

实时操作系统 VxWorks 在微机保护中的应用

屈国旺, 王增平, 吴永超, 陈洪雨, 陈贺, 吕燕石, 孔江涛, 刘海涛

(石家庄科林自动化有限公司, 河北 石家庄 050222)

摘要: 基于微机保护设备中引入嵌入式实时操作系统, 可以很好地满足设备实时性和保护稳定性的要求, 满足新型保护设备的发展要求, 在对多任务实时操作系统 (RTOS) 特点分析论述的基础上, 重点介绍 VxWorks 在保护设备中的应用, 分析了多任务实时操作系统 VxWorks 对微机保护的影响, 最后研究了针对微机保护装置如何进行系统移植和应用程序开发的详细步骤。

关键词: 实时操作系统; VxWorks; 微机保护; IEC61850; 多任务; 板级开发包

Application of VxWorks of RTOS in microprocessor based relay protection

QU Guo-wang, WANG Zeng-ping, WU Yong-chao, CHEN Hong-yu,

CHEN He, Lü Yan-shi, KONG Jiang-tao, LIU Hai-tao

(Shijiazhuang KELIN Automation Ltd., Shijiazhuang 050222, China)

Abstract: With a good performance in meeting the real time requirements, it can enhance the stability of microprocessor protection relay apparatus by introducing the embedded real time operation system. Especially emphasizing on application in microprocessor protection relay, the paper also introduces the character of RTOS. Then it analyses that the VxWorks system with a real time multitask management mechanism has made a deep impact on the protection relay. According to microprocessor protection relay apparatus, the paper introduces the process of how to transplant the operation system and compile the application process.

Key words: RTOS; VxWorks; microprocessor protection relay; IEC61850; multitask; BSP

中图分类号: TM77 文献标识码: A 文章编号: 1674-3415(2009)19-0088-04

0 引言

随着数字化技术的进步, 特别是近几年 IEC61850 标准的提出, 使继电保护向着数字化、智能化、网络化的方向快速发展, 同时电力系统对保护设备的功能和性能有了更高的要求, 如短时间内处理大量数据, 快速报文的传输要在 4ms 内完成等。在硬件方面, 采用多 CPU 技术, 大容量存储芯片, 先进的通信技术可以很好的满足通信性能要求, 但在软件方面, 由于传统的前后台系统无法按任务的优先级进行选择, 任务级响应时间由后台应用程序主循环执行时间决定, 再加上前台中断程序的影响, 保护设备的实时性无法得到保证, 已无法满足新型保护设备的要求。

实时操作系统是一种实时多任务操作系统, 其最主要的特点就是任务执行的确定性和执行时间的确定性, 将 RTOS 应用于继电保护中可以充分发挥硬件平台的优势, 提高微机保护产品的整体性能。

本文选取了一种适合继电保护的 RTOS, 并分析了系统移植步骤和应用软件的开发。

1 RTOS 介绍

1.1 RTOS 概述

RTOS 是一段嵌入在目标代码中的程序模块集合, 系统复位后首先执行, 相当于用户的主程序, 用于管理和控制嵌入式系统的硬件及软件, 合理地安排嵌入式系统工作流程, 以便有效地利用这些资源为用户提供一个功能强、使用方便的工作环境, 从而在嵌入式系统和用户之间起到接口的作用。嵌入式实时操作系统是对硬件层的一次扩充, 可以为程序员提供有效的服务, 用户的其它应用程序都建立在 RTOS 之上。

RTOS 包含了一个标准的实时内核, 将 CPU 时间、中断、I/O、定时器等资源都包装起来, 留给用户一个标准的 API, 并根据各个任务的优先级, 合理

地在不同任务之间分配 CPU 时间。实时内核可以根据实际需要进行裁剪, 各个任务为不同的函数, 由实时内核进行管理和调度, 实现了任务优先级的设置, 同优先级的任务可以通过时间片轮循来并行处理, 同时 RTOS 还支持微秒级的中断处理。采用 RTOS 程序的运行程序如图 1 所示。

RTOS 的实时性体现在能够对随机发生的外部事件做出及时的响应, 并对其进行及时的处理。对电力系统继电保护产品来讲, 外部事件是指与 AD 所采集到的数据。外部事件所触发的计算、逻辑判断、出口等一串处理任务可以在严格规定的时间内完成。

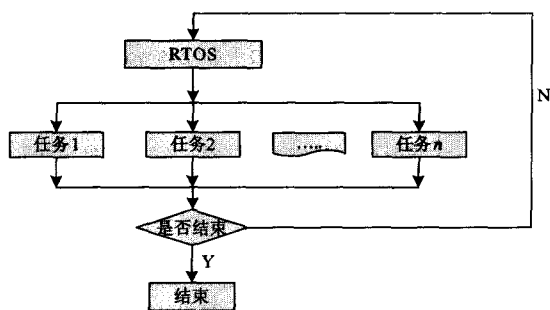


图 1 采用 RTOS 的单片机程序模型

Fig.1 Process model for microcontrollers with RTOS

1.2 RTOS 比较和选取

目前可选用的 RTOS 多达 40 多种, 每一种在性能上都有自己的特点和使用领域。由于电力系统继电保护装置软件的执行有非常严格的实时性要求和高可靠性要求, 选取一种合适的 RTOS 直接关系到保护装置的性能。

表 1 四种 RTOS 主要性能比较

Tab.1 Main performances comparison of four kinds of RTOS

性能	VxWorks	QNX	Uclinux	RTEMS
内核最小尺寸	8KB	12KB	100 多 KB	30KB
实时性	很好	很好	一般	很好
安全可靠	极高	极高	一般	很高
内核可移植性	优秀	容易	容易	容易
服务执行时间	确定	确定	不确定	不确定
系统 API	非常丰富	丰富	丰富	丰富
任务优先级数	256	255	40	255

文献[4]从多方面对 VxWorks, QNX, Uclinux, RTEMS 进行了全方面的比较, 表 1 列出了主要性能的比较。文献[5]从任务管理、任务及中断间的同步与通信机制、内存管理、中断管理、文件系统、

对硬件的支持和系统移植这几方面比较了 VxWorks、 μ Clinux、 μ C / OS-II 和 eCos 这 4 种操作系统, 并指出了各自使用的领域。

根据上述两篇文献的分析, WindRiver 公司的 VxWorks 应用范围最广, 具有高效的任务管理功能, 支持多任务多优先级, 支持优先级抢占式调试和时间片轮转调度, 实时性最好; 同时 VxWorks 具有极高的可靠性, 已成功应用于航天, 航空等关键领域; 其丰富的 API 接口和优秀的移植性能便于产品的开发和升级, 因此, VxWorks 非常适合在微机保护中应用。

2 VxWorks 对微机保护的影响

与传统的前后台系统相比, VxWorks 实时操作系统的应用将给微机保护带来很多优点, 下面列举了主要的几点:

1) VxWorks 将大大提高微机保护的实时性

VxWorks 的实时性很优秀, 其系统本身的开销很小, 进程调度、进程间通信、中断处理等系统公用程序精练而有效, 它们造成的延迟很短。VxWorks 提供的多任务机制中对任务的控制采用了优先级抢占和轮转调度机制, 也充分保证了可靠的实时性, 应用到保护装置中可以提高保护的实时性能, 满足继电保护的速动要求。

2) VxWorks 将提高保护软件的可靠性

VxWorks 在航天等关键领域的应用充分证明了其软件本身的高可靠性。另外, 与传统前后台程序相比, VxWorks 将应用程序分解成若干独立的任务, 而且可以另外启动一个监控进程, 监视各程序运行状况, 在遇到强干扰使某些进程遭到破坏时, 将有问题的进程终止掉, 再调用另一个进程将任务修复, 不会引起死机现象, 从而使系统的可靠性大大提高。

3) VxWorks 将提高软件的开发和维护效率

VxWorks 提供了非常丰富的 API 接口, 使应用程序编写简单, 直观。VxWorks 以板级支持包(BSP)的形式将低级的依赖硬件的代码和高级的应用程序代码分离开, 在做系统移植时, 只需改变 BSP 就可以完成, 无需改变上层的代码。同时, 风河公司提供了 2000 个板级支持包, 大大减小了软件开发和维护的难度和工作量。

3 VxWorks 操作系统移植

当一个硬件平台搭建完成后, 要将操作系统移植到板子上, 移植过程依赖具体的硬件配置, 主要包括 BSP 开发和映像文件的生成两个步骤。

3.1 Tornado 开发环境

Tornado 是实现嵌入式实时程序的软件开发平台，该平台运行在一台 PC 机或工作站上(宿主机)，宿主机通过某种通信方式与开目标板相连。采用这种前后台方式可以在宿主机上进行系统移植代码的开发和应用程序的编写，并将写好的程序进行编译，下载到目标板上进行调试。

Tornado 开发系统包含三个高度集成的部分：

(1) 运行在宿主机和目标机上的强有力的交叉开发工具和实用程序；

(2) 运行在目标机上的高性能、可裁剪的实时操作系统 VxWorks；

(3) 连接宿主机和目标机的多种通讯方式，如：以太网，串口线，ICE 或 ROM 仿真器等。

3.2 开发板级开发包 (BSP)

BSP 是指针对具体的硬件平台，用户所编写的启动代码和部分硬件设备驱动程序的集合，属于 VxWorks 操作系统的一部分。它由运行 VxWorks 的某些硬件驱动文件组成，如有串行线的 VME 板、时钟和其它设备。BSP 的作用是对各种板子的硬件功能提供了统一软件接口，屏蔽硬件差异，负责目标板硬件的初始化，实时内核的载入等。在做系统移植时只需改变板子的 BSP 部分，无需对上层软件进行修改。运行 VxWorks 的任何板子都需要 BSP。

BSP 的开发要依据目标板具体配置进行，可以向 WindRiver 公司或其他第三方公司订购，也可以根据 Tornado 环境自带的 BSP，结合目标板的硬件结构对相应文件进行修改得到。Tornado 自带了 2000 多个 DEMO 板 BSP，开发者总会找到和自己目标板相似的一个（最起码使用相同的 CPU）。

BSP 文件在 VxWorks/Target/Config/All 和 VxWorks/TarGet/ConFig/BspName 文件夹里。其中 all 文件夹里的文件是所有 BSP 的通用文件，一般不需要修改。BspName 文件夹的文件是用户自己开发的 BSP 文件，文件夹由开发者自己创建，里面文件最初由拷贝模板文件得到，开发者再根据目标板对文件进行相应修改。表 2 列出了需要修改的主要文件及文件的功能。

此外，如果系统硬件包括串口，还要根据具体情况修改 SysSerial.c 文件；如果包含网络部分要修改 ConfigNet.h；如果包含 NVRAM 要修改 Bootnv.h 文件。总之，BSP 的开发要根据具体目标板的硬件进行。

表 2 开发 BSP 主要需要修改的文件

Tab.2 Main files need amendment in BSP design

文件	功能
Makefile	定义编译和链接整个 BSP 的规则，控制生成 VxWorks 映像文件的类型，包含了一百多个宏，主要有：CPU, TOOL, RAM_LOW_ADRS, RAM_HIGH_ADRS 等
Bspname.h	根据具体目标板设置串行接口、时钟以及 I/O 设备等。在该文件中必须包含以下内容： • 中断向量/级别 • I/O 设备地址 • 设备寄存器位的含义 • 系统和附加时钟参数（最大和最小速率）
Config.h	缺省启动参数、Cache、MMU 开关、ROM 与 RAM 的起始地址及大小（必须与 MakeFile 中的定义一致）
RomInit.s	该文件包含引导 ROM 和基于 ROM 的 VxWorks 映像的入口初始化汇编代码，是系统加电启动后首先执行的代码。主要功能有：保存启动类型，使处理器复位；初始化 Flash 和 SDRAM，设置 MMU 到已知状态；指令 Cache 使能等。
SysALib.s	与 RomInit.s 文件实现的功能相似，但如果在 RomInit.s 文件里对 DRAM 和内存控制器进行了初始化在这里不再进行这项工作；

3.3 生成引导程序和映像文件

BSP 开发完后，就可以生成 BootRom 引导程序和 VxWorks 运行映像。

BootRom 的作用相当于 PC 机中的 BIOS，它完成对加载过程中所需设备的初始化及驱动，然后通过某种可选择的通信手段(如以太网口、串口)，将 VxWorks 映像加载到内存中。文献[6] 详细介绍了 Bootrom 的生成过程，需要注意的一点是在选择编译工具（GNU 或 DIAB）时，要与 MakeFile 中的设置一致。BootRom 生成以后需要有仿真器将其写入到目标板的 ROM/Flash 中。

VxWorks 映像是可以运行在目标板的操作系统本身的代码，包括内核和组件部分。生成过程如下：

- 创建一个“Bootable VxWorks Image”工程。
- 选择需要的 VxWorks 组件。

选择操作程序用到的系统组件，如果是在程序开发和调试阶段，应添加两个与通信相关的组件，Telnet Server 和 Target Shell。前者使我们可以透过 Telnet 协议登录到 VxWorks 操作系统中；后者则可以让我们通过命令行控制 VxWorks 系统。完成选择后，即可开始编译生成 VxWorks 映像。

通过网络配置，目标板上电后引导程序通过网络将 VxWorks 映像从宿主机上下载到目标板上并

进行调试,至此,完成了 VxWorks 到目标板的移植。需要注意的是,在每次修改完系统的配置信息(如: config.h)后,都要重新创建一个工程来编译 VxWorks 映像,以免出现代码不一致的问题。

4 应用程序开发

应用程序运行在操作系统之上,实现保护的各种具体功能。Tornado 提供了对 C 和 C++语言的支持,可以直接在 Tornado 环境内创建一个可下载的应用程序模块,并为工程填加 C/C++文件进行编写。

由于 VxWorks 采用多任务,优先级抢占机制,因此在编程中重点应放在对整个程序结构的把握以及对任务,中断进行划分,任务调度的实现等问题上,至于保护算法的具体实现与传统保护程序相似。下面以满足 IEC61850 标准的保护设备为例,对任务及优先级的划分问题进行简单分析。

VxWorks 将任务分为 256 个优先级,对任务及优先级进行合理的划分是实现实时性的保证。根据文献[6]给出的划分任务时的原则,即功能内聚性,时间紧迫性和周期执行原则。根据这一原则可将任务按优先级由高到低进行如下划分:

1) 与保护相关的计算和逻辑判断模块划分为一个任务,如将傅氏计算,CT/PT 断线判断,启动/闭锁条件计算,以及各种速动保护(电流速断,差动保护等)启动条件的判断。

2) 发送跳闸、闭锁、重合闸等信号划分为一个任务,负责向智能终端发送快速报文。

3) 故障录波功能划分为一个任务,负责记录故障时间和波形。

4) 具有时间延时的保护功能模块划分为一个任务,如定时限过流保护,过负荷保护等。

5) 报警功能,事件记录功能以及显示等功能可以分别划分为单独任务。

对于优先级比较低的任务,可以采取将几个任务划分为同一等级,采用时间片轮转调度的方法来执行,这样可以避免因某一任务优先级过低而长时间得不到执行。

Tornado 自带了图形化的交叉调试器,具有十分强大的调试功能,程序编写好后编译成.out 文件,可以下载到目标板上进行交叉调试。

5 结论

开发基于嵌入式实时操作系统的保护设备是继电保护发展的必然趋势。本文以 VxWorks 为例介绍了操作系统的移植过程以及应用软件的开发重点。需要指出的是,在开发过程需要将 bootrom 引导程

序, VxWorks 操作系统和应用程序分开进行,这样是为了便于修改调试。当程序完成以后可以合并引导程序和操作系统,生成自启动映像,用于最终产品。

参考文献

- [1] 王海吉,王书强.基于 IEC61850 标准开发数字变电站可编程的 IED[J].继电器,2008, 36(3):37-41.
WANG Hai-ji,WANG Shu-qiang.Development of Programmable IED for Digital Substation Based on IEC61850 Standard[J]. Relay,2008,36(3):37-41.
- [2] 夏林峰.嵌入式系统的 BSP 软件包设计[J].电子产品世界,2003,12:42-44.
XIA Lin-feng. BSP Design for Embedded System [J].Electronic Product World,2003,12:42-44.
- [3] 徐少毅.基于 VxWorks 的 BSP 概念与开发[J].电子产品世界,2002,3(8):26-28.
XU Shao-yi. Concept and Development of BSP Based VxWorks[J].Electronic Product World,2002,3(8): 26-28.
- [4] 孙鲁毅.四种流行的嵌入式实时操作系统的比较研究[J].计算机应用与软件,2007,24(8):196-197.
SUN Lu-yi. The Comparison and Research of Four Popular RTOS[J]. Computer Applications and Software, 2007,24(8):196-197.
- [5] 胡曙辉,陈健.几种嵌入式实时操作系统的分析与比较[J].单片机与嵌入式系统应用,2007,(5):56-58.
HU Shu-hui,CHEN Jian.Analysis and Compare of Several Embedded Real Time Operating System [J].Microcontrollers and Embedded Systems, 2007,(5):56-58.
- [6] 郭毅,马卫红,等.实时操作系统在微机继电保护装置中的应用[J].电力科学与工程,2008,24(3):20-23.
GUO Yi,MA Wei-hong,et al. Application of Real Time Operating System in Microprocessor Protection Relay Apparatus [J]. Electric Power Science and Engineering,2008,24(3):20-23.
- [7] 路小俊.基于嵌入式实时操作系统的全网络话数字保护设备的研究[D].南京:东南大学,2006.
LU Xiao-jun.Research on Networked Digital Protection Relay Based on Embedded Real Time Operation Sysetm[D].Nanjing: Southeast University,2006.

收稿日期:2008-10-23; 修回日期:2009-07-20

作者简介:

屈国旺(1969-),男,工程师,本科,主要研究方向电力系统继电保护;E-mail: 3838985@21cn.com

王增平(1964-),男,教授,主要从事电力系统继电保护和变电站自动化的教学和科研工作;

吴永超(1981-),男,硕士,主要研究方向电力系统继电保护。

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](#)

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](#)

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 I/O 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](#)

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 技术研究与实现](#)
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](#)

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](#)