

基于嵌入式 Linux 系统的图形用户界面综述

孙少华,徐立中

(河海大学 计算机及信息工程学院,江苏 南京 210098)

摘 要: 对目前市场上较新的 GUIXfree86 4.X、MiniGUI、MicoroWindows/NanoX 等作了详细介绍,对各自的优缺点进行了分析。

关键词: 嵌入式 Linux 图形用户界面(GUI) QT/Embedded MiniGUI

嵌入式系统是以应用为中心,以计算机技术为基础且软硬件均可裁剪的专用计算机系统。适用于对功能、可靠性、成本、体积、功耗有严格要求的应用领域系统^{[1][2]}。嵌入式系统的发展已有 20 多年的历史。虽然目前已出现了很多基于 Linux 的嵌入式系统,但由于嵌入式系统本身硬件的条件限制,常用 PC 机的 GUI 系统不适合在其上运行。嵌入式系统对 GUI 有着特殊的要求:由于嵌入式系统对实时性要求非常高,对 GUI 的要求也高,所以这种系统一般不宜建立在庞大臃肿、非常消耗系统资源的操作系统和 GUI 之上,如 Windows 或 X Window 均过于庞大和臃肿。因此,这些系统对轻型 GUI 的需求更加突出。另外嵌入式系统往往是一种定制设备,它们对 GUI 的需求也必须是可定制的。所以,嵌入式系统对 GUI 的基本要求包括轻型、占用资源少、高性能、高可靠性和可配置。从用户的观点来看,图形用户界面(GUI)是系统的一个最至关重要的方面:用户通过 GUI 与系统进行交互。所以 GUI 应该易于使用并且非常可靠。但它还需要是有内存意识的,以便在内存受限的、微型嵌入式设备上可以无缝执行。所以,它应该是轻量级的,并且能够快速装入。

现在已有许多公司和科研院校致力于 GUI 的开发研究,并已开发出许多 GUI 产品,如 Century Software 的 Microwindows(Nano-X)和 Trolltech 的 QT/Embedded,它们与 Xfree86 在嵌入式 Linux 的竞技舞台上展开了激烈竞争。

1 国内外研究与应用现状

1.1 XFree86 4.X(带帧缓冲区支持的 X11R6.4)

XFree86 是一个由 XFree86 Project 公司开发的可以重复分发、开放源码的 X Window 系统。X Window 系统(X11)为应用程序以图形方式进行显示提供了资源,并且它是 Unix 和类 Unix 的机器上最常用的窗口系统。它很小但很有效,可运行在众多的硬件上,对网络透明并且有良好的文档说明。X11 为窗口管理、事件处理、同步和客户机间通信提供强大的功能,并且大多数开发人员已经熟悉了

它的 API。X11 具有对内核帧缓冲区的内置支持,并占用非常少的资源,这非常有助于内存相对较少的设备。X 服务器支持 VGA 和非 VGA 图形卡,它对颜色深度 1、2、4、8、16 和 32 提供支持,并对渲染提供内置支持。最新的发行版是 XFree86 4.1.0。

X11 的优点包括:帧缓冲区体系结构的使用提高了系统性能;占用的资源相对很小——大小在 600K~700KB 的范围内,这使它很容易在小型设备上运行;非常好的支持:在线有许多文档可用,还有许多专用于 XFree86 开发的邮递列表;X API 非常适合扩展。

X11 的缺点包括:比最近出现的嵌入式 GUI 工具性能差;与 Nano-X 或 QT/Embedded 相比,XFree86 需要更多的内存。

1.2 MicoroWindows/Nano-X

MicroWindows^{[4][5]}是一个开放源码的项目,由美国 Century Software 公司主持开发,专为带小型显示单元的微型设备而设计。它有许多针对现代图形视窗环境的功能部件。与 X 一样,有多种平台支持 Microwindows。Microwindows 体系结构是基于客户机/服务器的并且具有分层设计。最底层的屏幕和输入设备通过驱动程序(关于键盘或鼠标)来与实际硬件交互。在中间层,可移植的图形引擎提供线条的绘制、区域的填充、多边形、裁剪以及颜色模型的支持。在最上层,MicroWindows 支持二种 API:一种是 Win32/WinCE API 实现,被称为 MicroWindows;另一种 API 与 GDK 非常相似,被称为 Nano-X。Nano-X 用在 Linux 上。它很像 X 的 API,用于占用资源少的应用程序。

MicroWindows 支持 1、2、4 和 8 bpp(每像素的位数)的 palletized 显示以及 8、16、24 和 32 bpp 的真彩色显示。MicroWindows 还支持使它速度更快的帧缓冲区。Nano-X 服务器占用的资源约为 100K~150KB。原始 Nano-X 应用程序的平均大小为 30K~60KB。由于 Nano-X 是为有内存限制的低端设备设计的,所以它不像 X 那样支持很多函数,

《微型机与应用》2005 年第 4 期

因此它实际上不能作为微型 X(Xfree86 4.1)的替代品。

可以在 MicroWindows 上运行 FLNX,它是针对 Nano-X 而不是 X 进行修改的快速轻巧工具箱 FLTK(Fast Light Toolkit)应用程序开发环境的一个版本。

Nano-X 的优点包括:与 Xlib 实现不同,Nano-X 仍在每个客户机上同步运行。这意味着一旦发送了客户机请求包,服务器在为另一个客户机提供服务之前一直等待,直到整个包都到达为止。这使服务器代码非常简单,而运行的速度仍非常快,占用很小的资源。

Nano-X 的缺点包括:联网功能部件至今没有经过适当的调整(特别是网络透明性);还没有太多现成的应用程序可用;与 X 相比,Nano-X 虽然近来正在加速开发,但仍没有足够多的文档说明而且没有很好的支持,但这种情形会有所改变。

Microwindows 上的 FLTK API(简称 FLTK)是一个简单而又灵活的 GUI 工具箱,它在 Linux 世界中赢得越来越多的关注,特别适用于占用资源很少的环境。它提供了从 GUI 工具箱中获得的大多数窗口构件,如按钮、对话框、文本框以及出色的“赋值器”选择(用于输入数值的窗口构件),还包括滑动器、滚动条、刻度盘和其他一些构件。

针对 MicroWindows GUI 引擎的 FLTK 的 Linux 版本被称为 FLNX。FLNX 由两个组件构成:FL_Widget 和 FLUID。FL_Widget 由所有基本窗口构件 API 组成。快速轻巧的用户界面设计器 FLUID(Fast Light User Interface Designer,FLUID)是用来产生 FLTK 源代码的图形编辑器。总之,FLNX 是能用来为嵌入式环境创建应用程序的一个出色的 UI 构建器。

FL_Widget 占用的资源大约是 40K~48KB,而 FLUID(包括了每个窗口构件)大约占用 380KB。其非常小的资源占用率使 FL_Widget 和 FLUID 在嵌入式开发世界中非常受欢迎。

FLNX 优点包括:习惯于在 Windows 之类已建立好的环境中开发基于 GUI 的应用程序的任何人都会非常容易地适应 FLTK 环境;它的文档包括一本十分完整且编写良好的手册;它使用 LGPL 进行分发,所以开发人员可以灵活地发放其应用程序的许可证;FLTK 是一个 C++库(Perl 和 Python 绑定也可用)。面向对象模型的选择是一个好的选择,因为大多数现代 GUI 环境都是面向对象的。这也使将编写的应用程序移植到类似的 API 中变得更容易;Century Software 的环境提供了几个有用的工具,诸如 ScreenToP 和 ViewML 浏览器。

FLNX 的缺点是:普通的 FLTK 可以与 X 和 Windows API 一同工作,而 FLNX 不能;它与 X 的不兼容性阻碍了它在许多项目中的使用。

1.3 OpenGUI

OpenGUI^[3]在 Linux 系统上存在已经很长时间了。最初《微型机与应用》2005 年第 4 期

的名字叫 FastGL,只支持 256 色的线性显存模式,但目前也支持其他显示模式,并且支持多种操作系统平台,如 MS-DOS、QNX 和 Linux 等,不过目前只支持 x86 硬件平台。OpenGUI 也分为三层。最低层是由汇编语言编写的快速图形引擎;中间层提供了图形绘制 API,包括线条、矩形、圆弧等,并且兼容于 Borland 的 BGI API。第三层用 C++编写,提供了完整的 GUI 集。

OpenGUI 提供了一个二维绘图原语,并提供对消息驱动的 APL 及 BMP 文件格式的支持。OpenGUI 功能强大,使用方便。OpenGUI 支持鼠标和键盘的事件,在 Linux 上基于 Frame buffer 或者 SVGALib 实现绘图。由于其基于汇编实现的内核并利用 MMX 指令进行了优化,所以 OpenGUI 运行速度非常快。但也由于它的内核是用汇编实现的,所以可移植性受到了影响。在驱动程序一级,可移植性和性能是矛盾的,开发人员必须取其折衷。

另外,OpenGUI 采用 LGPL 条款发布。OpenGUI 比较适合于基于 x86 平台的实时系统,但目前的发展基本停滞。

1.4 Qt/Embedded

Qt/Embedded^[4]是著名的 Qt 库开发商 TrollTech 发布的面向嵌入式系统的 Qt 版本。因为 Qt 是 KDE 等项目使用的 GUI 支持库,所以有许多基于 Qt 的 X Window 程序可以非常方便地移植到 Qt/Embedded 版本上。因此,自从 Qt/Embedded 以 GPL 条款形式发布以来,有大量的嵌入式 Linux 开发商转到了 Qt/Embedded 系统上。如韩国的 Mizi 公司,中国台湾的某些嵌入式 Linux 应用开发商等。

但是,Qt/Embedded 还存在以下问题值得开发者注意。

(1)目前,该系统采用两种条款发布,其中包括 GPL 条款。对函数库使用 GPL 条款,意味着其上的应用需要遵循 GPL 条款。当然,如果要开发商业程序,TrollTech 也允许采用另外一个授权条款,但这时必须向 TrollTech 交纳授权费用。

(2)Qt/Embedded 是一个 C++函数库,尽管 Qt/Embedded 声称最少可以裁剪到 630KB,但这时的 Qt/Embedded 库已经基本上失去了使用价值。低的程序效率、大的资源消耗也对运行 Qt/Embedded 的硬件提出了更高的要求。

(3)Qt/Embedded 库目前主要针对手持式信息终端,因为缺乏对硬件加速的支持,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(如机顶盒、游戏终端等)中。

(4)Qt/Embedded 提供的控件集风格沿用了 PC 风格,并不太适合许多手持设备的操作要求。

(5)Qt/Embedded 的结构过于复杂,很难进行底层的扩充、定制和移植,尤其是用来实现 signal/slot 机制的著名的 moc 文件。

由于上述这些原因,目前所见到的 Qt/Embedded 的运行环境几乎全是基于 StrongARM 的 iPAQ。

1.5 MiniGUI

MiniGUI 是由清华大学魏永明主持开发并由许多自由软件开发人员支持的一个自由软件项目,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 年底开始到目前为止,已经非常成熟和稳定。目前,已经正式发布了稳定版本 1.0.9,并且开始了新版本系列的开发,即 MiniGUI Version 1.1.x,该系列的正式版也即将发布。

MminiGUI 的主要特点有:(1) 遵循 GPL 条款的纯自由软件。(2) 提供了完备的多窗口机制,包括:多个单独线程中运行的多窗口;单个线程中主窗口的附属;对话框和预定义的控件类(按钮,单行和多行编辑框,列表框,进度条及工具栏等);消息传递机制。(3) 多字符集和多字体支持,目前支持 ISO88591,GB2312 和 ig5 等字符集,并且支持各种光栅字体和 TrueType、Type 1 等矢量字体。(4) 支持全拼和五笔等汉字输入法。(5) 支持 BMP, GIF, JPEG 及 PCX 等常见图像文件。(6) 支持 Windows 的资源文件,如位图、图标、光标、插入符、定时器及加速键等。(7) 库文件小,包括全部功能的库文件大小为 300KB 左右。(8) 可配置。可根据项目需求进行定制配置和编译。(9) 高稳定性和高性能。MiniGUI 已经在 Linux 发行版安装程序, CNC 系统及蓝点嵌入式系统等关键应用程序中得到了实际应用。(10) 可移植性好。目前, MiniGUI 可以在 X Window 和 Linux 控制台上运行。中科院 EEOS 开发组已经成功地 MiniGUI 移植到了他们的 POSIX 兼容系统上。蓝点软件(北京)研发中心也已经成功地将 MiniGUI 移植到了两款基于 StrongARM 的嵌入式系统上。

另外,在 MiniGUI 几年的发展过程中,有许多值得一提的技术创新点。正是由于这些技术上的创新,才使得 MiniGUI 更加适合实时嵌入式系统。而且 MiniGUI 的灵活性非常好,可以应用在包括手持设备、机顶盒、游戏终端等在内的各种高端或低端嵌入式系统中。这些技术创新包括:(1) 图形抽象层。图形抽象层对顶层 API 基本没有影响,但大大方便了 MiniGUI 应用程序的移植、调试等工作。目前包含三个图形引擎, SVGALib、LibGGI 以及直接基于 Linux FrameBuffer 的 Native Engine, 利用 LibGGI 时,可在 X Window 上运行 MiniGUI 应用程序,并可非常方便地进行调试。与图形抽象层相关的还有输入事件的抽象层。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运行。

(2) 两个不同架构的版本。最初的 MiniGUI 运行在 PT-head 库之上,这个版本适合于功能单一的嵌入式系统,但存在系统健壮性不够的缺点。在 0.9.98 版本中引入了 MiniGUI-Lite 版本,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 系统。

2 自主开发新的 GUI 系统

从技术角度讲,上述这些平台或多或少存在一些缺陷。有些平台原来是面向 PC 环境的,对于系统资源的要求较高,尽管经剪裁后可能可以适应对于静态空间的要求,但对动态空间的要求依然难以降低,而且经剪裁以后,其功能受到极大削弱;有些平台尽管较为成熟,但真正在商品化产品中使用时会面临很高的授权费用,无形中增加了产品的成本,使得产品在市场竞争中面临困境(如 Qt/Embedded);有些平台因其实现策略问题,使得二次开发受到限制(如 MiniGUI 只在一个进程中运行,不管是 desktop 还是其他所谓的应用程序,其实都是其中的一个线程,这并不是一个构架良好的 GUI 系统所应采取的策略);有些则因过于追求速度,实现中较多的代码采用汇编来实现,使得移植非常困难,而嵌入式系统是千差万别的,移植性差是其一大弊端(如 openGUI)。

从市场角度讲,嵌入式产品是高度个性化的,其外观、与用户交互的界面也是千差万别的。对于特定的产品定义,即使是以 LGPL 方式发布的产品,二次开发者也无法满足这一要求。而在源代码级拥有主动,就可以适应产品定义的任何变化。另外,由于 Linux 是开放源码,从而在嵌入式领域占有越来越重要的地位,拥有 GUI 的知识产权,并在此基础上开放核心应用,便足以打造企业的核心竞争力,使企业在嵌入式领域中占有一席之地。

另外,认真研究现有的 GUI 系统,可以从中发现其存在的一些不足。这样在开发新的 GUI 的时候,可尽量避免出现类似的问题,从而使得新的系统在性能、资源占用、二次开发支持等方面得到提高。

3 结束语

目前,嵌入式 Linux 系统的研发热潮正在蓬勃兴起,并且占据了很大的市场份额,除了一些传统的 Linux 公司(如 RedHat、MontaVista 等)正在从事嵌入式 Linux 的开发和应用之外,IBM、Intel、Motorola 等著名企业也开始进行嵌入式 Linux 的研究。虽然前景一片灿烂,但就目前而言,在嵌入式 Linux 图形领域,还有许多问题有待于开发人员深入研究和解决。

参考文献

- 1 邹思铁.嵌入式 Linux 设计与应用.北京:清华大学出版社,2002
- 2 魏永明.实时嵌入式 Linux 系统上 GUI 的发展与展望.微型电脑世界,2000;(49)
- 3 魏永明.MiniGUI 技术白皮书[J/OL].<http://www.minigui.org/cdoc.html>.2000.12
- 4 Abrash M.图形程序开发人员指南.北京:机械工业出版社,1998
- 5 Haerr G.Microwindows Architecture.www.microwindows.com,1999.12

(收稿日期:2004-10-15)

《微型机与应用》2005 年第 4 期