嵌入式 Linux 技术研究

朱玮玮　卢　虹　杨建明　东南大学（210096）

Abstract

Nowadays,embedded system is becoming one of the most active developing areas of IT industry,and Linux is playing an important role in embedded systems.In this paper,based on the introduction of the history and characteristics of embedded (linux) system,the technological features of several embedded linux distributions such as ETlinux,μClinux and several embedded GUI solutions such as Qt/Embedded,MiniGUI are discussed.At last,the importance of Linux in both research and application is pointed out.

Keywords: Embedded system,ETlinux,μClinux,MMU,Qt/Embedded,MiniGUI

摘　要

嵌入式系统正成为当今 IT 业发展最为迅速的领域，Linux 在嵌入式领域占有举足轻重的地位。本文在分析嵌入式(Linux)系统的历史和特点的基础上，讨论了 ETlinux、μClinux 等嵌入式 linux 发行版以及 Qt/Embedded、MiniGUI 等嵌入式图形用户界面的技术特色，指出嵌入式 Linux 在理论研究和实际应用中的重要意义。

关键词：嵌入式系统，ETlinux，μClinux，MMU，Qt/Embedded，MiniGUI

0　引言

嵌入式系统是一个难于严格定义的概念。一般来说，凡是带有微处理器的专用软硬件系统都可以称为嵌入式系统。现今主流的嵌入式操作系统有 VxWorks、QNX、WindowsCE、LynxOS、DOS、Neucleus、GreeHills 等，它们的市场份额如图 1 所示。其中，VxWorks 使用最为广泛、市场占有率最高，其突出特点是实时性强（采用优先级抢占和轮转调度等机制），可靠性和可剪裁性也相当不错。QNX 是一种伸缩性极佳的系统，其核心加上实时 POSIX 环境以及一个完整的窗口系统还不到一兆。相比之下，Microsoft WinCE 的核心体积庞大，实时性能也差强人意，但由于 Windows 系列友好的用户界面和为程序员所熟悉的 API，并捆绑 IE、Office 等应用程序，正逐渐获得更大的市场份额。

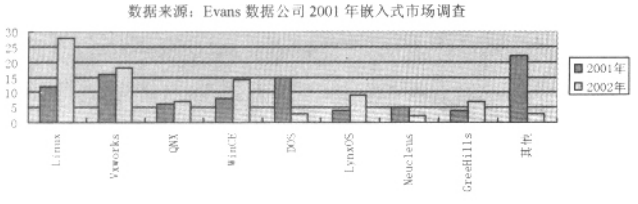

图 1　嵌入式市场份额

由图 1 可见，Linux 已经在嵌入式领域显示了强劲的发展势头。Linux 是一个成熟而稳定的网络操作系统，将它用于嵌入式设备具有众多的优点。首先，Linux 的源代码是开放的，任何人都可以获取并修改，用之开发自己的产品。其次，Linux 是可以定制的，其内核最小只有约 134KB。一个带有中文系统和图形用户界面的核心程序也可以做到不足 1MB，并且同样稳定。另外，它和多数 Unix 系统兼容，应用程序的开发和移植相当容易。同时，由于具有良好的可移植性，人们已成功使 Linux 运行于数百种硬件平台之上。然而，Linux 并非专门为实时性应用而设计，因此如果想在对实时性要求较高的嵌入式系统中运行 Linux，就必须为之添加实时软件模块。这些模块运行的内核空间正是操作系统实现进程调度、中断处理和程序执行的部分，因此错误的代码可能会破坏操作系统，进而影响整个系统的可靠性和稳定性。

Linux 的众多优点还是使它在嵌入式领域获得了广泛的应用，并出现了数量可观的嵌入式 Linux 系统。其中有代表性的包括：μClinux (www.uclinux.org)、ETlinux (www.ETlinux.org)，这两种发行版的技术特点将在下文详细讨论，ThinLinux (www.thinlinux.org) 面向专用的照相机服务器、X-10 控制器、MP3 播放器和其它类似的嵌入式应用，LRP(Linux Router Project)面向路由应用。各大主要的 Linux 开发商也推出了大量的商业化嵌入式 Linux 版本。

1　ETlinux

ETlinux 由 Prosa 公司开发和维护，通常用于在小型工业计算机，尤其是 PC/104 模块。其最新发行版本为 1.2.1，下文的讨论主要基于 ETlinux1.1，它使用的 linux 核心版本为 2.0.38。开发 ETlinux 最初的目的只是为了证明 linux 有能力在一些计算能力较弱的机器上流畅的运行，通常，它仅仅需要一个 386SX 的处理器，两兆的内存和两兆的磁盘空间，而且这样的配置就已经包括了一个 Tcl 解释器、一个 SMTP 邮件服务器和一个 Web 服务器了。

ETlinux 主要通过以下三种方法来缩小系统内核的体积：打开/关闭核心的配置选项、缩小一些核心数据结构的尺寸、移除一些不重要的核心子系统。比如：include/linux/fs.h 中的常数 NR_SUPER 由 64 减小

到了 4，此常数控制着系统可以同时安装的文件系统的数目；include/linux/tasks.h 中的常数 NR_TASKS 由 512 减小到了 32，此常数控制着系统可以同时启动的进程数；另外，系统中的字符设备和块设备数以及核心的消息缓冲区大小也被大幅缩减了。ETlinux 采取的最重要的举措还是取消对视频控制台 (video console)的支持，它节省了几乎整整 90K 的存储空间。

ETlinux 中广泛采用 Tcl 语言 [1]（一种由 John Ousterhout 发明的脚本语言，它简单易学，并可以通过其 C 语言 API 扩展功能）书写程序，以缩小应用程序的静态体积。一些 linux 的基本应用程序如 init、mount、ifconfig 等都是为台式机和服务器设计的，它们包含了大量嵌入式系统所不需要的功能和选项。另外，标准的 linux 采用 ELF 可执行文件格式，每个二进制应用程序都具有一个结构复杂而体积庞大的 ELF 文件头（此文件头被操作系统用来为应用程序的运行准备必要的初始环境），浪费了大量的存储空间。为解决以上两个问题，ETlinux 采用了基于 Tcl 的应用程序的开发方案。大量的应用程序甚至包括 init、mount、route 等这样基本的 linux 程序都是可执行的 Tcl 脚本，这些用 Tcl 重写过的脚本程序一方面精简了原来版本的功能，一方面避免了 ELF 文件头的巨大开销 (Tcl 脚本为 ASCII 格式)。许多的系统调用如：dup、fork、exec、wait、kill、pipe、nice、reboot、sync、hmod、umask、mknod 也都被集成进 Tcl 的解释环境中，这样大量的原来只能用 C 语言书写的应用程序就可以用 Tcl 来完成了。

基于 Tcl 的 ETlinux 系统，其运行时的性能和存储空间占用量也获得了一定程度的优化。传统的运行程序的方法依赖于一对系统调用：fork 和 exec，其中，fork 做当前进程的拷贝，exec 替换可执行文件映像。而 ETlinux 中则不同，其程序运行的过程如下：Tcl 解释器调用 fork 做自身的拷贝，生成一个子进程，父子进程同为 Tcl 解释器，因而它们可以共享代码甚至一部分数据，这样就大大减少了存储空间的占用量；然后，子 Tcl 解释器读入应用程序脚本，依次执行其中的指令。在 ETlinux 中运行程序则仅仅需要 fork 这一个步骤，一定程度上节省了处理器时间。需要指出的是，用解释型的 Tcl 还可以进行快速的原型开发和定制应用程序。事实上，在 ETlinux 中为 init 或 web 服务器增加功能是一件相当容易的事情。

2 μClinux

μClinux 是一个符合 GNU/GPL 公约的开放源代码项目，现由 Lineo 公司维护。μ 为 Micro，微小的意思，C 为 Control，控制的意思，即它专为微控制领域而设计。μClinux 采用了许多的做法以缩小整个系统(包括核心和应用程序) 的体积，比如它用 romfs 取代了 ext2 文件系统，后者是标准 linux 默认的文件系统，romfs 简单的多，内核仅需要很少的支持代码，且它的超级块占用更少的存储空间。romfs 的缺点是不支持动态擦写和保存，μClinux 采用虚拟 ram 盘的方法来弥补这个缺点[2]（ram 盘采用 ext2 文件系统）。μClinux 还重写了一些应用程序库，如将功能强大而体积臃肿的 glibc 和 libm 库精简成轻灵精悍的 μClibc 和 μClibm。还有，μClinux 采用 flat 可执行文件格式，这种格式对 ELF 文件头和一些段信息作了简化，缩小了可执行文件的体积。由于 gcc 通常只生成 coff 或 elf 格式的可执行文件，所以，开发者往往需要用 coff2flat 或 elf2flat 等工具进行格式转换，以获得 flat 文件。

μClinux 采取的最重要的举措是取消对虚拟存储器的支持。标准 Linux 的核心支持 MMU(内存管理单元)，其优点是：可运行比物理内存大的程序、可运行部分加载的程序、缩短启动时间、减轻程序员分配和管理内存的负担、实现代码共享、提供内存保护等。然而，虚拟存储器的管理需要一些数据结构，留给程序的物理内存必然会减少。同时，地址映射延长了指令的执行时间，降低了系统的性能。更重要的是，MMU 增加了硬件成本和运行功耗。因此，许多嵌入式 CPU 中取消了 MMU 部件，而 μClinux 正是专门为这类 CPU 设计的。取消 MMU 部件会引起一系列的问题。例如，μClinux 在装入程序时，必须一次性的分配足够的、连续的物理内存，而标准 Linux 在装入程序时，仅需为之分配足够的虚拟存储空间。又如，由于没有 MMU，程序的实际加载地址与.ld 文件中的预期加载地址并不相同，这就需要一个额外的地址重定位的过程。此外，由于不能使用磁盘交换空间，系统将更易于面临存储空间耗尽的窘境。还有，由于采用实存储器管理策略，所有的用户进程和内核共享一个全局的地址空间，它们所占据的内存区域之间缺乏硬件级别的保护，程序员必须确保他的代码不侵犯其它程序的运行空间。

μCLinux 采取一种和标准 Linux 完全不同的动态内存分配方案。标准 Linux 中，每个进程具有私有的虚拟内存空间，其中依次是代码段、静态数据段、堆段和栈段。堆顶和栈底之间存在着一个 256MB 的虚拟内存空隙。正是由于这个空隙，堆和栈的大小才可以动态的调整。系统调用 brk 可用于改变堆的大小，标准 C 函数 malloc 正是用 brk 的这个特性来分配内存。标准 Linux 中，栈的大小由系统根据实际的需要自动调整，由于栈和堆之间的内存空隙，系统能确保在扩缩栈的同时不侵犯堆空间[3]。μClinux 中情况则不同，为了避免存储空间的浪费，系统为进程分配的内存区域是连续的，代码段、数据段和栈段之间没有任何空隙。同样，出于节省内存的目的，进程的私有堆被取消，所有进程共享一个由操作系统管理的堆空间[4]。这样，基于 brk 的 malloc 实现已经不再适用。在 μCLinux 中，malloc 通过系统调用 mmap 从核心的空闲内存池中分配内

存 (munmap 用于将内存返还核心内存池)。

另外，由于 μCLinux 中栈和静态数据段之间没有间隙，因此栈段不能动态增长和缩减，程序员必须事先估算出栈的大小，并以编译选项的形式告诉编译器，才能确保程序运行时，栈段既不会侵犯数据段，又不至于浪费过多的内存。可见，μClinux 的程序员在开发应用时必须参与系统存储空间的分配和管理，其结果是，程序员对内存的控制权增大 (程序员可以访问任意的地址)，但管理负担增加，系统的安全性下降。从功能上来说，μClinux 的内存管理退回了早期的 UNIX、DOS 时代。但是，这种倒退是嵌入式系统硬件支持不足造成的，这也是大量的嵌入式系统中还在使用 DOS 的原因。

取消 MMU 对 μClinux 中的进程管理也造成了一定的影响。在标准 Linux 中，父进程用系统调用 fork 来创建子进程，子进程是父进程的精确拷贝，父子进程拥有各自的地址空间，从 fork 调用后同时开始运行。标准 Linux 中的 fork 操作是非常高效的，这主要是由于采用了写时拷贝(copy on write)技术，这种技术依赖于 MMU 的支持，因而无法用于 μClinux。在 μClinux 中，所有进程共享一个全局的地址空间，fork 产生的子进程中的指针将仍然指向父进程中的数据。基于以上两点原因，μClinux 取消了 fork 操作，而用 vfork 取而代之[5]。在 vfork 调用之后，父进程将自己的内存空间租借给子进程，即，父子进程占据同一块内存区域，父进程在 vfork 调用之后开始睡眠，直到子进程调用 exec 或 exit，内核才将内存空间返还给父进程，并将之唤醒。vfork 不涉及整个地址空间的数据复制，且不会产生子进程中的指针指向父进程中的数据这样的问题，因此它的语义更加适合于 μClinux 环境。需指出的是，由于 vfork 调用之后父子进程共享同一块内存区域，子进程必须尽快的调用 exec(exit)，并且在 vfork 和 exec(exit)之间不修改父进程的数据，也不破坏父进程的堆栈环境，否则极易引起系统崩溃。另外，大量现代的应用程序(如许多网络服务程序)依靠 fork 的语义来提高系统的吞吐率和缩短响应时间，它们必须经过相当的修改才能在 μClinux 上运行。

3 嵌入式 Linux 的图形用户界面

鉴于 GUI 在操作系统中的独特地位和重要作用，我们单独讨论嵌入式 linux 的图形用户界面。目前，面向嵌入式 Linux 的 GUI 系统有 Microwindows/NanoX、OpenGUI、Qt/Embedded、MiniGUI 等。其中，Qt/Embedded 由著名的 TrollTech 公司开发，因为 Qt 是 KDE 等项目使用的 GUI 支持库，所以许多基于 Qt 的应用程序可以方便地移植到 Qt/Embedded 上。因此，自从 Qt/Embedded 以 GPL 的形式发布以来，大量的嵌入式 Linux 开发商转到了 Qt/Embedded 系统上。然而，Qt/Embedded 本身也存在着一些问题。首先，Qt/Embedded 是一个 C++函数库，尽管它最小可以裁剪到 630K，但此时的 Qt/Embedded 已经基本上失去了使用价值。低运行效率、大资源消耗对运行 Qt/Embedded 的硬件提出了相当高的要求。其次，Qt/Embedded 目前主要针对手持式信息终端，对硬件加速支持匮乏，很难应用到对图形速度、功能和效率要求较高的嵌入式系统当中，比如机顶盒、游戏终端等。第三，Qt/Embedded 提供的控件沿用了 PC 上的风格，并不太适合许多手持设备的操作要求。最后，Qt/Embedded 的结构过于复杂，很难进行底层的扩充、定制和移植，尤其是那个用来实现 signal/slot 机制的著名的 moc 文件。由于上述的这些原因，目前所见到的 Qt/Embedded 的运行环境，几乎全部是基于 StrongARM 的 iPAQ。

MiniGUI 是由魏永明主持，由许多自由软件开发人员支持的一个自由软件项目 (遵循 LGPL 条款发布)，其目标是为基于 Linux 的实时嵌入式系统提供一个轻量级的图形用户界面支持系统。该项目自 1998 年底开始，现在已经非常成熟和稳定，其最新稳定版本为 1.2.x。MiniGUI 的技术特色包括①图形抽象层。图形抽象层对上层 API 基本没有影响，但方便了 MiniGUI 应用程序的移植和调试。图形抽象层目前包含三个引擎：SVGALib、LibGGI 以及直接基于 Linux FrameBuffer 的本地引擎。与图形抽象层相关的还有输入事件抽象层。MiniGUI 现在已经被证明能够在基于 ARM、MIPS、StrongARM 以及 PowerPC 等的嵌入式系统上流畅的运行。②多字体和多字符集支持。此特性通过设备上下文 (DC)的逻辑字体 (LOGFONT)实现，不管是字体类型还是字符集，都可以非常方便的扩充。应用程序在启动时，可切换系统字符集，如 GB、BIG5、EUCKR、UJIS 等。在调用 DrawText 等函数时，可通过指定字体而获得其他字符集支持。对于一个窗口来说，同时显示不同语种的文字是可能的。MiniGUI 的这种字符集支持不同于传统的通过 UNICODE 实现的多字符集支持，更加适合于嵌入式系统。③两个不同架构的版本。最初的 MiniGUI 运行于 PThread 库之上，这个版本适合于功能单一的嵌入式系统，但不够健壮。0.9.98 版本引入了 MiniGUI-Lite，这个版本在提高系统健壮性的同时，通过一系列创新途径，避免了传统 C/S 结构的弱点，为功能复杂的嵌入式系统提供了一个高效、稳定的 GUI 环境。

目前 MiniGUI 还在积极的开发和完善当中，其发展目标包括：提供运行在 MiniGUI 上的 JAVA 虚拟机 AWT 组件的实现、提供 MiniGUI 上的 OpenGL 实现、提供类 QT 控件集的 C++封装、提供窗口/控件风格主题支持、在 MiniGUI-Lite 当中增加对矢量字体的支持等。

4 结束语

目前，绝大多数的嵌入式系统硬件平台仍然掌握

(下转第 32 页)

1)因为源数据表 pxd.dbf 文件已存在,故不需要步骤 1。

2) 配置分别对应源数据表 pxd.dbf 和目标数据库的 ODBC。

3) 利用 Delphi 5.0 中的数据导换工具 DATA PUMP 把源数据表 pxd.dbf 导入到 Microsoft SQL Server 2000 中一个名为 pxd.mdf (与目标数据表 pxbt.mdf 处在同一个数据库管理系统 DBMS 中)表中,称它为中间表。中间表 pxd.mdf 和源数据表的数据,字段,数据类型都相同,只是中间表处在 Microsoft SQL Server 2000 DBMS 中。

4)设计中间表 pxd.mdf,添加字段 '客户代号 ' varchar(20)及字段 '平均折扣 2 'float 型。在前端编制程序生成字段 '客户代号 '和 '平均折扣 2 '信息。部分实现代码如下:

```
procedure Tproducepxdform.BitBtn1Click(Sender:TObject);
begin
with cds do  //从中间表 pxd 中查询出批销单信息。
  begin
   close;
   commandtext:='select * from pxd';
   open;
  end;
end;
```

下面一段程序是循环往中间表 pxd 中字段 '客户代号 '和 '平均折扣 2 '插入信息

```
procedure Tproducepxdform.BitBtn2Click(Sender:TObject);
var
 tsdbh,khid,zk:string;  d:integer;  zk2:real;
begin
with cds do
 begin
   first;
   while not eof  do
   begin
tsdbh:=fieldbyname('提书单编号').asstring;    //从提书单编号中提取出客户代号, //并赋给 khid
d:=pos('.', tsdbh);
khid:=copy(tsdbh,1,d-1);
   zk:=fieldbyname('平均折扣').asstring;      //把平均折扣从百分比形式转换为小数形
d:=pos('%',zk)                //式,并赋给 zk2
   zk2:=strtoint(copy(zk,1,d-1))/100;
  with cds2 do  //往中间表 pxdt 对应记录中字段 '客户代
     begin                //号 '和 '平均折扣 2 '插入信息
       close;
       commandtext:='update pxd set 平均折扣 2='
                +floattostr(zk2)+',客户代号='''
+khid+''' where 提书单编号='''+tsdbh+'''';
       execute;
     end;   //with cds2 do
    next;
  end;   //while not eof  do
 end;  //with cds do
end;
```

5)经过步骤 4 处理,在中间表 pxd 中已生成客户代号信息,并把平均折扣信息由字符型转为浮点型存放在字段 '平均折扣 2 '中。至此,中间表的数据字段,数据类型与目标表中的基本一致。这时可以利用 Microsoft sql server 2000 自带的数据迁移工具 DTS 把数据从中间表 pxd.mdf 导入到目标表 pxbt.mdf 中 (在这过程中是把中间表的 '客户代号 '信息、'平均折扣 2 '信息 (不是 '平均折扣 '信息)分别导入目标数据表中的字段 '客户代号 '和 '平均折扣 '中)

3 结束语

在管理信息系统开发中数据迁移可以有多种方式,这里只是讨论了其中一种。但是不管采用何种方式,都要尽可能的减少丢失数据,以确保迁移质量。

参考文献

1 [美]MIKE Gunderloy,Mike Chipman 著.SQL SERVER 7 轻松进阶.电子工业出版社,1999

2 徐新华编著.Delphi 5 高级编程——Database 与 MIDAS 编程.人民邮电出版社,2000

3 新智工作室编著.Delphi 5 数据库编程. 电子工业出版社,2000

4 王能斌.数据库系统.电子工业出版社,1998

收稿日期:2002.8.23]

(止接第 30 页)

在国外大公司的手中,国产的嵌入式操作系统在技术含量、兼容性、市场运作模式等方面还有相当的差距。但由于 Linux 自由操作系统的出现,特别是将嵌入式系统和 Linux 有机结合起来,给我们提供了跟踪国外嵌入式操作系统最新应用技术的难得机遇。可喜的是,目前国内已经出现了一些开发商,如合肥华恒、中科红旗、广州博利思等,积极的从事嵌入式 linux 的研究和开发,并出现了由中国人领导的嵌入式 GUI 系统——MiniGUI。可以肯定的是,嵌入式 Linux 将在工业控制、机顶盒、掌上电脑或 PDA、手机和寻呼机上网、车载盒等方面获得更为广泛地应用。

参考文献

1 John Ousterhout.Tcl and the Tk Toolkit.Addison-Wesley,1994

2 陈云鹏. 实现 8-16 位嵌入式系统的网络互联. 学士学位论文].中国科技大学,2000

3 Scott Maxwell.Linux 内核源代码分析. 机械工业出版社,2000

4 μClinux White Paper Overview.http://www.arcturusnetworks.com/Docs/UCLINUXWP.pdf

5 μClinux Technical Brief.http://www.arcturusnetworks.com/Docs/UCLINUXTB.pdf

收稿日期:2002.8.5]