughput）。

（1）响应时间（Response Time）：是计算机识别一个外部事件到做出响应的时间，在控制应用中，它是最重要的指标，如果事件不能及时地处理，系统可能就会崩溃。对于不同的过程，有不同的响应时间要求。对于有些慢变化过程，具有几分钟甚至更长的响应时间都可以认为是实时的。对于快速过程，其响应时间可能要求达到毫秒、微秒、纳秒级甚至更短。因此，实时性不能单纯从绝对的响应时间长短上来衡量，应当根据不同的对象，在相对意义上进行评价。

（2）生存时间（Survival Time）：是数据有效等待时间，在这段时间里，数据是有效的。

（3）吞吐量（Throughput）：是在一给定时间内，系统可以处理的事件总数。例如通信控制器用每秒钟处理的字符数来表示吞吐量，吞吐量可能是平均响应时间的倒数，但它通常要小一些，因为在每次响应后，可能需要一段时间进行清理（Cleanup），这段时间就称为恢复时间（Recovery Time）。

实时系统强调的是实时性和可靠性，这两方面除了与计算机硬件有关（如 CPU 的速度，访问存储器的速度等）外，还与实时系统的软件密切相关。硬件是实时的，而软件往往不一定是实时的。

如何实现实时的应用系统呢？可以通过以下的途径：

（1）使用硬件的功能。

（2）微处理器的中断机制。

（3）简单的单线程循环程序。

（4）基于实时操作系统的复杂多线程程序。

这种实时系统的软件是实时应用软件和实时操作系统（Real-Time Operating System —— RTOS）两部分的有机结合，其中 RTOS 起着核心作用，由它来管理和协调各项工作，为应用软件提供良好的运行软件环境及开发环境。

1.2　实时系统的典型应用及特点

实时应用的范围很广，主要有两种应用，即嵌入式应用和一般应用。

1.2.1　嵌入式应用

嵌入式应用就是一种计算机部件内装于专用设备/系统的应用。大多数实时系统都是嵌入式应用（Embedded Applications）系统。

嵌入式计算机是一种智能部件内装于专用设备/系统的高速计算机。它的主要功能是作为一个大型工程系统中的信息处理部件，用来控制专门的硬件设备的。

这种嵌入式系统自动化程度高、威力大、反应速度快。用户不需知道装置内计算机的存在，一般不能被用户编程，它有一些专用的 I/O 设备，对用户的接口是专用的。

嵌入式计算机系统广泛地用于办公自动化、消费、通信、汽车、工业和军事领域，其中，办公自动化、消费和通信领域占据的份额最大，约 90%以上。嵌入式的典型应用有：

（1）过程控制（Process Control），即对生产过程中各种动作流程的控制，这种控制是在对被控对象和环境进行不断观测的基础上做出及时的、恰当的反应。在控制过程中，计

算机扮演着中心的角色。它通过传感器从外部接收有关过程的信息，对这些信息进行加工处理，然后对执行机构发出控制指令。

（2）通信设备（Telecommunication），如程控交换机、路由器、BB 机、大哥大、桥接器、集线器、Modem 等。

（3）智能仪器（Intelligent Instrument），如示波器、医疗仪器等。

（4）消费产品（Consumer Product），如洗衣机、微波炉、电视机、游戏机等。

（5）机器人（Robot）。

（6）计算机外设设备（Computer Peripheral），如打印机、终端、磁盘驱动器。

（7）军事电子设备和现代武器，如雷达、电子对抗，坦克、战机、战舰等。

1.2.2　一般应用

一般应用的计算机系统对用户来说是可见的，如 PC、工程工作站等。它具有标准的计算机外设，如键盘、显示器、磁盘等，它的软件具有通用的人机接口，其应用程序可按用户需要随时改变，即重新编制。它通常用于开发、数据处理等，其典型应用如下：

（1）测控计算机。在大型控制系统中，它通常与几个嵌入式计算机相连，作为它们的上位机，进行系统的总控、协调及数据存储等工作。

（2）交互式系统。实时信息查询系统，如飞机订票系统、银行交易和股票交易系统等，这些系统的响应时间要求不高，只要人可以忍受就行了。

1.3　嵌入式实时系统软件的基本特征

不难看出，与一般的计算机应用相比，嵌入式实时应用系统是具有高速处理、配置专一、结构紧凑和坚固可靠等特点的实时系统，相应的软件系统应是一种别有特色、要求更高的实时软件。对这种实时软件的主要要求是：

（1）实时性。实时软件对外部事件做出反应的时间必须要快，在某些情况下还需要是确定的、可重复实现的，不管当时系统内部状态如何，都是可预测的（Predictable）。

（2）有处理异步并发事件的能力。实际环境中，嵌入式实时系统处理的外部事件往往不是单一的，而是同时出现的，而且发生的时刻也是随机的，即异步的。实时软件应有能力对这类外部事件组有效地进行处理。

（3）快速启动、并有出错处理和自动复位功能。这一要求对机动性强、环境复杂的智能系统显得特别重要，快速机动的环境，不允许控制软件临时从盘上装入，因此嵌入式实时软件需事先固化到只读存储器中，开机自行启动，并在运行出错死机时能自动恢复先前运行状态。因此嵌入式实时软件应采用特殊的容错、出错处理措施。

（4）嵌入式实时软件是应用程序和操作系统两种软件的一体化程序。对于通用计算机系统，例如 PC、工作站，操作系统等系统软件和应用软件之间界限分明。换句话说，统一配置的操作系统环境下，应用程序是独立的运行软件，可以分别装入执行。但是，在嵌入式实时系统中，这一界限并不明显。这是因为，应用系统配置差别较大，所需操作系统繁简不一，I/O 操作也不标准，这部分驱动软件常常由应用程序提供。这就要求采用不同配置的操作系统和应用程序，链接装配成统一的运行软件系统。也就是说，在系统总设

计目标指导下将它们综合加以考虑、设计与实现。

（5）嵌入式实时软件的开发需要独立的开发平台。由于嵌入式实时应用系统的软件开发受到时间、空间开销的限制，常常需要在专门的开发平台上进行软件的交叉开发，其交叉开发环境如图 1-1 所示。

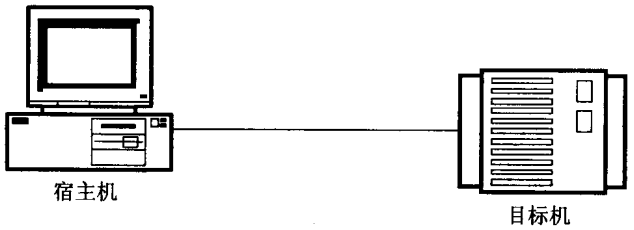


图 1-1　交叉开发环境

开发平台称为宿主机，应用系统称作目标机。宿主机可以是与目标机相同或不相同的机型。这种不同机型的开发平台又称作交叉式开发系统。显然，在这种独立的实时软件开发系统上，应配备完整的实时软件开发的工具，如高级语言、在线调试器和在线仿真器等。因此，嵌入式实时软件开发过程较为复杂。

1.4　嵌入式实时系统的分类

可按照速度即系统响应时间（Response Time）或吞吐量（Throughput）、确定性及软件结构对嵌入式实时系统进行分类。

1.4.1　按速度分类

按速度，即按实时性的强弱（根据系统响应时间的长短）来分可将嵌入式实时操作系统大致分为以下几种：

（1）强实时系统，其系统响应时间在毫秒或微秒级。

（2）一般实时系统，其系统响应时间在几秒的数量级上，其实时性的要求比强实时系统要差一些。

（3）弱实时系统，其系统响应时间约为数十秒或更长。这种系统的响应时间可能随系统负载的轻重而变化，即负载轻时，系统响应时间可能较短，实时性好一些；反之，系统响应时间可能加长。

1.4.2　按确定性分类

按确定性来分，可将嵌入式实时系统分为硬实时和软实时系统。

（1）硬实时系统。系统对系统响应时间有严格的要求，如果系统响应时间不能满足，就要引起系统崩溃或致命的错误。

（2）软实时系统。系统对系统响应时间有要求，但是如果系统响应时间不能满足，不会导致系统出现致命的错误或崩溃。

1.4.3　按软件结构分类

按软件结构来分，可将嵌入式实时系统分为以下几种：

1. 单线程程序（Single-threaded Program）

单线程程序也称为顺序程序（Sequential Program），它又分为两种：

（1）循环轮询系统（Polling Loop）。最简单的软件结构是循环轮询，程序依次检查系统的每一个输入条件，一旦条件成立就进行相应的处理。其通常的软件结构如下：

```
initialize()
while(true){
  if(condition_1)
    action_1();
  if(condition_2)
    action_2();
  ......
  if(condition_n)
    acition_n();
}
```

其优点是：

1）对于简单的系统而言，便于编程和理解。

2）没有中断的机制，程序运行良好，不会出现随机的问题缺点。

3）有限的应用领域（由于不可确定性）。

4）对于大量的 I/O 服务的应用，不容易实现。

5）大的程序不便于调试，因此，它适合于慢速和非常快速的简单系统。

（2）有限状态机（Finite State Machine —— FSM），如图 1-2 所示。这个 FSM 的两个输入和四个状态矩阵见表 1-1。

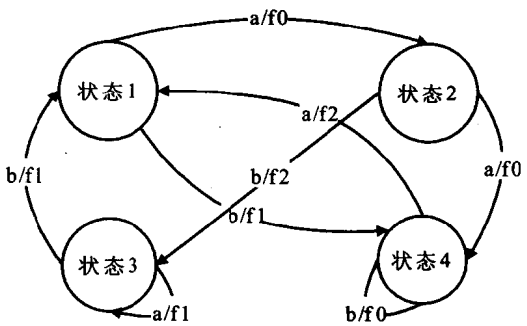


图 1-2　有限状态机示意图

表 1-1　有限状态机的状态转换矩阵

输入	a	b
状态 1	转到状态 2，输出 f0	转到状态 4，输出 f1
状态 2	转到状态 4，输出 f0	转到状态 3，输出 f2
状态 3	转到状态 3，输出 f1	转到状态 1，输出 f1
状态 4	转到状态 1，输出 f2	转到状态 4，输出 f0

其程序如下：

```
int f0(char ch)… f2(char);
struct{
  int NexState;
  int (*function)(char ch);
}StateTable[4][2]={
{{1,f0}},{{3,f1},
{{3,f0}},{{2,f2},
{{2,f1}},{{0,f1},
{{0,f2}},{{3,f0},
};

main(){
  char ch;
  int I,state;
  while(c=getch()!=q){
      I=xlate(C);
      (*StateTable[state][I].function)(c);
      state=StateTable[state][I].NextState;
  }
}
```

其优点是：

1）对于小的系统而言，便于编程和理解。

2）可以快速执行。

3）只是通过改变输出功能来改变机器的响应。

其缺点是：

1）有限的应用领域。

2）不能保证确定性。

3）对于大的应用系统，难以调试。

2. 事件驱动系统（Event-driven System）

事件驱动系统是能对外部事件直接响应的系统。它包括前后台、实时多任务、多处理器三个系统。是嵌入式实时系统的主要形式。

（1）前后台系统（Foreground/Background）又叫中断驱动系统，后台是一个循环轮询系统，一直在运行，前台是由一些中断处理过程组成的。当有一前台事件（外部事件）发生时，引起中断，中断后台运行，进行前台处理，处理完成后又回到后台（通常又称主程序）。

这种系统的一个极端情况是，后台只是一个简单的循环不做任何事情，所有其他工作都是由中断处理程序完成的。但大多数情况是中断只处理那些需要快速响应的事件，并且把 I/O 设备的数据放到内存的缓冲区中，再向后台发信号，其他的工作由后台来完成，如

对这些数据进行处理、存储、显示、打印等。

这种系统的软件运行的方式如图 1-3 所示。

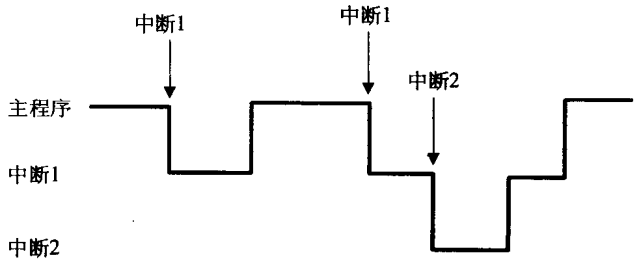


图 1-3　前后台系统运行方式

这种系统需要考虑的是中断的现场保护和恢复、中断嵌套、中断处理过程与主程序的协调（共享资源）问题。系统的性能主要由中断延迟时间（Interrupt Latency Time），响应时间（Response Time）和恢复时间（Recovery Time）来刻画。

中断延迟时间是指从中断发生到系统获知中断，并且开始执行中断服务程序所需要的最大滞后时间。

（2）实时多任务（Multitasking 或 Multi-thread Program Model）系统，如图 1-4 所示。对于一个复杂的嵌入式实时系统来说，当采用中断处理程序加一个后台主程序这种软件结构难以实时地、准确地、可靠地完成时，或存在一些互不相关的过程需要在一台计算机中同时处理时，就需要采用实时多任务系统。

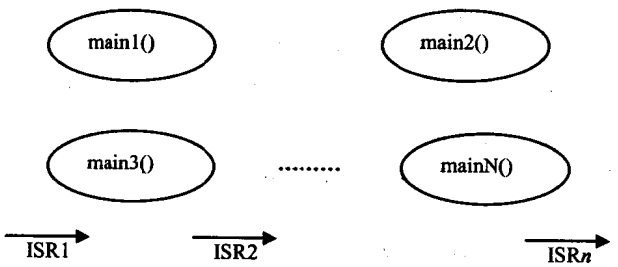


图 1-4　实时多任务系统示意图

其中，ISR 即中断服务例程（Interrupt Service Routine）。

其主要特点是：

1）多个顺序执行的程序并行运行。

2）宏观上看，所有的程序同时运行，每个程序运行在自己独立的 CPU 上。

3）实际上，不同的程序是共享同一个 CPU 和其他硬件。因此，需要 RTOS 来对这些共享的设备和数据进行管理。

4）每个程序都被编制成无限循环的程序，等待特定的输入，执行相应的任务等。

5）这种程序模型将系统分成相对简单的、相互合作的模块。

其主要优点是：

1）将复杂的系统分解为相对独立的多个线程，达到“分而治之”的目的。

2）降低系统的复杂性。

3）保证系统的实时性。

4）系统的模块化好，提高系统的可维护性。

缺点是：

1）需要采用一些新的软件设计方法。

2）需要增加功能：线程间的协调、同步和通信功能。

3）需要对每一个共享资源互斥。

4）导致线程间的竞争。

5）需要使用 RTOS，RTOS 要增加系统的开销。

实时多任务系统实际上是由多个任务、多个中断处理过程、实时操作系统组成的有机的整体。每个任务是顺序执行的，并行性通过操作系统来完成，任务间的相互通信和同步也需要操作系统的支持。

RTOS 的需求是：

1）足够的快（上下文切换和系统调用等）。

2）可确定的性能。

3）任务调度机制是基于优先级的。

4）最小的中断延迟。

5）可伸缩、可配置的体系结构。

6）可靠、健壮。

实时多任务系统的实现必须有实时多任务操作系统的支持，操作系统主要完成任务切换，任务调度，任务间通信、同步、互斥，实时时钟管理，中断管理。

（3）多处理机系统。当有些工作用单台计算机来处理难以完成时，就需要增加另外的计算机，这就是多处理机系统的由来。在单处理机系统中，多个任务在宏观上看是并发的，但在微观上看实际是顺序执行的；在多处理机系统中，多个任务可以分别在不同的处理机上执行，宏观上看是并发的，微观上看也是并发的。前者称为伪并发性，后者称为真并发性。

按照多处理机的结构，多处理机系统分为紧耦合系统（Tightly-coupled System）和松耦合系统（Loosely-coupled System）两种。紧耦合系统是多个处理器通过共享内存空间来交换信息的系统（如 SMP），而松耦合系统的多个处理器大多是通过通信线路来连接的，通过它来交换信息。

1.5 嵌入式实时操作系统及发展

1.5.1 嵌入式实时操作系统的发展

在计算机技术发展的初期阶段，计算机系统中没有操作系统这个概念。为了给用户提供一个与计算机之间的接口，同时提高计算机的资源利用率，便出现了计算机监控程序（Monitor），使用户能通过监控程序来使用计算机。随着计算机技术的发展，计算机系统的硬件、软件资源也愈来愈丰富，监控程序已不能适应计算机应用的要求。于是在 20 世

纪 60 年代中期，监控程序又进一步发展，形成了操作系统（Operating System）。发展到现在，广泛使用的有三种操作系统，即多道批处理操作系统、分时操作系统以及实时操作系统。

多道批处理操作系统一般用于计算中心较大的计算机系统中。由于它的硬件设备比较全，价格较高，所以此类系统十分注意 CPU 及其他设备的充分利用，追求高的吞吐量，不具备实时性。

分时系统的主要目的是让多台计算机用户能共享系统的资源，能及时地响应和服务于联机用户，只具有很弱的实时功能，与真正的实时操作系统有明显的区别。

20 世纪 50 年代中期到后期开发的操作系统几乎毫无结构可言，在这些整体操作系统（Monolithic Operation System）中，任何过程可以调用其他任何过程，由于低估了过程相互之间的依赖性和互操作性，产生了一系列问题。

为了解决这些问题，引入了模块化程序设计技术，特别是开发了分层操作系统（Layered Operation System）。分层操作系统的功能按照层次组织，相互之间作用只能通过邻接层。使用分层方法，大多数层或者全部层在内核模式下执行。

20 世纪 60 年代以来，对于 Unix 操作系统的研究和发展达到了几乎完美的程度。而商业 RTOS 正是基于 Unix 思想的实时多任务操作系统，只是为了满足嵌入系统的特殊需要，系统对于外部事件的响应速度保证不大于某个特定的时间间隔。

商业 RTOS 利用了计算机科学数十年发展的精美成果，包含了软件理论最精华的部分。这一点从内核技术上看得最为清楚，各个厂商的内核大同小异。

20 世纪 80 年代后期，国外提出了微内核（Microkernel）的思想，即将传统操作系统中的许多共性的东西抽象出来，构成操作系统的公共基础，即微内核，真正具体的操作系统功能则由构造在微内核之外的服务器实现。这是一种机制与策略分离的开放式设计思路。在理论上，这种方法提供了高度的灵活性、模块性和可移植性。

那么什么样的操作系统才能称为实时操作系统呢？IEEE 的实时 Unix 分委会认为，实时操作系统应具备以下几点：

（1）异步的事件响应。实时系统为能在系统要求的时间内响应异步的外部事件，要求有异步 I/O 和中断处理能力。I/O 响应时间常受内存访问、盘访问和处理机总线速度所限制。

（2）切换时间和中断延迟时间确定。

（3）优先级中断和调度。必须允许用户定义中断优先级和被调度的任务优先级，并指定如何服务中断。

（4）抢占式调度。为保证响应时间，实时操作系统必须允许高优先级任务一旦准备好运行，马上抢占低优先级任务的执行。

（5）内存锁定。必须具有将程序或部分程序锁定在内存的能力，锁定在内存的程序减少了为获取该程序而访问盘的时间，从而保证了快速的响应时间。

（6）连续文件。应提供存取盘上数据的优化方法，使得存取数据时查找时间最少。通常要求把数据存储在连续文件上。

（7）同步。提供同步和协调共享数据使用和时间执行的手段。

总的来说，实时操作系统是事件驱动的（Event-driven），能对来自外界的作用和信号