基于 Android 平台的软件自动化监控工具的设计开发

郑云卿　黄　琦

安徽大学电气工程与自动化学院　安徽 合肥 230601）

摘　要　主要研究搭载 Android 操作系统的智能手机软件自动化监控工具的开发，介绍 Android 开发平台，分析软件需求以及实现方法。针对软件监控所需数据的采集，介绍了实现方法和步骤，并针对不同数据的作用加以分析，完成软件设计，实现了提出的自动化监控的需求。

关键词　Android　Java　智能手机　自动化监控

中图分类号　TP311　　**文献标识码**　A　　DOI 10.3969/j.issn.1000-386x.2013.02.062

DESIGNING AND DEVELOPING ANDROID-BASED SOFTWARE AUTOMATED MONITORING TOOLS

Zheng Yunqing　Huang Qi

(School of Electrical Engineering and Automation, Anhui University, Hefei 230601, Anhui, China)

Abstract　In this paper our study focuses on the development of automated monitoring tools for software of smart phone equipped with Android operating system. Therefore the article introduces the Android development platform, the analyses on software requirements and its realisation approach. For collecting the data required by the software monitoring, in this paper we also introduce the implementation methods and steps, analyse the functions of different data, and finally complete the software design. The automated monitoring demand presented in the paper is then achieved.

Keywords　Android　Java　Smart phone　Automated monitoring

0　引　言

随着信息化和互联网技术的发展，智能手机正在越来越多地被消费者接受和使用[1]。而在众多的智能手机系统中，Android 以开源的优势以及系统本身不断的优化更新，迅速占领市场，据统计，2011 年 Android 手机全球出货量增长近 250%，市面上几乎近一半的智能手机都采用 Android 系统，第四季度 Android 手机全球出货量为 2.378 亿部。Android 手机设备的迅速发展，带动着整个 Android 软件行业的发展，越来越多的开发者加入 Android 游戏和应用的开发行列，软件公司纷纷推出在 Android 平台上的经典软件。

Android 系统的开放使得 Android 软件开发变得容易，任何开发者，无论是专业公司还是个人都可以开发出属于自己并给别人分享的应用软件，这也使得 Android 软件日益庞大，对于某一个功能需求，Android 手机用户可能有太多的选择，而这些软件有的比较精致，有的却比较粗糙，用户在使用过程中，并不能完全掌握这款软件的行为，比如内存占用，CPU 占用，是否存在上传流量等等。本文的目的是针对 Android 平台开发一款可以提供自动化软件监控的工具，针对待测软件，记录待测软件的行为数据，并记录日志文件以分析待测软件所占用的资源是否合理，是否存在恶意上传流量等等。

1　Android 开发简介

Android 是由 Google 公司推出的，以 Linux 为核心的开放源码的操作系统[2]，可以运行在手机、平板电脑、GPS、播放器等各种手持信息通信工具上。直到本文为止已经从 Android 1.0 版本更新推出到 Android 4.0 版本。无论是操作流畅度还是硬件设备的提升，都在不断地提高，也因为如此的发展，几乎所有设备商都投入到 Android 手机的开发中。

Android 开发环境可以是 Java，也可以是 C ++ 等经典语言，一般使用 Java 进行开发，开发平台使用 Eclipse 软件（开放源代码的、基于 Java 的可扩展开发平台）并下载 Android 提供的 SDK（软件开发工具包）进行开发。Android 的 SDK 提供可视化的模拟器，以及 DDMS 工具（Dalvik 虚拟机调试监控服务）用于实时查看软件运行情况以及文件管理、截屏等

操作，即使没有真机也可以非常直观迅速地进行开发，并基本符合真实场景[3]。本文所开发的自动化监控工具正是在以上的基础上进行开发的，并在真机环境下顺利通过测试，投入使用。

2 自动化监控软件编写

2.1 软件需求分析

对于本文设计的自动化监控软件[4]，需要提供选择可以监控的对象和监控记录的数据，并需要在后台记录，因为只有待测软件进入前台显示操作时采集的数据才是最准确的，由于 Android 系统的特点，非前台显示的进程都处于挂起阶段，内存会释放掉一部分，而后台显示占用的那部分内存实际上是不运行的，所以必须记录运行状态下的数据才能正确反映此软件的信息。

我们可以先描绘出整个软件的工作流程，如图 1 所示。

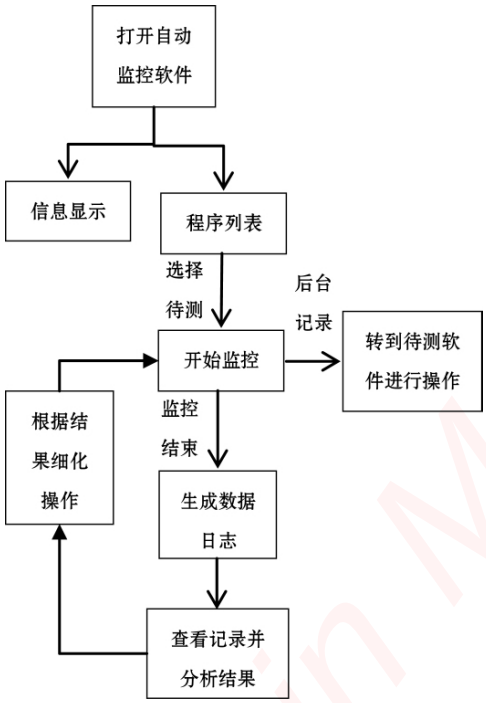

图 1 自动监控工具运行流程

本文设计的监控工具必须采集很多种信息才能最终判断待测软件的性能，并且尽量满足当前所有主流 Android 版本以及各个设备商自行二次开发的 Android 手机系统。下面将分析具体需要采集的数据及其方法。

2.2 数据采集

判断一个软件在手机上的运行情况，一般由以下几个方面着手 内存、CPU 使用率、CPU 占用时间、流量、消耗的电量等信息。Android 是基于 Linux 内核的系统，因此本文采集的数据很多方式都通过读取 Android 的系统文件信息来获取。

2.2.1 内存信息获取

对于手机来说需要记录的内存信息有手机的总内存、当前剩余可使用的内存、以及待测软件占用的内存。手机的总内存可以通过读取/proc/meminfo 取得，第一行即为手机总内存如图 2 所示。

```
C:\Mobile Phone DEV\tools>adb shell cat /proc/meminfo
MemTotal:         190368 kB
MemFree:            8560 kB
Buffers:            4868 kB
Cached:            62060 kB
SwapCached:            0 kB
Active:            81852 kB
Inactive:          80252 kB
Active(anon):      43168 kB
```

图 2 手机总内存的获取

而对于手机剩余内存，本文采用通过 Android 提供的接口函数来实现[5]：

```
ActivityManager am = (ActivityManager)mContext.getSystemService(Context.ACTIVITY_SERVICE);
ActivityManager.MemoryInfo mymem = new ActivityManager.MemoryInfo();
am.getMemoryInfo(mymem);
return mymem.availMem / 1024;
```

这里返回的值即为当前剩余的内存。待测软件使用的内存情况就比较复杂，对于 Android 系统来说，所有运行的软件既是独立的又是共享的，在使用过程中有很多调用方法和数据都是共享的，而这些共享的内存是不会被系统回收掉的，而软件本身占用的内存部分会被系统不断回收，又不断申请。对于一个软件的内存有四个数据值得深入研究：

虚拟耗用内存 VSS (Virtual Set Size)这个值一般情况下比较大，因为它包括了所有共享库占用的大小，所有在程序运行过程中可能涉及到的内存都被归为 VSS 中，所以对于判断待测运行情况来说不够标准。

包含共享库的实际使用物理内存 RSS (Resident Set Size)：RSS 内存显示的也包括了共享库占用的内存，此内存值是实际运行时程序占用的以及运行时占用的共享库的大小的总和，即使只载入一次的共享库也被纳入此内存值。

只包括共享库实际使用的物理内存 PSS (Proportional Set Size)这个值的大小是在程序运行过程中，系统分配给进程需要的共享库大小的值，仅为共享库的值，在单独对比软件资源消耗时比较有用。

进程独自占用的物理内存 USS (Unique Set Size)这个值仅为系统分配给进程的物理内存，即进程在运行时需要的内存大小，而调用的共享库的值不计入此内存值，因此单独判断待测软件对于物理内存大小的需求比较有意义。

这些值的获取分为两部分，首先通过调用 Linux 的系统命令 Top 即可采集到当前运行程序的 VSS 和 RSS 数据。同时，由图 3 可以看到，通过 top 命令也可以采集到进程的身份标识 PID，对应程序包名 Name 以及程序占用 CPU 的大小。

```
C:\Mobile Phone DEV\tools>adb shell top -n 1 -s vss

User 7%, System 5%, IOW 0%, IRQ 0%
User 24 + Nice 0 + Sys 19 + Idle 278 + IOW 0 + IRQ 0 + SIRQ 0 = 321

  PID CPU% S  #THR     VSS     RSS UID      Name
   83   4% S    48 199108K  25740K system   system_server
  143   0% S    12 152160K  18584K nobody   com.htc.launcher
  135   0% S    19 151852K  19288K misc     android.process.acore
  522   0% S    11 141116K  21364K app_40   com.iflytek.inputmethod
```

图 3 Top 命令能够获取的数据

对于 PSS 和 USS 内存大小的获取，分两种情况，Android 在 2.2 以上的版本提供了接口函数，只要调用即可获取到数据，而

2.2 版本以下的 Android 系统可以通过读取系统文件的形式获取内存大小。

2.2.2 CPU 使用率

同样 CPU 使用率分为手机所有进程占用的使用率以及待测软件占用的使用率。在系统文件/proc/stat 中对于 CPU 的使用情况有很多信息,如图 4 所示。

```
C:\Mobile Phone DEV\tools>adb shell cat /proc/stat
cpu  88772 2877 29328 666877 233 4 123 0 0
cpu0 88772 2877 29328 666877 233 4 123 0 0
intr 1676159 3246 0 0 0 0 277 0 1070751 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ctxt 3167331
btime 1321493445
processes 5987
procs_running 1
procs_blocked 0
```

图 4 CPU 使用情况的信息

CPU 总使用率通过采集一定时间隔的/proc/stat 文件的数据信息,并进行处理即可得到,在图 4 中,CPU 因为是单核,所以只需要考虑第一行的数据,这一行的数据信息分别指的是:

user 88772 处于用户态的运行时间,不包含没有良好状态的进程。

nice 2877 良好值为负的进程所占用的时间。

system 29328 系统内核运行的时间。

idle 666877 除接口等待时间以外的其它等待时间。

iowait 233 接口等待时间。

irq 4 硬中断时间。

softirq 123 软中断时间。

guest 0 其他用户占用时间。

因此采集到一定时间间隔的 CPU 使用数据进行处理即可得到当前 CPU 使用率。

Cpu_use_time = user + nice + system + idle + iowait + irq + softirq + guest

Cpu_free_time = idle

Cpu usage = (idle2-idle1)/(cpu2-cpu1)*100

对于进程占用的 CPU 使用率,通过上一节中的 Top 命令就能够获得,只需要记录存储数据即可。

2.2.3 CPU 使用时间

进程占用 CPU 的使用率只能反映当前瞬间进程对 CPU 资源的消耗,而在一段时间内,进程占用 CPU 总使用时间才能反映出待测软件对于系统的消耗,在 Android 系统文件/proc/pid/stat 中也有许多的数据,而需要的进程占用时间只有两种:

utime 进程占用 CPU 用户态运行的时间。

stime 进程处于 CPU 核心态运行的时间。

而这两种进程占用 CPU 的运行态指的是申请占用的级别的高低,stime 是最高级别的申请,而 utime 是一般级别的申请,两种时间的相加即为待测软件占用 CPU 的时间。

同样通过前面 Cpu_use_time 与 Cpu_free_time 之和就能得到 CPU 运行的总时间,那么进程占用 CPU 的时间与 CPU 总运行时间之比就可以间接得到待测软件消耗的资源大小。

2.2.4 流量大小

手机流量大小分为 Wi-Fi 流量和 GPRS 流量,对于用户来说,Wi-Fi 一般都属于免费的,而 GPRS 是收费的,因此本工具在设计时采集的数据是 GPRS 的流量大小。

而对于待测软件流量来说,需要知道的有上传和下载的流量,同时还有手机本身全部的上传和下载的流量,这两组数据的对比既可以分析出待测软件是否存在恶意上传,也可以分析出手机本身有没有未通知用户的流量通信。

通过读取/proc/net/dev 系统文件即可得到当前所有流量信息,如图 5 所示。

```
Inter-|   Receive                |  Transmit
 face |bytes    packets multicast|bytes    packets
    lo:  859055   12362         0   859055   12362
dummy0:       0       0         0        0       0
rmnet0:  311206     844         0   194415    1078
```

图 5 流量信息

如图 5,rmnet0 一行所列出的数据就是流量信息,在椭圆框中的数据分别是手机下载和上传的流量数据,单位是 byte。

待测流量信息在 Android 2.2 版本之后就有接口函数可以调用,但是因为本工具设计的目的是尽量满足市面上所有 Android 版本,所以仍然采用读取系统文件,写成完整的类进行调用。进程的上传下载分为两个文件:

/proc/uid_stat/uid/tcp_rcv 为进程下载流量大小。

/proc/uid_stat/uid/tcp_snd 为进程上传流量大小。

其中文件位置需要知道 UID 用户标识符才能采集到数据,UID 是 Android 系统配置给用户的标识符,进程之间可以共享 UID,通过 ActivityManager. RunningAppProcessInfo 即可获得当前运行的进程的 UID 等信息。

2.2.5 电量

对于电量,同样通过读取系统文件夹 /sys/class/power_supply/battary 可以列出很多关于电池的信息,如图 6 所示。

```
cd battery
$ ls
ls
uevent
subsystem
device
power
type
status
health
present
technology
capacity
batt_id
batt_vol
batt_temp
batt_current
charging_source
charging_enabled
full_bat
smem_raw
smem_text
delta
full_level
$
```

图 6 电池信息

通过此文件夹里的文件可以读到当前电池电量百分比、充电状态、当前电池温度等等信息,本文设计的工具只取了电量百分比作为参考数据。

2.3 软件实现以及测试结果

所需要的数据获取写成模块化的类进行调用,用户界面需要进行一些相关设置即可实现工具的功能,本文设计的自动化监控工具是运行于 Android 手机上的,因此界面也都使用 An-

droid 一般工具界面。本工具分为四个界面，通过选项卡的形式进行选择页面切换。分成首页显示信息、程序、设置、关于四个界面，如图 7 所示。

图 7　工具界面

首页自动刷新一些基本的信息，程序页面会显示出当前运行的所有程序列表，通过程序列表进行待监控软件选择，在设置页面完成基本设置后就可以开始自动监控待测软件了，所有信息被记录在存储卡上，并以一定的格式保存，在后期处理数据时既可通过本工具直接查看，也可将数据直接拷入到 Excel 等表格处理工具查看。

利用本工具对某 Android 软件进行监控，通过对待测软件一段时间的使用，后台记录了此段时间的数据，通过工具浏览这些数据如图 8 所示，可以看出此款软件运行时的状态数据，可以看到软件运行时占用 CPU 资源少，但占用的内存比较大，没有流量消耗。这些数据既可提供作为专业测试人员进行分析，也可作为 Android 手机用户的参考数据，实际操作中非常方便。

```
TIME NAME PID CPU VSS RSS PSS
USS uTime sTime UNUSED TOTAL
MEM CPU TOTAL CPU Time
uidDOWNLOAD uidUPLOAD
phoneDOWNLOAD phoneUPLOAD
BATTARY
2012-02-28 16:25:43 com.spb.
wirelessmonitor 18594 0% 106044KB
28660KB 9582KB 8032KB 17.71s 3.43s
95100KB 487348KB 9.52% 43277.64s -
- 0.00k 0.00k 60%
2012-02-28 16:25:47 com.spb.
wirelessmonitor 18594 0% 106044KB
28660KB 9577KB 8032KB 17.71s 3.43s
93340KB 487348KB 14.29% 43282.31s -
- 0.00k 0.00k 60%
```

图 8　测试数据结果

3　结　语

本文基于 Android 系统开发的自动化监控工具，可以运行于市面上大部分 Android 手机终端，可以自动化采集记录数据，数据经验证都真实可靠，可以作为专业的分析数据使用，对于专业的测试人员来说，减轻了他们的工作负担，提高了工作效率，同时得到的数据也更具有价值。而对于普通用户来说，对于一款 Android 软件，通过本工具监控的数据，可以很清楚地知道这款软件在运行阶段是否一直处于资源占用高峰，程序进入后台后，是否仍然不断消耗系统资源、偷传流量等。作为一款 Android 平台的监控，本工具非常具有应用市场，而随着工具的不断完善，将提供更多可靠的分析数据，因此，可以看到本文设计的 Android 软件自动化监控工具对于监控分析 Android 软件具有非常重要的作用。

参　考　文　献

[1] 童承凤，胡庆. 基于 Android 平台的双网双待的研究与设计 [J]. 计算机应用与软件，2012，29(2)：250-253.
[2] 左天军，左圆圆，陈平. Linux 操作系统的实时化分析 [J]，计算机科学，2004，31(5)：110-112.
[3] Ed Burnette. Hello，Android Introducing Google's Mobile Development Platform [M]. American Pragmatic Bookshelf，2009.
[4] 付剑平，陆民燕. 软件测试性定义研究 [J]. 计算机应用与软件，2010，27(2)：141-143，153.
[5] 李宁. Android 开发权威指南 [M]. 人民邮电出版社，2011.

（上接第 215 页）

据分配给多个线程并行处理以生成和扩充码书。实验结果表明 ACVQ 算法的并行结构极大的加速了压缩进程，码书在初始化选取和优化时的自适应性和可扩充性提高了码书的质量，适用于较大型的体数据在 GPU 中的压缩。

若扩大 Grid 网格规模以提高 GPU 每次并行处理数据量，执行效率还会更高，较之传统 VQ 方法，ACVQ 因其速度优势，更适合线上实时的数据压缩和解压绘制，后续的研究是将 ACVQ 方法用于图像数据、医学数据等其他数据上，检验其优势。另一个研究重点是思考如何有机地融合其他方法的优点，例如文献 [10] 中很好地融合了生长型神经气以提高学习型矢量量化的训练速度。再进一步减少压缩时间和保证码书质量之间谋求平衡点。

参　考　文　献

[1] Linde Y，Buzo A，Gray R M. An algorithm for vector quantizer design [J]. IEEE Transactions on Communications，1980，28(1)：84-95.
[2] 赵利平，肖德贵，李肯立，等. 一种高效体数据压缩算法及其在地震数据处理中的应用 [J]. 计算机辅助设计与图形学报 2009，21(11)：1606-1611.
[3] 陈善学，徐皓淋. 基于子矢量技术的矢量量化码字快速搜索算法 [J]. 重庆邮电大学学报 自然科学版，2010，22(3)：302-306.
[4] 木春梅，韩守梅. 一种基于不等式的快速码字搜索算法 [J]. 电子学报，2010，38(B02)：218-220.
[5] 李碧，林土胜. 初始码字间距最大化的矢量量化码书设计算法 [J]. 小型微型计算机系统，2009，30(4)：780-783.
[6] 姜来，许文焕. 模糊强化学习型的图像矢量量化算法 [J]. 电子学报，2006，34(9)：1738-1741.
[7] 厉旭杰. GPU 加速的图像匹配技术 [J]. 计算机工程与应用，2012，48(2)：173-176.
[8] 孔勇平. 矢量量化 LBG 算法的研究 [J]. 硅谷，2008，6：39-40.
[9] 乔阳，潘志斌，乔瑞萍. 码书排序对快速码字搜索算法性能影响的分析 [J]. 中国图象图形学报，2010，15(8)：1182-1188.
[10] 王修君，沈鸿. 一种基于增量学习型矢量量化的有效文本分类算法 [J]. 计算机学报，2007，30(8)：1277-1285.

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

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

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

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

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

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

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 NiosⅡ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

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