EFI及其安全性研究

谢　勇，来学嘉，邓子健

（上海交通大学 计算机科学与工程系，上海 200240）

【摘　要】为了解决传统PC BIOS的局限性及其面临的问题，Intel公司提出了可扩展固件接口(EFI)的规范标准。作为下一代BIOS，EFI为启动操作系统前的程序提供了一个标准环境。文中详细介绍了EFI，指出EFI存在的一些安全问题，并分析相关的安全机制，指出了实现EFI安全必须考虑的因素。

【关键词】EFI；可扩展固件接口；BIOS；安全性

【中图分类号】TP309　　【文献标识码】A　　【文章编号】1002-0802(2007)08-0175-03

Security Analysis of EFI Framework

XIE Yong , LAI Xue-jia , DENG Zi-jian

(Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

【Abstract】In order to solve the limitations and difficulties of traditional PC BIOS, Intel has proposed Extensible Firmware Interface (EFI) Specifications. As the next generation of BIOS, EFI provides a standard environment for booting an OS. This paper gives a brief introduction of EFI, takes a closer look at some security issues in EFI, and analyzes its security architecture. Moreover, some key factors that have to be taken into account in a secure EFI environment are proposed.

【Key words】EFI;extensible firmware interface;BIOS; security

0 引言

传统计算机操作系统启动前的硬件初始化工作及操作系统的引导控制权都是BIOS (Basic Input/output System)来完成的。随着计算机科学的发展，传统 BIOS 逐渐成为一个瓶颈，难以适应现代计算机技术的发展需求，比如 16 位实模式 BIOS 不适用于 Intel 安腾体系架构。因此有必要使用一种新的固件技术来代替传统 PC BIOS。EFI 是 Intel 公司提出的用来替代现行 BIOS 的一项技术，其全称是 Extensible Firmware Interface（可扩展固件接口）。它定义了操作系统与平台固件之间的接口模型。EFI 主要由一系列包含平台相关信息的数据表(Data Tables)和供操作系统引导程序、操作系统调用的启动服务（Boot Service）和运行时服务(Runtime Service)构成。这些部件联合起来为一个操作系统的启动与预启动程序的执行提供了一个标准环境[1]。

由于 EFI 具有良好的前瞻性，2005 年业界成立了 UEFI(Unified EFI)论坛，共同制订适应于各种平台的接口标准，并于 2006 年 1 月发布了 UEFI2.0 版本[1]。UEFI 的制订建立在 EFI 1.10 的基础上，现在的最新版本为 2.1。UEFI 的主要成员有 AMD、AMI、Dell、HP、IBM、Insyde、Intel、Microsoft 和 Phoenix 等。

为了更好地实现 EFI 规范，Intel 发起并建立了一个软件框架，工程代码为 Tiano。Tiano 采用了 EFI 行业标准，正式名称是“EFI 创新框架英特尔平台”。它能够实现初始化系统配置，然后加载操作系统或基于 Intel 架构(IA)处理器的嵌入式操作环境。

1 EFI 及其框架

1.1 EFI 系统架构

EFI 规范被设计成是一个纯粹的接口标准。这个规范定义了一组平台固件必须实现的接口和数据结构。类似的，规范定义了一组 OS 在启动过程中可以使用的接口和结构。EFI 的设计主要是基于以下核心元素：现存架构接口的重复使用、系统分区、启动服务和运行时服务。图 1 是 EFI 的系统架构图，描绘了 EFI 与固件、硬件、OS 加载程序与 OS 的关系。图中的阴影部分（“ H ”槽）为 EFI 的核心部分。可以清晰地看到，EFI 连接着操作系统和平台固件，而且 EFI 接口是与 CPU 架构无关的，同时 EFI 采用了驱动模型来使多方码和程序能够协调工作。因此，可以在不影响操作系统的情况下对 EFI 进行修改。此外，EFI

的可扩展性也使它能够适应不同的平台，并且能方便的给平台增加新的功能接口。

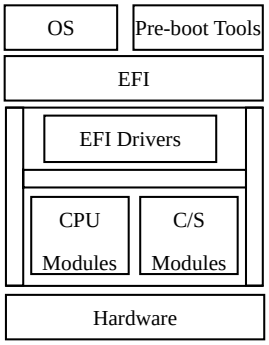

图 1 EFI 系统架构

1.2 EFI/Tiano 框架流程

EFI/Tiano 采用模块化设计，并基于现代软件体系设计的思想，将 EFI/Tiano 框架流程主要划分为 SEC、PEI、DXE、BDS、TSL、RT 和 AL 等 7 个阶段，如图 2 所示。

SEC 阶段是平台上电后执行的第一步。这个阶段的主要目的是对平台固件进行验证，确立整个平台的可信根。EFI 使用 C 语言编写，需要内存的开销。然而 PEI 阶段初始时系统并不能感知内存的存在，它能够使用的仅仅是 SEC 阶段传入的临时内存信息（通过 CPU 缓存建立）。因此 PEI 阶段的主要任务是“ 唤醒 ”系统内存，同时提供一种机制来初始化 CPU，芯片或其它平台组件，然后将平台固件的信息传递到 DXE 阶段。PEI 的核心部件包含一个 PEI 核和一个或多个 PEI 初始化模块（PEIMs）。其中，PEIM 是单独连接成的二进制模块，经过压缩后存储于 ROM 中。PEIMs 之间的调配访问是通过一组 PPIs（PEIM-to-PEIM Interfaces）来实现的。

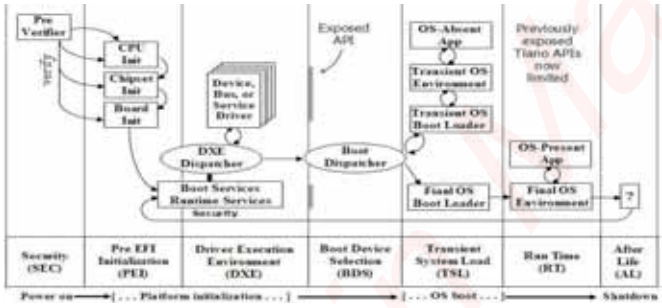

图 2 EFI 框架流程

平台固件的主要初始化配置都是在 DXE 阶段完成，如硬件的驱动等。DXE 阶段通过加载驱动的方式（轮询检测）来为操作系统的启动管理构建环境。这个阶段码也将不再涉及硬件编码，而是通过调用相关协议（Architectural Protocols）来执行硬件操作。PEI 和 DXE 为 EFI 的核心部分，它们构成了图 1 中的“ H ”槽。BDS 阶段是 EFI 拥有平台控制权的最后一个阶段。它结合 DXE 为启动操作系统建立控制台，提供图形用户界面，确定可用的引导设备和启动镜像，并支持远程启动等服务。

TSL 即操作系统启动管理器尝试引导操作系统的阶段。RT 是操作系统启动运行后，EFI 提供的一组运行时服务。最后一个阶段，即 AL 阶段提供一种机制来保证用户在有意或无意地情况下终止操作系统后，让 EFI 重新获得系统控制权。

1.3 EFI 的优点

尽管 EFI 的功能类似于 BIOS，但它定义了更加明细完善的固件平台接口，有着自己的文件系统，突破了传统 BIOS 16 位实模式和 1MB 内存寻址的限制，开机后进入 32/64 位保护模式。此外，EFI 使用 C 语言编写，采用模块化、驱动模式和动态链接的形式来实现。EFI 的驱动也有别于 BIOS 的中断检测模式，而是通过轮询检测的方式来加载驱动程序。当 EFI 所有组件加载完毕后，系统会提供图形用户界面，可以开启 SHELL 环境，并支持系统管理配置及网络协议等服务。这些优点在保证 EFI 向下兼容性和扩展性的同时，也表明了 EFI 能够充分替代 BIOS 的角色。

2 EFI/Tiano 的安全性问题

虽然 EFI 比传统 BIOS 有诸多优点，解决了 BIOS 设置难、扩展难等瓶颈，但由于其组件加载模式、高级语言编写和软件框架特性，我们需要更加重视它的安全性。

2.1 缓冲区溢出漏洞

在程序中，C 语言不进行数组的边界检查。在动态分配缓冲区后，如果尝试向该缓冲区写入超出其所能装载的数据，就会覆盖掉堆栈的其他数据，甚至覆盖函数的返回地址，这时就发生了缓冲区溢出现象。利用缓冲区溢出进行攻击是黑客常用的攻击手段。堆栈缓冲区溢出就是典型的攻击形式。而目前的静态和动态检测方法还不能有效地解决缓冲区溢出的问题。EFI 框架中，从 DXE 阶段开始就可以自由分配系统内存，加之 EFI 驱动模式的灵活应用，因此缓冲区溢出漏洞成为 EFI 不可避免的安全隐患。

2.2 EFI 安全启动

能否安全启动对于平台的安全性至关重要。首先，由于 EFI 不像传统 BIOS 那样固化在只读存储器芯片内，它可以存放在闪存中，甚至是硬盘上。因此 EFI 需要诊断其映像文件的安全性，并能够准确而及时地判断出映像文件是否被非法篡改。其次，固件平台在启动时有必要建立一条唯一的可信链。从 EFI 框架的流程看，SEC 是 Tiano 的第一个阶段。它会执行一些必要工作及查找和加载下一阶段 PEI 的入口点（PEI 核），SEC 也自然而然地成为了这条可信链的可信根。可想而知，一旦 SEC 被攻击，那么整个平台固件就无安全性可言。因此，EFI 必须保证处理器执行的第一条指令是完全可信安全的，并依此来决定后续代码的安全性，从而顺序地建立平台的唯一可信链。

另外，要有效地实现安全启动的目标还需要硬件的支持，仅仅依靠软件是难以实现的。2004 年，Hendricks 和 Doorn 指出需要使用可信计算基 TCB(Trusted Computing Base)来加强系统的启动安全[3]。TCB 是操作系统用于实现安全策略的一个集合体。目前的硬件，如 CPU 和芯片组还不支持安全特性，在下一代 CPU 和芯片组的实现中可以加以利用，甚至考虑将 EFI 框架的可信根引入 CPU。采用这种方法，CPU 厂商会为所有的 PEI 模块加上安全的 CRC 或 MAC 信息，防止 PEI 模

签名确认失败，不满足可信关系时，处理器能够重置一些恢复条件或终止相应对象的执行权限。

2.3 网络安全

EFI 在其所有组件都加载完成后，可以在终端开启一个 SHELL 程序，使得系统在不进入操作系统的情况下就可以进行系统配置、诊断等操作，另一方面，EFI 规范提供了 ARP、TCPv4、IPv4 和 UDPv4 等网络协议。这些网络协议的加载和应用，又使得平台可以执行远程启动及配置操作，于是在连接通讯前，服务器和客户端都有必要对对方进行身份鉴别和完整性检测，保护通信及系统平台的安全性。例如，在一个局域网内，所有系统平台都是基于 EFI，服务器需要不定时地统计所有客户机的状态信息，或者发送控制命令（如重启、关机命令）给某客户机。因为 EFI 具有硬件驱动，可以独立地控制系统。所以在使用 EFI 的网络协议实现通讯时，如果缺少相应的安全检测，将会对系统平台的安全构成威胁。

3 EFI/Tiano 的安全机制

EFI 及其框架的安全机制主要集中在两个方面。一是 SEC 阶段的安全机制，二是 EFI 框架提供的安全服务机制。

3.1 SEC 阶段的安全机制

SEC 阶段的安全是保障固件平台安全启动的重要环节。此时，固件平台没有硬件支持，所以它更注重于完整性检测，而非安全性检测。如前所述，EFI 框架需要为平台建立唯一的一条可信链。当可信根产生后，后续执行阶段就必须遵守这样的安全策略：SEC 本身及其之后即将获得平台控制权的对象模块，如 PEI 核，它们在执行前，平台使用哈希扩展函数对它们进行鉴别验证。SEC 阶段的代码逻辑等价于 BIOS 框架的可信根的核心 CRTM（Core Root-of-Trust Module）。但它又与传统 CRTM 有所区别，其显著特点是它包含了 TCB 和一些执行安全服务的组件，如安全审计等服务。

另外，SEC 会传递一个非空的 PPI 描述表给 PEI 核，并与 PEI 共享部分服务集，其中包括 Security PPI 和可信计算组织 TCG（Trusted Computing Group）PPI。这两个 PPI 为 CRTM 提供支持，让 CRTM 拥有足够的安全函数来对 PEI 核进行鉴别，同时允许 PEI 核重用安全代码。但在这种情况下，SEC 需要利用安全协处理器，如 TCG 中的可信平台模块 TPM[4]。

3.2 安全服务机制

EFI 提供了相应的机制来保护其框架的安全性。配置安全的 EFI 环境关键在于运行可信程序的保障能力和 EFI 映像的完整性检测能力。EFI 的完整性和鉴别授权验证是基于公钥密码、单向函数、数字签名、公钥证书和可信根等密码体制。

EFI 框架在 PEI 向 DXE 阶段转换的过程中，DXE 会加载一

（1）支持启动完整性服务 BIS（Boot Integrity Services）和可信计算组织 TCG 等标准的安全服务；

（2）安全体系架构协议；

（3）映像的签名实现。

EFI BIS 协议提供了校验 EFI 映像完整性的有效策略机制。它引入了启动授权 BA 公钥（Boot Authentication Public Key）的概念。平台系统的合法所有者拥有一对 BA 公、私钥，用来实施 EFI 驱动及应用程序在平台上运行的授权。映像加载过程中，系统会对每个驱动或应用程序进行完整性检测，判断其是否得到合法授权。与此同时，系统会对目标对象的附属签名清单（该清单包括每个对象相关属性、HASH 值及其数字签名等）重新签名。之后，系统使用 BA 公钥对所有的模块或应用程序及清单进行签名验证，从而保证平台的安全性。

这些服务为平台提供了多种安全功能，如密码原函数、获取及更新 BA 公私钥对、验证对象清单和记录验证结果并执行应用程序等。安全体系架构协议和 BDS 启动管理器使用这些安全服务来查找、加载和执行平台驱动。平台启动中所有发现的模块和驱动的完整性及真实性也都会通过这些安全服务来实现。

4 结语

综上所述，与传统 BIOS 相比较，EFI 体现出了更多的优势及发展空间。当 EFI 的应用日益广泛时，做好 EFI 及其框架的安全性研究是十分必要的。实现安全的 EFI 环境关键在于运行安全可信的程序代码，并且检测 EFI 映像文件的完整性，以确保其未被非法篡改。因此固件平台需要安全且及时地建立可信根，而 EFI 框架的后续执行阶段也需遵守 EFI 的安全策略。此外，结合 TPM 和支持安全特性的硬件也将增强 EFI 及其框架的安全性。

参考文献

[1] Intel Corporation. Extensible Firmware Interface Specification Version 1.10 [DB]. http://developer.intel.com/technology/efi/, 2002.12.

[2] Unified EFI Forum. Unified Extensible Firmware Interface Specification Version 2.0 [DB]. http://www.uefi.org/specs/, 2006.01.

[3] James Hendricks and Leendert van Doorn. Secure Bootstrap is Not Enough: Shoring up the Trusted Computing Base [J].In Proceedings of the Eleventh SIGOPS European Workshop, ACM SIGOPS, Leuven, Belgium, 2004：11～15.

[4] The Trusted Computing Group. TPM Main Part 1 Design Principles [DB/OL]. https://www.trustedcomputinggroup.org/specs/TPM/, 2006.03.

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. 基于磁盘异或引擎的 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
67. 基于 ARM7 的 MVB CAN 网关设计
68. 机车车辆的 MVB CAN 总线网关设计
69. 智能变电站冗余网络中 IEEE1588 协议的应用
70. CAN 总线的浅析 CANopen 协议
71. 基于 CANopen 协议实现多电机系统实时控制
72. 以太网时钟同步协议的研究
73. 基于 CANopen 的列车通信网络实现研究
74. 基于 SJA1000 的 CAN 总线智能控制系统设计
75. 基于 CANopen 的运动控制单元的设计
76. 基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现

77. 分布式控制系统精确时钟同步技术
78. 基于 IEEE 1588 的时钟同步技术在分布式系统中应用
79. 基于 SJA1000 的 CAN 总线通讯模块的实现
80. 嵌入式设备的精确时钟同步技术的研究与实现
81. 基于 SJA1000 的 CAN 网桥设计
82. 基于 CAN 总线分布式温室监控系统的设计与实现
83. 基于 DSP 的 CANopen 通讯协议的实现
84. 基于 PCI9656 控制芯片的高速网卡 DMA 设计
85. 基于以太网及串口的数据采集模块设计
86. MVB1 类设备控制器的 FPGA 设计
87. MVB 接口彩色液晶显示诊断单元的显示应用软件设计
88. IPv6 新型套接字的网络编程剖析
89. 基于规则的 IPv4 源程序到 IPv6 源程序的移植方法
90. MVB 网络接口单元的 SOC 解决方案
91. 基于 IPSec 协议的 IPv6 安全研究

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
43. 实时操作系统 VxWorks 在微机保护中的应用
44. 基于 VxWorks 的多任务程序设计及通信管理
45. 基于 Tilcon 的 VxWorks 图形界面开发技术
46. 嵌入式图形系统 Tilcon 及应用研究
47. 基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现
48. 基于嵌入式的 Tilcon 用户图形界面设计与开发
49. 基于 Tilcon 的交互式多页面的设计
50. 基于 Tilcon 的嵌入式系统人机界面开发技术
51. 基于 Tilcon 的指控系统多任务人机交互软件设计
52. 基于 Tilcon 航海标绘台界面设计
53. 基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发
54. VxWorks 环境下内存文件系统的应用
55. VxWorks 下的多重定时器设计
56. Freescale 的 MPC8641D 的 VxWorks BSP
57. VxWorks 实验五[时间片轮转调度]
58. 解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题
59. 基于 VxWorks 系统的 MiniGUI 图形界面开发
60. VxWorks BSP 开发中的 PCI 配置方法
61. VxWorks 在 S3C2410 上的 BSP 设计
62. VxWorks 操作系统中 PCI 总线驱动程序的设计与实现

63. VxWorks 概述
64. 基于 AT91RM9200 的 VxWorks END 网络驱动开发
65. 基于 EBD9200 的 VxWorks BSP 设计和实现
66. 基于 VxWorks 的 BSP 技术分析
67. ARM LPC2210 的 VxWorks BSP 源码
68. 基于 LPC2210 的 VxWorks BSP 移植
69. 基于 VxWorks 平台的 SCTP 协议软件设计实现
70. VxWorks 快速启动的实现方法[上电到应用程序 1 秒]

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 CC++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
52. Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档
53. 基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现
54. 路由信息协议在 Linux 平台上的实现
55. Linux 下 IPv6 高级路由器的实现

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
28. 基于 XPE 的数字残币兑换工具开发
29. Windows CENET 下 ADC 驱动开发设计
30. Windows CE 下 USB 设备流驱动开发与设计
31. Windows 驱动程序设计
32. 基于 Windows CE 的 GPS 应用
33. 基于 Windows CE 下大像素图像分块显示算法的研究
34. 基于 Windows CE 的数控软件开发与实现

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
28. 基于 PowerPC 架构多核处理器嵌入式系统硬件设计
29. 基于 PowerPC 的多屏系统设计
30. 基于 PowerPC 的嵌入式 SMP 系统设计
31. 基于 MPC850 的多功能通信管理器
32. 基于 MPC8640D 处理系统的技术研究
33. 基于双核 MPC8641D 处理器的计算机模块设计

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μC-OSⅡ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
36. UBoot 在 AT91RM9200 上的移植简析
37. 基于工控级 AT91RM9200 开发板的 UBoot 移植分析

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
35. 基于 GPU 通用计算的分析与研究
36. 面向 OpenCL 架构的 GPGPU 量化性能模型
37. 基于 OpenCL 的图像积分图算法优化研究
38. 基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究
39. 基于 OpenCL 的异构系统并行编程
40. 嵌入式系统中热备份双机切换技术研究
41. EFI-Tiano 环境下的 AES 算法应用模型

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
10. 基于小波变换与偏微分方程的图像分解及边缘检测
11. 基于图像能量的布匹瑕疵检测方法
12. 基于 OpenCL 的拉普拉斯图像增强算法优化研究
13. 异构平台上基于 OpenCL 的 FFT 实现与优化
14. 数据结构考题 - 第 4 章 串
15. 数据结构考题 - 第 4 章 串答案
16. 用 IPv6 编程接口实现有连接通信的方法

FPGA / CPLD:

1. 一种基于并行处理器的快速车道线检测系统及 FPGA 实现
2. 基于 FPGA 和 DSP 的 DBF 实现
3. 高速浮点运算单元的 FPGA 实现
4. DLMS 算法的脉动阵结构设计及 FPGA 实现
5. 一种基于 FPGA 的 3DES 加密算法实现
6. 可编程 FIR 滤波器的 FPGA 实现
7. 基于 FPGA 的 AES 加密算法的高速实现
8. 基于 FPGA 的精确时钟同步方法
9. 应用分布式算法在 FPGA 平台实现 FIR 低通滤波器
10. 流水线技术在用 FPGA 实现高速 DSP 运算中的应用
11. 基于 FPGA 的 CAN 总线通信节点设计
12. 基于 FPGA 的高速时钟数据恢复电路的实现
13. 基于 FPGA 的高阶高速 FIR 滤波器设计与实现
14. 基于 FPGA 高效实现 FIR 滤波器的研究
15.