

# 基于 VxWorks 的多任务程序设计及通信管理

闫得杰

(中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033)

**摘要:** 为了满足系统各个任务对实时性的要求, 需要实现各个任务的并行处理。针对 VxWorks 操作系统的多任务调度机制和任务通信方式进行了分析; 采用基于时间片轮转调度实现多任务程序设计, 可以动态改变各个任务期望运行的时间片; 对各种通信方式和实现方法进行了比对, 并给出了优化方案。上述设计方法实时性强, 可靠性高, 系统可扩展性良好, 能够很好地满足工程需要。

**关键词:** VxWorks; 多任务; 实时操作系统; 任务通信

**中图分类号:** TP274                      **文献标识码:** A

**DOI:** 10.3788/OMEI 20102709.0064

## Design of Multi-task Programmer and Task Communication based on VxWorks

YAN De-jie

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China)

**Abstract:** In order to meet the real-time demands of the multi-task of the system, the parallel processing for all kinds of tasks was required. Multi-task scheduling and task communication mechanism about VxWorks operating system were analyzed. The design of rotating schedule based on time slice was put in use on the multi-task programming, as well as, the time slice of the expectations of each task running could be dynamically changed. At the same time, most of the communication means and methods were compared, and the better optimization was given. The method could be satisfied the engineering requirements with good performance on real time characteristics, high reliability, and good expansibility.

**Keywords:** VxWorks; multi-task; real-time operating systems; task communication

## 1 引言

在嵌入式系统的应用开发中，采用嵌入式实时操作系统能够支持多任务，使得程序开发更加容易，便于维护，同时能够提高系统的稳定性和可靠性<sup>[1]</sup>，这已逐渐成为嵌入式系统开发的一个发展方向。VxWorks 是嵌入式实时操作系统典型产品之一，由美国 Wind River System 公司推出的一个实时操作系统。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。VxWorks 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，如卫星通信、军事演习、弹道制导、飞机导航等<sup>[2]</sup>。在美国的 F-16、F-18 战斗机、B-2 隐形轰炸机和爱国者导弹上，甚至连 1997 年 4 月在火星表面登陆的火星探测器上也使用了 VxWorks。当然，在民用方面，VxWorks 也占有很大一部分市场<sup>[1]</sup>。可以说，VxWorks 操作系统在很多领域和行业都得到了极大的认可和广泛的应用。

VxWorks 操作系统为了方便程序开发及满足实时性要求，在系统中需要有多任务的调度运行，而根据不同的需要对多任务程序设计及通信管理的要求也不同<sup>[1]</sup>。操作系统虽然提供了多任务的环境，但并没有提供任务调度机制。因此，本文针对 VxWorks 操作系统的多任务调度机制和任务通信方式进行分析，采用基于时间片的轮转调度实现多任务程序设计<sup>[3]</sup>，同时讨论了任务间通信的方式和实现方法，并进行了分析和对比，给出了优化方案。

## 2 多任务程序设计

### 2.1 任务结构

多任务设计能随时打断正在执行的任务，对内部发生的事件在确定的时间内作出响应。VxWorks 实时内核 Wind 提供了基本的多任务环境。从表面上看，多个任务同时执行；实际上，系统内核根据某一调度策略使其交替运行<sup>[1]</sup>。系统调度器使用任务控

制块的数据结构（简称 TCB）来实现任务调度功能。任务控制块用来描述一个任务，每个任务都与一个 TCB 关联。TCB 包括了任务的当前状态、优先级、要等待的事件或资源、任务程序码的起始地址以及初始堆栈指针等信息；调度器在任务最初被激活以及从休眠态重新被激活时，会用到这些信息。此外，TCB 还被用来存放任务的“上下文”（Context）。任务的上下文就是当一个执行中的任务被停止时所要保存的所有信息。当任务被重新执行时，必须要恢复上下文。通常，上下文即计算机当前的状态，也就是各寄存器中的内容，同发生中断时所要保存的内容一样。当发生任务切换时，将当前运行任务的上下文存入 TCB；而将要被执行的任务的上下文从其 TCB 中取出，放入各寄存器中，于是转而执行这个任务，执行的起点是上一次它在运行时被中止的位置<sup>[4]</sup>。

### 2.2 任务状态和状态转移

实时系统的一个任务可有多种状态，其中最基本的状态有以下 4 种<sup>[5]</sup>：

- (1) 就绪态：任务只等待系统分配 CPU 资源；
- (2) 挂起态：任务须等待某些不可利用的资源而被阻塞；
- (3) 休眠态：如果系统无需某个任务工作，则该任务处于休眠状态；
- (4) 延迟态：任务被延迟时所处的状态。

表1 与任务状态迁移有关的系统调用

状态转移	调用
就绪态挂起态	semTake() /msgQReceive()
就绪态延迟态	taskDelay()
就绪态休眠态	taskSuspend()
挂起态就绪态	semGive() /msgQSend()
挂起态休眠态	taskSuspend()
延迟态就绪态	expired delay
延迟态休眠态	taskSuspend()
休眠态就绪态	taskResume() /taskActivate()
休眠态挂起态	taskResume()
休眠态延迟态	taskResume()

当系统函数对某一任务进行操作时，任务从一种状态转移到另一种状态，处于任一状态的任务都可被删除。与任务转移有关的系统调用见表 1。任务的状态及转移关系如图 1 所示。

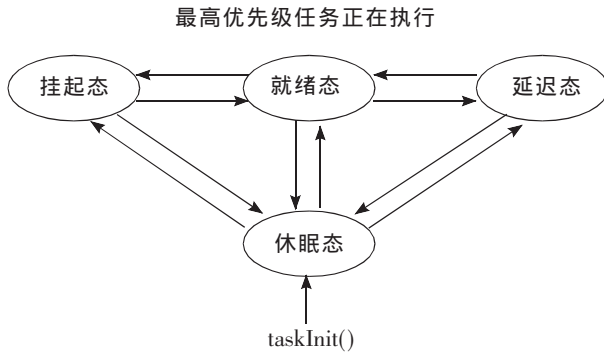


图1 任务的状态及转移关系

## 2.3 任务调度

VxWorks 操作系统支持优先抢占调度和时间片轮转调度策略，而且任务本身的开销也非常小，实现了很好的实时性；多任务调度必须采用一种调度算法来分配 CPU 给就绪态任务<sup>[6]</sup>。

### 2.3.1 基于优先级的抢占式任务调度

VxWorks 内核将优先级划分为 256 级 (0~255)，优先级 0 为最高优先级，优先级 255 为最低优先级。当任务被创建时，系统根据给定值分配任务优先级。基于优先级的抢占式具有很多优点，为每个任务指定不同的优先级。没有处于挂起或休眠态的最高优先级任务将一直运行下去；当更高优先级的任务由就绪态进入运行时，系统内核会立即保存当前任务的上下文，以便切换到更高优先级的任务<sup>[1]</sup>。任务抢占调度的实例见图 2。

优先级抢占调度就是高优先级就绪状态的任务可以抢占正在运行的低优先级的任务，中断可以抢占任何任务，因而可以保证系统的实时性<sup>[1]</sup>。但这种调度算法也增加了系统的复杂性，系统任务的执行流程是不可控的。对于共享资源需要加以保护，任务间的资源竞争会使一个高优先级的任务被迫等待一个低优先级任务完成才能执行<sup>[1]</sup>，这种情况就产生

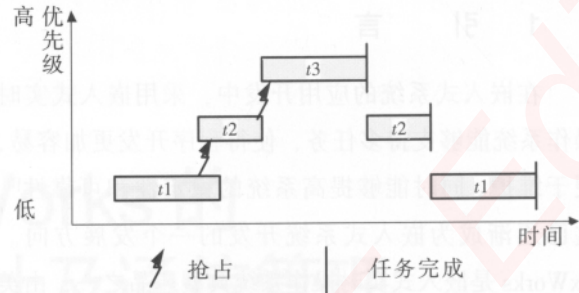


图2 任务的抢占执行

了优先级翻转。因此，要小心使用任务间的同步机制，防止出现系统死锁。

### 2.3.2 基于时间片的轮转调度

VxWorks 任务默认的调度方式为基于优先级的抢占式调度，允许和时间片轮转调度共存。基于时间片的轮转调度分配给处于就绪态的每个同优先级任务一个相同的执行时间片，从而避免长时间占用 CPU。时间片的长度可以由系统调用 KernelTimeSlice() 通过输入参数值来指定，这样，同级的任务就会按照时间片的方式轮转运行<sup>[1]</sup>。显然，每个任务都有一个运行时间计数器，任务运行时每一时间计数加一。一个任务完成时间片之后，即进行任务切换，停止执行当前运行的任务，将其放入就绪队列尾部；将运行时间计数器清零，并开始执行就绪队列中的下一个任务。当运行任务被更高优先级的任务抢占时，此任务的运行时间计数器被保存，直到该任务下次运行<sup>[1]</sup>。轮转调度及优先级调度结合实例如图 3 所示。

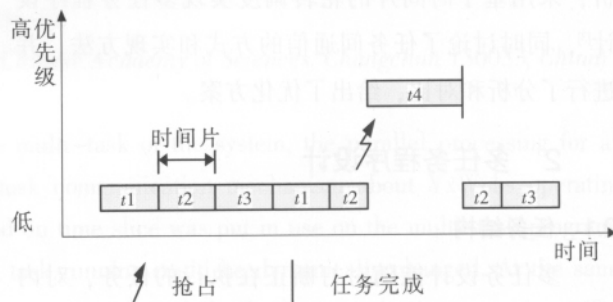


图3 任务的轮转执行

## 3 多任务通信管理

在多任务系统中，需要使用任务间通信机制协

调各任务间的活动。VxWorks 支持各任务间的通信机制，提供了多样的任务间通信方式，主要有共享内存、信号量、消息队列和管道等<sup>[7]</sup>。

### 3.1 共享内存

共享内存是任务间通信最简单的方法。通常情况下，各任务的内存空间在系统管理下是互不干扰的，但在任务间通信时需要将地址空间的同一块内存进行共享<sup>[1]</sup>。

共享内存最大的好处就是便利、快捷，同时也存在很大的危险性。当多个任务同时访问和修改共享内存区时，就很难保证数据的正确一致性。解决该问题的办法就是对共享内存进行访问上锁，即不允许多个任务同时访问该共享内存区。

### 3.2 信号量

信号量提供最快速的任务间通信机制，主要用于解决任务间的互斥和同步。针对不同类型的问题，有以下 3 种信号量，即二进制信号量、互斥信号量和计数器信号量<sup>[6]</sup>。二进制信号量使用最快捷、最广泛，主要用于同步和互斥，非常适合嵌入式软件；互斥信号量可以解决内在的互斥问题，主要用于优先级继承、安全删除等；计数器信号在二进制信号量基础上增加了计数功能，可以解决一个资源多个实例需要保护的情况。

信号量的问题在于无法实现一个任务与多个任务的通信，且无法附带更多的信息。但信号量机制开销小，通信速度快，处理方法简单，是多任务间通信的主要手段<sup>[4]</sup>。

### 3.3 消息队列

消息队列机制是相对比较高级的一种任务间通信方式，实现起来也比信号量机制复杂。使用消息队列进行任务间通信实际上是非同步的，送信任务与收信任务并不需要同时进行数据和信息的交换。可以有多个任务向一个消息队列中发送消息，也可以有多个任务从一个消息对列中去取消息，如果一直没有任务去取消息的话，消息将一直囤积直至达到最大长度<sup>[8]</sup>。消息队列机制原理如图 4 所示。

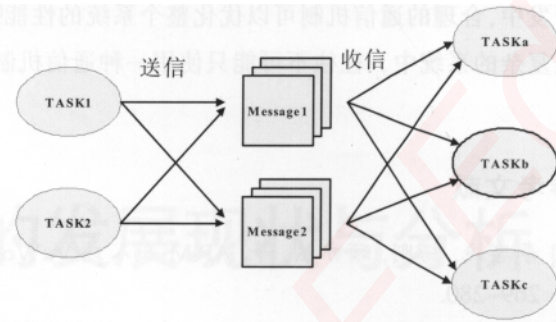


图 4 消息队列机制

在 VxWorks 中，消息队列是一种系统开销较高的通信机制，当送信消息长度超长时，送信的效率将会下降，这对于实时性要求比较高的系统来说有可能导致通信超时问题<sup>[4]</sup>。解决办法是为送信的内存单独开一个内存区域，送信时，不是之间传送信内容，而是将送信内容拷贝到 QUE\_BUFF 内存区域后，将该内存区域的指针作为送信内容；收信任务收到该送信内容后，通过指针到 QUE\_BUFF 内存区域去取实际的信息内容。

### 3.4 管道

管道是一种由 pipeDrv 驱动程序管理的虚拟 I/O 设备。管道在许多时候可以代替消息队列，使用方法一般也比消息队列简便。由于管道是 I/O 设备，这样就使得它和其他标准 I/O 一样可以使用 select 机制，从而可以方便地实现与多个异步 I/O 设备一起工作<sup>[4]</sup>。

## 4 结束语

VxWorks 是一种支持多任务的嵌入式实时操作系统，由于其高可靠性和实时性，在航空、航天、军事等领域的电子联网系统中得到了广泛应用。本文在 VxWorks 操作系统中介绍了任务结构、任务状态及状态转移的关系，针对多任务调度和通信机制进行分析，给出了基于时间片轮转调试方法更适合于多任务程序设计；同时对几种主要的通信机制的优缺点及优化方案进行了对比。上述方法综合考虑了系统中多任务程序的设计及任务间的通信机制，满足系统对各个任务实时性的要求。在系统设计和



开发中,合理的通信机制可以优化整个系统的性能<sup>[9]</sup>。根据各通信机制的特点灵活地选用多种通信机制,在复杂的系统中,往往不可能只使用一种通信机制,可达到设计开发出高效系统的目的。

## 参考文献

- [1] 李忠民, 杨刚, 顾亦然, 等. ARM 嵌入式 VxWorks 实践教程 [M]. 北京: 北京航空航天大学出版社, 2006: 269-280.
- [2] 唐晓平, 何峰, 梁甸农. 基于 VxWorks 的多中断处理设计[J]. 计算机工程, 2009, (35): 249-251.
- [3] Wind River Inc. VxWorks Programmer's Guide 5.5[EB/OL]. [2008-06-23]. <http://www.windriver.com>.
- [4] 李俊. VxWorks 下多任务间通信方式的比较与分析[J]. 信息技术, 2009, (25): 24-25.
- [5] 王永亮, 李秀娟. 嵌入式多任务程序设计[J]. 电子科技, 2010, 23(1): 94-96.
- [6] 李洪亮, 候朝桢, 周绍生. VxWorks 下实时多任务程序的实现[J]. 微计算机信息, 2008 (7-2): 90-91.
- [7] 王普, 张亚庭. 一种简单的实时多任务操作环境的实现[J]. 北京工业大学学报, 1998, 3(1): 116-122.
- [8] 曾凡林, 陶洪. 嵌入式实时操作系统中零拷贝消息队列的设计[J]. 无锡职业技术学院学报, 2006, 5(2): 21-23.
- [9] 周龙, 刘海萍. 无线通信系统信道模型研究[J]. 光机电信息, 2010, 27(4): 26-30.

作者简介: 闫得杰(1979-), 女, 汉族, 吉林德惠人, 硕士, 助理研究员。2010年于长春光机所获得硕士学位, 主要从事空间相机像移补偿和嵌入式系统的软件设计。E-mail: dejieyan@yahoo.com.cn

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)

35. [一种新的基于 RAID 的 CACHE 技术研究](#)与实现
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)
38. [北斗卫星系统在海洋工程中的应用](#)
39. [北斗卫星系统在远洋船舶上应用的研究](#)
40. [基于 CPCI 总线的红外实时信号处理系统](#)
41. [硬件实现 RAID 与软件实现 RAID 的比较](#)
42. [基于 PCI Express 总线系统的热插拔设计](#)
43. [基于 RAID5 的磁盘阵列 Cache 的研究与实现](#)
44. [基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发](#)
45. [基于磁盘阵列引擎的 RAID5 小写性能优化](#)
46. [基于 IEEE1588 的时钟同步技术研究](#)
47. [基于 Davinci 平台的 SD 卡读写优化](#)
48. [基于 PCI 总线的图像处理及传输系统的设计](#)
49. [串口和以太网通信技术在油液在线监测系统中的应用](#)
50. [USB30 数据传输协议分析及实现](#)
51. [IEEE 1588 协议在工业以太网中的实现](#)
52. [基于 USB30 的设备自定义请求实现方法](#)
53. [IEEE1588 协议在网络测控系统中的应用](#)
54. [USB30 物理层中弹性缓冲的设计与实现](#)
55. [USB30 的高速信息传输瓶颈研究](#)
56. [基于 IPv6 的 UDP 通信的实现](#)
57. [一种基于 IPv6 的流媒体传送方案研究与实现](#)
58. [基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现](#)
59. [RS485CAN 网关设计与实现](#)
60. [MVB 周期信息的实时调度](#)
61. [RS485 和 PROFINET 网关设计](#)
62. [基于 IPv6 的 Socket 通信的实现](#)
63. [MVB 网络重复器的设计](#)
64. [一种新型 MVB 通信板的探究](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)

5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)
23. [Vxworks 下的冗余 CAN 通讯模块设计](#)
24. [WindML 工业平台下开发 S1d13506 驱动及显示功能的实现](#)
25. [WindML 中 Mesa 的应用](#)
26. [VxWorks 下图形用户界面开发中双缓冲技术应用](#)
27. [VxWorks 上的一种 GUI 系统的设计与实现](#)
28. [VxWorks 环境下 socket 的实现](#)
29. [VxWorks 的 WindML 图形界面程序的框架分析](#)
30. [VxWorks 实时操作系统及其在 PC104 下以太网编程的应用](#)
31. [实时操作系统任务调度策略的研究与设计](#)
32. [军事指挥系统中 VxWorks 下汉字显示技术](#)
33. [基于 VxWorks 实时控制系统中文交互界面开发平台](#)
34. [基于 VxWorks 操作系统的 WindML 图形操控界面实现方法](#)
35. [基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发](#)
36. [VxWorks 下的多串口卡设计](#)
37. [VxWorks 内存管理机制的研究](#)
38. [T9 输入法在 Tilcon 下的实现](#)
39. [基于 VxWorks 的 WindML 图形界面开发方法](#)
40. [基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现](#)
41. [基于 VxWorks 的通信服务器实时多任务软件设计](#)
42. [基于 VXWORKS 的 RS485MVB 网关的设计与实现](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)



## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)
24. [掌握 Android 7.0 新增特性 Quick Settings](#)
25. [Android 应用逆向分析方法研究](#)
26. [Android 操作系统的课程教学](#)
27. [Android 智能手机操作系统的研究](#)
28. [Android 英文朗读功能的实现](#)
29. [基于 Yocto 订制嵌入式 Linux 发行版](#)
30. [基于嵌入式 Linux 的网络设备驱动设计与实现](#)
31. [如何高效学习嵌入式](#)
32. [基于 Android 平台的 GPS 定位系统的设计与实现](#)
33. [LINUX ARM 下的 USB 驱动开发](#)
34. [Linux 下基于 I2C 协议的 RTC 驱动开发](#)
35. [嵌入式下 Linux 系统设备驱动程序的开发](#)
36. [基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现](#)
37. [Linux 系统中进程调度策略](#)
38. [嵌入式 Linux 实时性方法](#)

39. [基于实时 Linux 计算机联锁系统实时性分析与改进](#)
40. [基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究](#)
41. [Android 手机应用开发之音乐资源播放器](#)
42. [Linux 下以太网的 IPv6 隧道技术的实现](#)
43. [Research and design of mobile learning platform based on Android](#)
44. [基于 linux 和 Qt 的串口通信调试器调的设计及应用](#)
45. [在 Linux 平台上基于 QT 的动态图像采集系统的设计](#)
46. [基于 Android 平台的医护查房系统的研究与设计](#)
47. [基于 Android 平台的软件自动化监控工具的设计开发](#)
48. [基于 Android 的视频软硬解码及渲染的对比研究与实现](#)
49. [基于 Android 移动设备的加速度传感器技术研究](#)
50. [基于 Android 系统振动测试仪研究](#)
51. [基于缓存竞争优化的 Linux 进程调度策略](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)
17. [Windows CE 环境下无线网卡的自动安装](#)
18. [基于 Windows CE 的可视电话的研究与实现](#)
19. [基于 WinCE 的嵌入式图像采集系统设计](#)
20. [基于 ARM 与 WinCE 的掌纹鉴别系统](#)
21. [DCOM 协议在网络冗余环境下的应用](#)
22. [Windows XP Embedded 在变电站通信管理机中的应用](#)
23. [XPE 在多功能显控台上的开发与应用](#)

24. [基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现](#)
25. [虚拟仪器的 Windows XP Embedded 操作系统开发](#)
26. [基于 EVC 的嵌入式导航电子地图设计](#)
27. [基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)
13. [基于 PowerPC 的雷达通用处理机设计](#)
14. [PowerPC 平台引导加载程序的移植](#)
15. [基于 PowerPC 嵌入式内核的多串口通信扩展设计](#)
16. [基于 PowerPC 的多网口系统抗干扰设计](#)
17. [基于 MPC860T 与 VxWorks 的图形界面设计](#)
18. [基于 MPC8260 处理器的 PPMC 系统](#)
19. [基于 PowerPC 的控制器研究与设计](#)
20. [基于 PowerPC 的模拟量输入接口扩展](#)
21. [基于 PowerPC 的车载通信系统设计](#)
22. [基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法](#)
23. [基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究](#)
24. [基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计](#)
25. [基于 PowerPC603e 通用处理模块的设计与实现](#)
26. [嵌入式微机 MPC555 驻留片内监控器的开发与实现](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)
15. [CortexA8 平台的  \$\mu\$ C-OS II 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)
19. [Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计](#)
20. [Nand Flash 启动模式下的 Uboot 移植](#)
21. [基于 ARM 处理器的 UART 设计](#)
22. [ARM CortexM3 处理器故障的分析与处理](#)
23. [ARM 微处理器启动和调试浅析](#)
24. [基于 ARM 系统下映像文件的执行与中断运行机制的实现](#)
25. [中断调用方式的 ARM 二次开发接口设计](#)
26. [ARM11 嵌入式系统 Linux 下 LCD 的驱动设计](#)
27. [Uboot 在 S3C2440 上的移植](#)
28. [基于 ARM11 的嵌入式无线视频终端的设计](#)
29. [基于 S3C6410 的 Uboot 分析与移植](#)
30. [基于 ARM 嵌入式系统的高保真无损音乐播放器设计](#)
31. [UBoot 在 Mini6410 上的移植](#)
32. [基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现](#)
33. [基于 ARM11 的电源完整性分析](#)
34. [基于 ARM S3C6410 的 uboot 分析与移植](#)
35. [基于 S5PC100 移动视频监控终端的设计与实现](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)
15. [基于 UEFI 固件的攻击验证技术研究](#)
16. [基于 UEFI 的 Application 和 Driver 的分析与开发](#)
17. [基于 UEFI 的可信 BIOS 研究与实现](#)
18. [基于 UEFI 的国产计算机平台 BIOS 研究](#)
19. [基于 UEFI 的安全模块设计分析](#)
20. [基于 FPGA Nios II 的等精度频率计设计](#)
21. [基于 FPGA 的 SOPC 设计](#)
22. [基于 SOPC 基本信号产生器的设计与实现](#)
23. [基于龙芯平台的 PMON 研究与开发](#)
24. [基于 X86 平台的嵌入式 BIOS 可配置设计](#)
25. [基于龙芯 2F 架构的 PMON 分析与优化](#)
26. [CPU 与 GPU 之间接口电路的设计与实现](#)
27. [基于龙芯 1A 平台的 PMON 源码编译和启动分析](#)
28. [基于 PC104 工控机的嵌入式直流监控装置的设计](#)
29. [GPGPU 技术研究与发展](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPUGPU 异构计算的混合编程模型](#)
32. [面向 OpenCL 模型的 GPU 性能优化](#)
33. [基于 GPU 的 FDTD 算法](#)
34. [基于 GPU 的瑕疵检测](#)

Programming:



1. [计算机软件基础数据结构 - 算法](#)
2. [高级数据结构对算法的优化](#)
3. [零基础学算法](#)
4. [Linux 环境下基于 TCP 的 Socket 编程浅析](#)
5. [Linux 环境下基于 UDP 的 socket 编程浅析](#)
6. [基于 Socket 的网络编程技术及其实现](#)
7. [数据结构考题 - 第 1 章 绪论](#)
8. [数据结构考题 - 第 2 章 线性表](#)
9. [数据结构考题 - 第 2 章 线性表 - 答案](#)