

VxWorks图形用户界面开发中双缓冲技术应用

Applying of double buffering technology in VxWorks

(武汉理工大学¹;湖南科技大学²) 陈君^{1,2}唐秀明²
Chen,Jun Tang,Xiuming

摘要 介绍和分析了嵌入式实时操作系统 VxWorks及媒体库 WindML的结构和特点。结合实际应用 WindML提供的 API函数对 VxWorks操作系统进行用户图开界面开发 研究在 VxWorks开发图形用户界面的技术难点问题 并给出了问题解决方案和程序实现。

关键词 实时操作系统 图形用户界面 ; VxWorks; WindML双缓冲
中图分类号 : TP311 文献标识码 : A

Abstract: This paper analyzes and introduces the structure and characteristic of a embedded real-time operating system VxWorks and its media library WindML. Applying the API functions supplied by WindML, the paper gives the program of using double buffer technology which can realize graphics outputting and eliminate its flickering on VxWorks Operating System.

Key Words Real time operation system GUI VxWorks WindML Double Buffering

前言

在计算机技术和信息技术高速发展的今天 计算机和计算机技术大量应用在人们的日常生活中。现代的计算机早已超出早期计算机的概念 广泛应用的实时嵌入式系统便是其中的一种。嵌入式系统中的操作系统是整个系统的核心部分 操作系统性能的各项性能指标直接影响到整个系统的性能。 VxWorks是典型的嵌入式实时多任务操作系统 (RTOS) 具有高可靠性、高实时性、可裁减性和对于异步、并行的事件可以在严格的时限内作出响应等特点。为了提高系统的易用性 , VxWorks提供了一个媒体库— WindML(Wind Media Library) 利用它可以更快的、更简单地开发资源消耗更小的图形用户界面。本文结合 VxWorks在某设备信息化改造项目中的应用 主要利用 WindML的双缓冲技术解决了嵌入式系统图形开发中遇到的图像闪烁抖动的问题。

特点包括 快速多任务切换、抢占式任务调度、任务间通信手段多样化、中断延时小、网络流量大等。它还能与 Wind River公司的其它产品以及 32多个公司的合作伙伴的产品联合使用。它的特点、功能及主要结构可以参见参考文献。

WindML2是嵌入式实时操作系统 VxWorks的一个可裁减的多媒体库 其本身也具有可裁剪功能和可配置性 以适应不同的应用要求 , WindML可以为 VxWorks操作系统下开发的软件提供基本图形、视频和音频方面的操作。 WindML API函数库提供对多种 CPU结构和操作系统适用的图形硬件接口。同时 WindML还提供操作输入设备和处理输入设备事件的功能。 WindML由两部分组成 软件开发包 (SDK)和硬件开发包 (DDK) 软件开发包提供多媒体 二维图形、窗口、色彩管理、视频、位图等 应用程序接口 (API) 事件服务、内存管理、扩展接口、设备管理 硬件开发包提供普通硬件的驱动。其层次结构如图 1所示 :

实时操作系统 VxWorks和 WindML

简介

VxWorks是由美国 WRS (Wind River Systems Inc.) 公司开发的一套微内核、高可靠性、可裁减的嵌入式实时操作系统。它以其卓越的可靠性、可裁减性、实时性、开放性和易用性 被广泛地应用在通信、军事、航空、航天等实时性要求极高的领域中 如 卫星通讯、弹道制导、飞机导航等。 VxWorks微内核 Win是一个具有较高性能的、标准的嵌入式操作系统内核 其主要

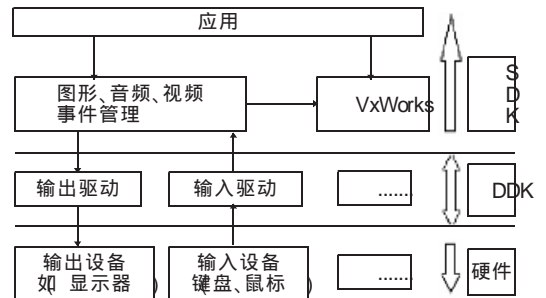


图 1

图形用户界面开发

3系统设置

在应用 WindML (Wind Media Library) 进行事件响应和图形开发之前。根据具体的应用需要 必须对目标机的 WindM 库进行设置。一般采用在 Tornado 的 WindM 设置工具设置 , 打开 Tornado>Tools>WindM 设置目标机的类型、输入设备、通用图形库位图、字体、音频等 保证实际项目开发的需要。

3 程序总体框架

在进行某设备信息化改造项目应用方面的编程有一个程序总体框架 遵循此框架能减少编程时的错误 使在开发过程有清晰的编程思路 提高编程的效率。在实际应用中的总体框架如下 :

初始化程序环境 `uglInitialize ()` 完成 UGL 和 WINM 库函数初始化。

获得到设备驱动 `uglDriverFind` (包括图形设备、字体、鼠标、触摸屏等)。

创建图形设备场景 `uglGcCreate` (初始化颜色并分配定义的 RGB. `uglColorAlloc()`)

构建图形用户界面 `uglBitmapCreate` (显示图形用户界面 `uglBitmapBlit()`)

图一个无限循环 (UGL_FOREVER) 构成事件处理的机制。所有与用户的交互 (鼠标、键盘等) 由它来产生或完成 而所有的消息也由 `uglEventGet` 得到 收到命令后再跳出循环 执行指定的程序。

完成对所有资源的释放并退出 如 `uglBitmapDestroy()`, `uglGcDestroy()`, `uglDeinitialize` 等 初始化过后的资源都需要释放 否则导致内存的泄露。

4 关键技术研究

WindM 提供了 2 图形的解决方案 但在某设备信息化改造项目中。由于需要实时采集、显示虚拟仪表图形信息、驾驶员终端显示综合控制信息、车内各设备状态信息车长命令、报警等 都需要实时绘制画面 每次采集到数据和各信息互相切换都要刷新画面。画面闪烁、抖动比较剧烈 因此可以把在显示时不变化的信息以位图背景的形式保存 , 调用 (`uglDefault-`

`BitmapSet`) 实现 在此背景上来动态刷新变化的仪表指针、设备状态、报警等 这样在刷新画面时数据量减少 画面的效果有所改善 但是由于采集到的数据、状态等是实时变化 画面闪烁、抖动存在 视觉效果不佳 本文应用 WindM 的双缓冲技术来解决这一问题。

双缓冲技术是指不改变可见视频缓存区起始地址 而是在内存中开辟一块与屏幕一样大小的存储区域 作为缓冲屏幕 同时将下一帧要显示的图像绘制到这个缓冲屏幕上面 在显示的时候将虚拟屏幕中的数据直接复制到可见视频缓存区里。由于图形绘制过程是在不可见的虚拟屏幕中进行 所以可以消除屏幕的闪烁、抖动。

以虚拟仪表图形信息显示为例 在具体的实现中 ,

让仪表指针、设备状态、报警等的过程在 `Off_Screen` (后台) 实现。即在绘制当前图形信息的时候 调用函数 `uglPageDrawSet` (另外指定绘制下一图形操作的页面缓冲。) 在绘制完成后 再通过 `uglPageVisibleSet` 把 `Off_Screen` 的图像拷贝到前台 即实现图像的刷新 这就可以有效的解决图像闪烁、抖动问题。具体实现如下 :

```
page[0] = UGL_PAGE_ZERO_ID;
page[1] = uglPageCreate(devId); 创建双缓冲页 ( )
bgBitmap = uglBitmapCreate (devId, &XNBPDib,
UGL_DIB_INIT_DATA, 0, UGL_NULL); 位图数据
信息创建位图
uglDefaultBitmapSet (gc, bgBitmap); 设置 bgBitmap
为画图背景
uglPageDrawSet (devId, page[ 1 ]) 设置活动的绘
图双缓冲页 ( )
uglBitmapBlit (gc, DisplayBitmap, 0, 0, 16, 16,
UGL_DEFAULT_ID, 0, 0);
..... 在变化的位置绘制的位图和其它
具体图形绘制
uglPageVisibleSet (devId, page[1]) 图像在前台显示
uglPageDrawSet (devId, page[0]); 在后台绘制下
一图像
.....
uglPageVisibleSet (devId, page[0]) 图像在前台显示
uglPageDrawSet (devId, page[1]); 在后台
绘制下一图像
.....;
uglPageDestroy (devId, page[1]) 释放缓冲
```

结束语

应用 WR 公司提供的媒体库—— WindML (Wind Media Library) 进行图形用户界面的开发是嵌入式系统 VxWorks 输出的一种重要方法。应用本文给出的例程可以很好的消除 VxWorks 系统下图形用户界面时的闪烁、抖动问题 得到好的视觉效果。该方法已经在基于 VxWorks 开发的某设备信息化改造项目中得到了良好的应用。

参考文献

陶霞 何强 . MAX1245 片在嵌入式数据采集系统中的应用 [J] 计算机信息 , 2005, 2: 111- 113
陶国庆 等编著 . VxWorks 嵌入式软件开发 [M] 北京 机械工业出版社 , 2003
陶方敏 编著 . VxWorks 高级程序设计 [M] 北京 清华大学出版社 2004
作者简介 陈君 男 , 1977 年 1 月出生 硕士研究生 通信与信息系统专业 助教 湖南科技大学信息学院 研究方向 嵌入式系统 计算机网络 ; Email: jchen1@hnust.edu.cn
唐秀明 女 , 1977 年 10 月出生 助教 湖南科技大学信息学院 研究方向 电力系统及自动化 见 (139)

```
BYTE SyncByte;同步字
WORD RecvBuffLen; 接收缓冲区长度
WORD TranBuffLen; 发送缓冲区长度
BYTE ProWordWidth; 协议字宽
void (*FrameProcess)(void); 帧处理函数指针
void (*InitProcess)(BYTE Num);初始化函数指针
```

```
};
加入新协议时应进行初始化 如对于 USE协议 :
void InitUSERProt(void)初始化 USE协议
{ ProtocolDisp [USERProtNo].ProtocolType=USER-
ProtNo; 协议类型
.....
ProtocolDisp [USERProtNo].FrameProcess =USER-
ProtProcess;帧处理函数指针
ProtocolDisp[USERProtNo].InitProcess=InitUSER; //
初始化处理函数指针
}
```

USERProtProcess是需要具体编写的帧处理函数，也是协议的核心。InitUSER是协议私用变量初始化函数。加入一个新协议除用到公共数据结构外，如实时数据库、参数数据库等，还需协议私用变量。定义在一个结构中，用InitUSER函数实现其初始化。把上述协议描述结构初始化函数void InitUSERProt(void)加到总初始化函数void InitProcess(void)：

```
void InitProcess(void)
{
.....
#ifdef USERProtNo
InitUSERProt();
#endif
}
```

(编写帧处理函数)

具体帧处理函数包括两大部分：解帧函数和组帧函数。解帧即解析上位机传输的应用协议并转入相关执行程序；组帧即组织本机所需上传应用信息。对USE规约：

```
void USERFrameProcess(void)
{ USERUnlockFrame ();//USE协议的解帧处理
函数
USERFraming();//USE协议的组帧处理函数
}
```

协议的具体工作在两大函数中展开。另外，在.h文件中编写协议私用变量结构，包括上下行数据计数、定时校时计数及校验和等，此不赘述。

结语

本文根据数据采集系统的要求，研究并给出了数据采集终端网络接入模块的功能结构模型，设计了适用于数据采集系统的网络传输协议，并结合编程加以实现。本文的讨论为数据采集系统的网络化提供了一

个很好的解决方案。由于主从处理器之间接口电路采用了可编程器件FPGA设计，具有一定的通用性，能够满足多种数据采集终端。设计的网络传输协议通过参数信息来协调连接的双方，以满足不同需求的应用。测试表明，数据传输实时正确，运行稳定。

参考文献：

[1] 玉东 王非非. Linux操作系统结构分析 [M]. 西安: 西安电子科技大学出版社, 2002.

[2] 李驹光. AR应用系统开发详解 - 基于S3C4510的系统设计 [M]. 北京: 清华大学出版社, 2004.

[3] DAVIDCOM公司. DM9162技术手册. 2001-08.

[4] 佛 李伟生. 嵌入式系统在曙光4000大规模机群监控系统中的应用 [J]. 计算机信息, 2005, 5: 57-59.

作者简介: 杨健男, 1965年生, 北京工业大学博士研究生, 高级工程师, 主要从事网络监控、嵌入式技术、计算机应用与研究。发表论文十余篇, 取得多项省级成果。获省部科技进步奖两项。E-mail: yangjian@e-mails.bjut.edu.cn
张慧慧女, 1944年生, 北京工业大学教授, 博士生导师。

About the authors: Yang Jian, male, received the B.S. degree in instrument engineering from Shanghai Jiaotong University, Shanghai, China, in 1988. Now he is Ph.D. candidate of Beijing University of Technology, and is also a senior engineer. His major interests is in network supervising, embedded technology and computer application. Zhang Hui-hui, female, supervisor of Ph.D. candidate of Beijing University of Technology, her major interests is in mechanical engineering.

(100022北京工业大学机电学院 杨健 张慧慧)

(250014济南山东师范大学图书馆 于东琴)

(College of Mechanical Engineering & Applied Electronic Technology, Beijing University of Technology, Beijing 100022, China) Yang, Jian Zhang, Huihui

接 9页)

(430070湖北武汉 武汉理工大学信息学院 陈君)

(41120湖南湘潭 湖南科技大学信息学院) 陈君

唐秀明

(College of Information Engineering, Wuhan University of Technology, Hubei Wuhan 430070) Chen, Jun
College of Information Engineering, Hunan University of Science and Technology, Hunan XiangTan 411201) Chen, Jun Tang, Xiuming

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

RT Embedded <http://www.kontronn.com>

16. [基于 UEFI 的 Application 和 Driver 的分析与开发](#)

Programming:

1. [计算机软件基础数据结构 - 算法](#)