

# Linux 下 IPv6 高级路由器的实现

胡文江 薛花

(内蒙古科技大学信息工程学院, 包头 014010)

**摘要** 介绍了 IPv6 的网络路由器的原理和在 Linux 上的具体实现方法。讨论了在 Linux 下如何组建 IPv6 网络试验平台,重点讨论如何将 Linux 工作站配置成路由器。然后介绍了静态路由和动态 RIPng 路由服务在 Linux 下的实现方法。

**关键词** 路由器 IPv6 Linux RIPng

文章编号 1002-8331-(2004)27-0117-03 文献标识码 A 中图分类号 TP393

## A IPv6 Router is Implemented under Linux

Hu Wenjiang Xue Hua

(College of Information Engineering, University of Science and Technology Inner Mongolia, Baotou 014010)

**Abstract:** This paper introduces the principle of IPv6 network router and the implementation under Linux. It discusses how to build a IPv6 network under Linux and how to configure IPv6 router. Then it introduces the static route and that the dynamic RIPng route are implemented under the Linux.

**Keywords:** router, IPv6, Linux, RIPng

### 1 引言

以 IPv4 为基础的 Internet 目前面临两个重大问题:地址空间的耗尽和路由表的过度膨胀。IPv6 正是用来取代 IPv4 的下一代网络协议,它解决了 IPv4 的地址问题,改善了 IPv4 的性能<sup>[1]</sup>。IPv6 的出现,使得很多相关的网络设备也要做出相应的改变以适应其 128 位长的地址。

支持跨路径选择的路由器是 Internet、Intranet、网际网、内部网必备的路由设备。路由器是通信子网中的通信节点,每个路由器都计算并维护一张路由表,并据此指导数据报前往最佳路径中的下一站,这便是所谓的路由。这样,经过互联网上所有路由器的通力合作,数据报就能够沿着一条“最佳”路径到达目的地。

目前, Linux 操作系统发展迅速、功能强大、内置网络支持,已逐渐成了 PC 平台上主流的 UNIX 操作系统。它具有性能稳定、网络功能超强、免费使用等特性。而且 Linux 内核从 2.2.x 之后开始正式支持 IPv6。基于以上 Linux 的优点,下面主要讨论如何使用最新的 Red hat Linux(2.4.18-14)来架设 IPv6 路由器,并且当网络由多个子网构成时,如何使用动态路由协议建立路由器上的路由表。

### 2 组建 IPv6 网络平台

#### 2.1 内核准备

在建立一个 Linux IPv6 路由器之前,所用的系统必须能支持 IPv6,也就是核心以及大部分的网络设定工具都要支持 IPv6。如果用最新的 Linux 套件,大部分的工具都会支持 IPv6,否则就需要手动升级这些工具。在设计中将会需要 net-tools 与 iproute2 这两个套件。Net-tools 包含了基本的网络工具,像 netstat、ifconfig、route 等等,而 iproute2 里面有 ip、ifcfg 以及 rtmon 等用于使用 Linux 核心中进阶网络功能的工具。如果编译

内核就要把 Networking options 选项里的 The IPv6 protocol 选成 yes(默认是 modules)。在重启系统之后,在登陆之前会发现有关支持 IPv6 的内核信息出现,这时该主机就是纯 IPv6 主机了<sup>[2]</sup>。具体要怎样编译内核不在论文的讨论范围之内。

由于所使用的实验系统 Red hat Linux(2.4.18-14)已把 IPv6 做成模块,所以直接加载即可。在/etc/modules.conf 里加一行 alias net -pf-10 ipv6。

#### 2.2 网络配置

把 Linux 配置成 IPv6 主机之后,在网卡上就有了 IPv6 的地址,但是它并不是全局地址,不能用来通信,需要为 IPv6 主机手工配置地址。现在新的 Linux 自带的网络配置工具都支持 IPv6,直接用 ifconfig 命令为以太网卡配置。如 ifconfig eth0 add 3ffe:ffff:0:1100::1/64 这里的 64 表示地址前缀长度为 64<sup>[3]</sup>。这时还要启动 IPv6 网络, Linux 的网络配置文件如下:

```
/etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=yes
IPV6FORWARDING=yes
HOSTNAME=ipv6router1
ONBOOT=yes
```

### 3 IPv6 路由器的架设

在使用 Linux 工作站担任两个网络之间的路由之前,首先在这台计算机上必须安装两块网卡,这两块网卡上的设置必须符合所属子网的相关设置。以图 1 所示的网络来说明。

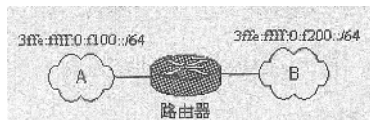


图 1 IPv6 路由器的架设

### 3.1 安装配置两块网卡

要担任路由器的计算机分别可以使用两块网卡连接两个子网络 A、B (如图 1) 3ffe:ffff:0:100::/64 和 3ffe:ffff:0:200::/64。两块网卡的地址要在相应的子网范围之内。下面是 Linux 的网络配置文件和网卡配置文件的具体内容。

路由器网卡 1 的配置文件是：

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPV6INIT=yes
IPV6ADDR=3ffe:ffff:0:100::1/64
```

网卡 2 的配置文件是：

```
/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPV6INIT=yes
IPV6ADDR=3ffe:ffff:0:200::1/64
```

其中 IPV6INIT=yes 是为了让在网卡上配置 IPv6 地址, 如果 IPV6INIT 不是 yes 就不会进行后面的地址配置, 而直接退出了。如果还想在一块网卡上配置多个 IPv6 地址, 还要在 ifcfg-eth\* 里面加一句 IPV6ADDR\_SECONDARIES=3ffe:ffff:0:300::1/64。

### 3.2 配置路由

首先在路由器上启动 IPv6 转发, 也就是在上面的网络配置文件里加入 IPV6FORWARDING=yes, 就说明计算机可以转发 IPv6 包。在路由客户机端, 要使缺省网关地址与本地路由器接口的 IPv6 地址相匹配, 在网络 A、B 的其余客户机除了在相应的网络范围内设置地址外, 默认网关要分别指向路由器, 即网络 A 中的客户机默认网关指向网卡 A 的 IPv6 地址, 网络 B 中的客户机默认网关指向网卡 B 的 IPv6 地址<sup>[4]</sup>。以网络 A 为例, 在网络 A 的客户机端的网络配置文件中加上默认网关: IPV6\_DEFAULTGW=3ffe:ffff:0:100::1%eth0。当网络 A 中的客户机所发分组的目标地址不是在本地图网时, 就可发到缺省网关地址, 交给路由器来进行转发。

## 4 多网络区段的路由器

当分组数据包的目标位于相邻网络时, IP 路由选择很直接, 就像上面介绍的一样。但实际情况要复杂得多。如果, 目的网络不是在直连的路由器上, 数据包便会不知道目的网在何处。如图 2 所示。

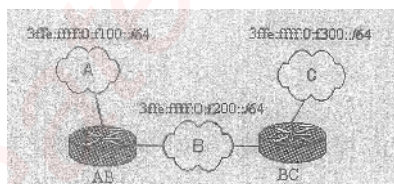


图 2 多网络区段路由器的说明

假设网络 C 上的计算机, 目标网络是“3ffe:ffff:0:300::1/64”。对连接网络 A 和网络 B 的路由器 AB 来说, 它只负责网络 3ffe:ffff:0:100::/64 和 3ffe:ffff:0:200::/64 的 IPv6 路由, 当

它看到信息上所指的域后, 它根本不知道 3ffe:ffff:0:300::/64 在何处。为了解决这个问题, 必须告诉路由器 AB, 如果有信息送到网络 C, 就先送到网络 B 与网络 C 的路由器 BC 上由它来负责传送数据。同理, 也必须告诉路由器 BC, 凡是要送到网络 A 的信息, 先送到路由器 AB 上, 由它来负责传送。所以路由器需要一个表来指明去目标网络的全部路径, 这就是路由表。路由表保存在 IPv6 路由器中, IPv6 查阅该表以确定需要的网络位于何处, 以便将分组发往该处。路由表由静态表和动态表两种。静态表是网络维护者人工保存的, 而动态路由表则是由路由选择协议自动支持的。静态路由器需要人工建立和更新路由表, 它不与动态路由器交换路径, 如果路径改变, 它们就自动地更新路由器的路由表, 并通知网络上的其它路由器进行修改。这样, 网络管理员只需进行很少的配置。所以, 在网络很小, 且与其它网络只有单个连接点又没有多余路由时, 可以使用静态路由选择。而在多网络区段的大规模网络中, 或这些条件不能全部满足时, 一般要使用动态路由选择。下面结合图 2 说明如何在 Linux 上设置静态和动态路由。

### 4.1 静态路由

要建立静态路由表就要使用 route 命令。首先来设置路由器 AB: 传送到网络 C 的信息先送到路由器 BC, 路由器 BC 连接在网络 B 的 IPv6 地址是 3ffe:ffff:0:200::2/64。所以要在路由器 AB 上执行下面的命令:

```
route -A inet6 add -net
3ffe:ffff:0:300::/64 gw
3ffe:ffff:0:200::2
```

该命令将会在路由器 AB 上的路由表中建立一个对应项。它的含义是, 凡是要送到网络 C 的信息, 先送到路由器 BC。同样的理由, 在路由器 BC 上执行下面的命令:

```
route -A inet6 add -net
3ffe:ffff:0:100::/64 gw
3ffe:ffff:0:200::1
```

凡是要送到网络 A 的信息, 先送到路由器 AB。这时候静态路由就配置好了。

### 4.2 动态路由

这里目标是用 Linux 打造一个“高级 IPv6 路由器”, 它必须能够利用动态路由协议工作。这些动态路由协议能够使路由器交换互换相关信息, 从而共享穿越网络时所用的那些路径——路由。这一点对于大型网络(比如 Internet)而言是异常重要的, 因为此时再用静态路由是根本不能实现的。

路由信息协议 RIPng (Routing Information Protocol next generation) 是 TCP/IP 网络上较常用的动态路由协议之一, 也称距离矢量路由协议, 它给出到达给定路径的跳数 (hops), 并将这一信息存入路由表中。RIPng 路由器会每 30 秒广播路由表并接收其它路由器的广播, 如有变化就修改自己的路由表以做到同步<sup>[5]</sup>。在这里使用支持 IPv6 路由的软件 Zebra, GNU Zebra 支持 BGP4、BGP4+、OSPFv3、RIPv1、RIPv2 以及 RIPng。

Zebra 是一个开放源代码的程序包, 可以从 www.zebra.org 网站下载 Zebra 的最新源程序。从源代码中进行软件安装, 就会发现使用的是一些普遍的安装过程。简介如下:

```
./configure
make
make install
```

如果熟悉 Cisco IOS ,那就能在短时间内掌握 Zebra ,因为两者极其相似。Zebra 的每个守护程序使用一个单独的 VTY ,这些 VTY 可以通过一个远程登录会话进行动态配置。所以 ,如果要设置 RIPng ,简单地远程登录到该 Linux 的 2603 端口 ;为了修改内核的路由表或设置路由协议间的再分发 ,可以远程登录到端口 2601 ,该 Zebra 守护程序充当内核管理器 ,管理其他的守护程序和系统本身之间的通信。

装好 Zebra 后 ,下一步要启动一些必要的程序。用以下命令完成 :

```
/usr/sbin/zebra -dk
/usr/sbin/ripng -d
```

第一个命令启动 Zebra ,该守护的程序实际上是用来更新路由表。-dk 告诉该程序作为一个守护程序运行(d) ,它的大部分时间在后台运行。K 是另外一个选项 ,告诉 Zebra 维护所有已配置的路由。它用来保证在测试 Zebra 的时候不会意外地删除路由表。一般情况下 ,设置路由和接口 ,需要将 ifconfig 和 route 这两条命令配合使用。而 Zebra 完全可以代替这种路由管理方式 ,使用起来更为简洁<sup>[6]</sup>。

配置 RIPng 时 ,需要 Telnet 到本地机器的 2603 端口。为进入特权模式 ,键入 enable( 正如在 Cisco IOS 中的一样 ) ,然后键入特权模式口令。接下来 ,用 config terminal 命令切换到配置模式。再键入 router ripng 来配置。图 2 所示 ,键入 network 3ffe ffff 0 f200 ::/64 ,告诉路由器将使用 RIPng 广播网络号长度为 64 的 3ffe ffff 0 f200 ::/ 的网络。一旦修改完成 ,用 end 命令从配置模式中退出 ,然后用 write 命令保存配置。新的配置文件就会保存在 /etc/zebra/ripng.conf 里。下面就是路由器 AB 的 RIPng 配置文件 :

```
!Zebra configuration saved from vty
! 2003/10/09 21 30 20
!hostname ripng
password zebra
log stdout
!
```

(上接 116 页)

等。又由于 JSDT 依据不同的应用可以通过 HTTP、SOCKET、LRMP 来实现 ,故此系统对 IP 网络没有限制性要求 ,用户可以针对自己的具体需求和网络来选用不同的底层通信机制。其网络部署如图 4 所示。

## 6 总结

论文分析了数据共享及 JSDT 技术 ,并在此基础上提出了基于数据共享的电子白板系统的体系框架 ,最后在此框架上给出了关键模块的具体实现。该系统已在“中国网上教育平台”<sup>[8]</sup>的“实时交互式答疑系统”<sup>[9]</sup>中得以应用。下一阶段将进一步优化系统的性能 ,并侧重于用户验证、网络传输延时、控制信息和数据同步方面的研究工作。(收稿日期 :2004 年 2 月)

```
interface lo
!
interface sit1
!
interface eth0
!
interface eth1
!
router ripng
network 3ffe ffff 0 f200 ::/64
!
line vty!
```

下面用命令 route -A inet6 ,就可以看到路由器 AB 的 IPv6 路由表了 ,这时注意到有了 3ffe ffff 0f300 ::/64 这一条路由 ,这就是通过动态路由协议 RIPng 加进去的路由。

## 5 结束语

以上讨论的路由实现方法均在 Redhat Linux 8.0 (内核版本 2.4.18-14 )下实际搭建并测试通过。试验证明在 Linux 操作系统下 ,通过简单的设置就可以将一台 Linux 工作站变成稳定的、功能完整的 IPv6 路由器 ,可以依据需要灵活实现静态路由、动态路由以及动静态路由相结合的路由选择方案。

(收稿日期 :2004 年 2 月)

## 参考文献

- 1.Deering S ,Hinden R.RFC 2460 :Internet protocol version 6 specification.1998
- 2.伍海桑 ,陈茂科等.IPv6 原理与实践[M].北京 :人民邮电出版社 ,2000
- 3.Peter Bieringer.Linux IPv6 HOWTO
- 4.卢泽新等译.Faraz Shamim IP 路由协议疑难解析[M].北京 :人民邮电出版社 ,2003-10
- 5.G Malkin.RIPng for IPv6[S].RFC 2080 ,1997
- 6.http :www.zebra.org

- 1.Steven Mc Canne.A Distributed Whiteboard for Network Conferencing[R].Technical Report LBL ,1995
- 2.李向阳 ,鲁东明.共享电子白板系统设计与实现[J].计算机工程与应用 ,1999 ,35(7) :98-100
- 3.葛晓虎 ,尹浩 ,朱耀庭等.基于 CSCW 的远程教育中电子白板的实现[J].计算机工程 ,2002
- 4.高旭 ,沈苏彬 ,赵桂平等.多媒体会议系统电子白板工具的设计与实现[J].小型微型计算机系统 ,2000
- 5.Java Shared Data Toolkit.http ://Java.sun.com/products/Java-media/jsdt
- 6.Light-weight Multicast Reliable Protocol.http ://webcanal.inria.fr/lrmp
- 7.Deering SE Host extensions for IP multicasting[R].Internet Engineering Task Force( IETF ) ,TechRep :RFC1112 ,1989
- 8.杨贯中.中国网上教育平台试点工程项目可行性研究报告[R].技术报告 ,2000
- 9.杨贯中 ,陆绍飞.实时交互式答疑系统[R].技术报告 ,2002

# 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](#)
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](#)

## 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](#)

## 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](#)
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](#)

## 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](#)

## 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](#)
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](#)

## 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.

## 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.