

基于 Qt 的电纸书图形用户界面设计^①

方世烟, 林 东

(福州大学 物理与信息工程学院, 福州 350108)

摘 要: 电子书在日常生活中应用广泛, 为了便于用户直观、舒适地阅读电子书, 电纸书应运而生。在 Linux 操作系统上, 利用 Qt 图形用户界面开发工具, 实现电纸书图形用户界面的设计, 并在电子纸上实现文本文档的显示。程序实现了主窗口和对话框的创建, 以及主窗口和对话框之间的切换与交互。同时, 在加载文本文档的过程中, 实现对不同编码方式的文本文档的解码及分页。

关键词: 电纸书; Qt; 图形用户界面设计; 主窗口; 对话框; 文本文档

Design of Graphical User Interface for E-Paper Book Based on Qt

FANG Shi-Yan, LIN Dong

(School of Electronic Science and Technology, Fuzhou University, Fuzhou 350108, China)

Abstract: In order that users can read E-books more intuitively and comfortably, the E-paper Book emerged for E-books' wide applications in daily life. In this paper, design of graphical user interface of E-paper Book is implemented based on Linux operating system, with Qt graphical user interface development tools. Then, text documents can be showed on the electronic paper. The program implements the creation of the main window and some dialog boxes, and the switching and the interaction between the main window and the dialog boxes. Then, text documents decoding and paging of the different encoding methods are implemented in the process of loading text documents.

Key words: e-paper book; Qt; design of graphical user interface; main window; dialog box; text documents

从广义上说, 电纸书是电子书的一部分, 它通常是指一种采用电子纸的显示屏幕的新式数字阅读器, 可以阅读网上绝大部分格式的电子书, 比如 PDF, CHM, TXT 等。与传统的手机, MID, UMPC 等设备相比, 采用电子纸技术的电子书阅读器的优点是辐射小、耗电低、不伤眼睛、携带方便, 而且它的显示效果逼真, 具有类似纸质书本的效果^[1]。由于电子纸显示屏屏幕大小及灰阶的限制和硬件上按键个数的限制, 设计适合在电子纸上显示的图形用户界面至关重要。电纸书图形用户界面在满足用户基本需求的情况下, 界面应尽量简洁、美观。

1 电纸书图形用户界面设计

图形用户界面是嵌入式系统软件开发的重要内

容, 基于 Linux 内核开发实用、高效、美观的 GUI 是目前研究的一个重要课题, 而作为应用于嵌入式 Linux 的主流 GUI 系统之一的 Qt/Embedded 也得到青睐。实验中设计的电纸书图形用户界面是在 Linux 操作系统下, 利用 Qt 图形界面开发工具完成电纸书图形界面设计, 并采用电子纸作为显示屏, 实现文本文档的显示。实现过程的关键技术在于: 实现文档文件的读取及分页; 设计适合电纸书的图形用户界面。

1.1 嵌入式 GUI

图形用户界面(GUI)是当前嵌入式操作系统的一个重要组成部分, 它将极大的促进嵌入式技术的发展和运用。图形用户界面系统通常是指由相应硬件平台和操作系统支持的显示模型、窗口模型和用户模型、以及由这三个模型的应用程序接口 API(Application Program

① 收稿时间:2011-07-02;收到修改稿时间:2011-07-27

Interface)。图形用户界面模型决定了图形在屏幕上的基本显示方式，即用位映象图形显示技术显示各种图形对象的方式，不同的图形用户界面系统采用的显示模型各不相同。窗口模型确定了在屏幕上显示的多个应用程序的形象，体现在程序设计上可表示为构造可移动、可伸缩的窗口、菜单、滚动条、对话框等，它通常包括两部分：一是编程工具，如函数集；二是对如何移动、输出和读取屏幕显示信息的说明。图形用户界面的用户模型也称为图形用户界面的外观与视觉，它也包括两部分：一是构造用户界面的工具，包括对高层界面构件对象元的是数据结构的定义和说明；二是定义在屏幕上组织各种图形对象以及这些对象之间的行为规范和协议。图形用户界面系统的应用程序接口 API 由显示模型、窗口模型和用户模型的应用程序接口共同组成，各模型的应用程序接口主要是该模型提供给开发者的开发工具。桌面管理系统是人机交互的图形化管理系统，它通常包括以下几个组成部分：图形化的文件管理系统、供用户使用的图标库、桌面管理机构、图标库管理机构等^[2-4]。

1.2 Qt

Qt 是 Trolltech 公司开发的跨平台的 C++图形用户界面库。QT/Embedded 是面向嵌入式 Linux 系统的 Qt 版本，它提供了完全的 GUI 功能，有丰富的控件资源，无需 X11 或者 Motif 的支持。Qt/Embedded 是模块化的，用户可根据实际需要对其进行裁剪，以适应不同的应用环境，是可配置和可伸缩的。Qt/Embedded 窗口系统通常包括一个服务器进程和若干客户进程。该系统采用 C/S 机制，服务器进程为客户进程分配显示区域，并产生鼠标事件和键盘事件；客户进程和服务器进程进行通信，向服务器进程申请显示区域并接收鼠标和键盘事件^[5]。Qt/Embedded 的实现结构如图 1 所示：

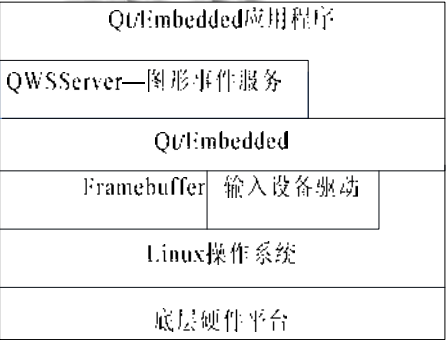


图 1 Qt/Embedded 的实现结构

1.3 电纸书图形用户界面的总体结构设计

在 Linux 操作系统下，利用 Qt Creator 图形界面开发工具，设计电纸书图形用户界面，主要内容包括主窗口的设计、对话框的生成及文档内容的显示。

其中主窗口包括：文本编辑区，用于显示文本文档内容；状态栏，用于显示总页数和当前页。Menu 对话框包含以下三个按钮：File Dialog、Font Dialog 和 Turn to Page，File Dialog 用于打开文件浏览器 File Viewer，Font Dialog 用于打开字体设置对话框 Font Set，Turn to Page 用于打开指定页阅读对话框 Turn to Page，如图 2 中的(a)所示。

文件浏览器用于实现文件系统的浏览功能，浏览所有的文件，其中顶部的编辑框显示当前所在的目录路径，列表中显示该目录下的所有文件，如图 2 中的(b)所示。

字体设置对话框用于设置字体类型和字体大小，其中字体类型包括宋体、楷体、黑体；字号包括大号字体、中号字体、小号字体，如图 2 中的(c)所示。

指定页阅读对话框用于实现用户切换到文档的指定页数，其中对话框显示了文本文档的总页数以及当前页，用户在文本编辑框中输入所要切换的页数，确定后即可显示指定页，如图 2 中的(d)所示。

qvfb 上显示的电纸书图形用户界面如图 2 所示：



图 2 qvfb 上显示的电纸书图形用户界面

2 电纸书图形用户界面实现

2.1 图形用户界面的实现流程

创建应用程序用户界面的具体实现过程如下:

首先,创建应用程序主窗口,构造 QTextEdit 对象,生成中心部件并添加到主窗口。创建一个状态栏,用于显示文档的总页数和当前页。

其次,建立自定义对话框类,包括 Menu 对话框、File Viewer 对话框、Font Set 对话框以及 Turn to Page 对话框。在 Menu 对话框中添加窗口部件,即 filePushButton、fontPushButton、turnPageButton,利用 Qt 布局管理器管理窗口部件,并添加界面控制代码,包括 File Viewer 对话框、Font Set 对话框和 Turn to Page 对话框的显示代码。在 File Viewer 对话框中添加 Line Edit 和 List Widget 窗口部件,即 lineEdit 和 listWidget,并添加界面控制代码,即 QDir 显示文件系统目录,过滤方式显示文件列表。在 Font Set 对话框中添加 Line Edit 和 List Widget 窗口部件,即 familyEdit、familyList、sizeEdit、sizeList,用于显示字体类型和字号,并添加界面控制代码,即字体信息的设置。在 Turn to Page 对话框中,添加 Label 和 Line Edit 窗口部件,即 label、label_2 和 lineEdit, label、label_2 分别用于显示“总页数 当前页”、“请输入页数:”, lineEdit 用于输入页数,并添加界面控制代码,即指定页显示。

最后初始化对话框,并运行,实现对用户交互中的选择作出响应。利用事件机制,自定义事件类型,包括菜单键、确定键、退出键、上下键以及 Tab 键,根据事件类型调用相应的事件处理器 keyPressEvent(),实现对对话框的调用及部件相应功能的操作。

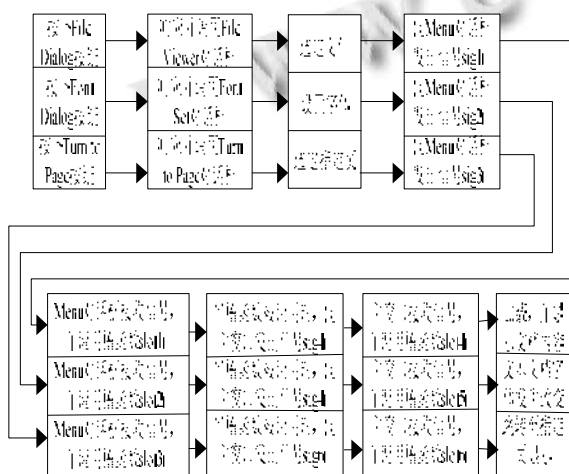


图 3 对话框的消息响应机制

当对话框被创建时,通常是把它们的信号与对用户交互作出响应的槽连接起来。Menu 对话框是一个让用户能够打开 File Viewer 对话框、Font Set 对话框以及 Turn to Page 对话框的窗口。使用 Qt 的信号和槽机制,设计槽函数来响应用户调用相应窗口,并实现相应功能的操作。其中,对话框的消息响应机制如图 3 所示。

2.2 图形用户界面的功能实现

实现的功能如下:

当用户打开主窗口界面后,按菜单键,即可打开 Menu 对话框,按 File Dialog 按钮,即可弹出 File Viewer 对话框,通过移动上下键实现在该目录下不同文件间的移动,确定键实现文件的打开操作,取消键实现撤销、返回等操作。当需要对字体进行设置时,可通过 Menu 对话框中的 Font Set 按钮,实现对 TXT 文本的字体类型和字号的修改。当用户需要阅读指定页内容时,按 Menu 对话框中的 Turn to Page 按钮,即可弹出 Turn to Page 对话框,在编辑框中输入页数,确定后即可显示指定页内容。用户可通过上下键实现 TXT 文档内容的翻页阅读,以及取消键实现退出操作。

2.3 不同编码方式的 TXT 文档的读取

TXT 文本文档一般是由两个或三个字节的标识 TXT 文件字符编码方式的的开头字节和字符数据构成,它的每一个字节存放一个 ASCII 代码,代表一个字符。在 Windows 操作系统中,常见的编码方式包括 ANSI、Unicode、Unicode big endian、UTF-8 等。由此,要实现对 TXT 文档文件的读取,关键在于区分不同的字符编码方式。不同编码 TXT 文件的开头字节内容为: ANSI 没有开头字节、Unicode 的开头字节内容为 FFEH、Unicode big endian 的开头字节内容为 FEFFH、UTF-8 的开头字节内容为 EFBBBFH。

本文对 ANSI、Unicode、Unicode big endian、UTF-8 四种编码方式的 TXT 文档进行解码,并在电子纸上实现文档显示。一般情况下,每个 ANSI 字符占用一个字节(8bits),每个 Unicode 字符占用两个字节(16bits),而 UTF-8 是变字节长度,英文只占一个字节,中文占三个字节,其中字符的显示需要字库文件的支持。TXT 文件的解析流程图如图 4 所示:

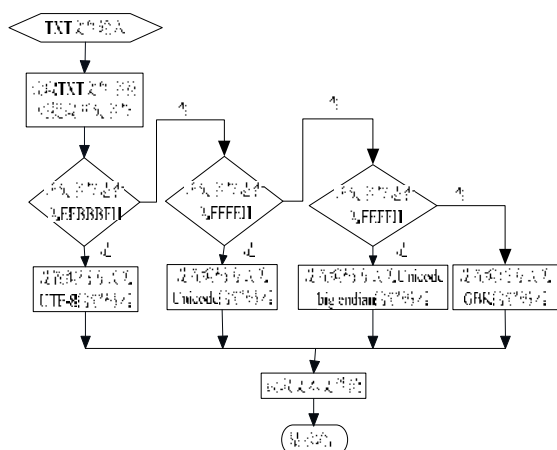


图4 TXT文件解析

2.4 TXT文档分页显示

为了实现文本文档的翻页及指定页阅读,在文本文档的加载过程中,首先对文本文档进行解码,其次对解码后的文本流进行分页,最后实现显示。文本文档分页流程图如图5所示:

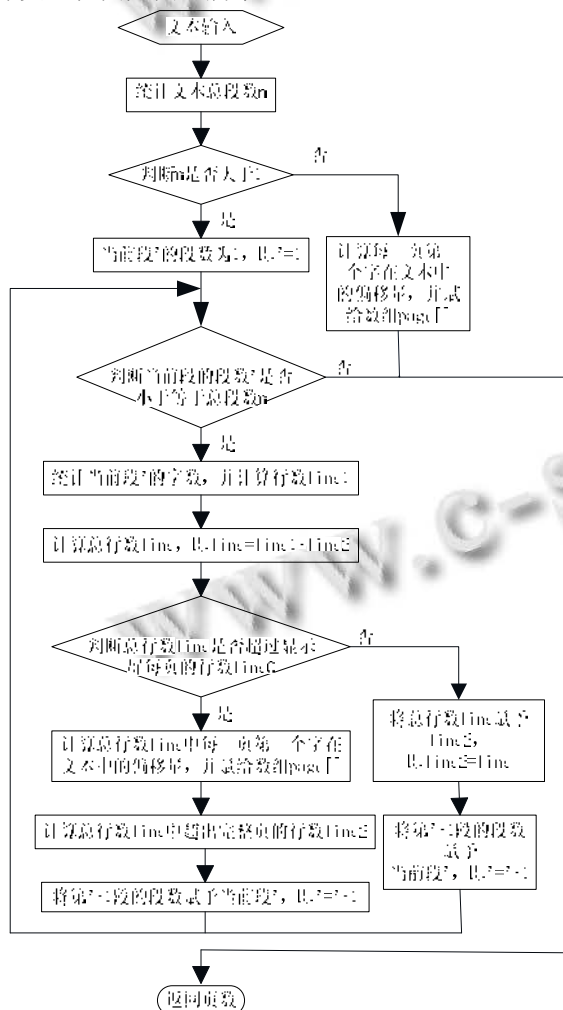


图5 文本文档分页

3 实验结果及分析

根据以上设计的模型及算法,编程实现电子书图形用户界面。实验中在完成电子书图形用户界面应用程序的编写后,编译X86版本并在qvfb上仿真运行,运行成功后,更换编译器,重新编译出适合在arm开发板上运行的版本。即对源程序进行交叉编译,将生成的可执行文件和相关的库文件拷贝到开发板上,实现移植,使得在电子纸上显示图形用户界面。在电子纸上显示的电子书图形用户界面如图6所示:

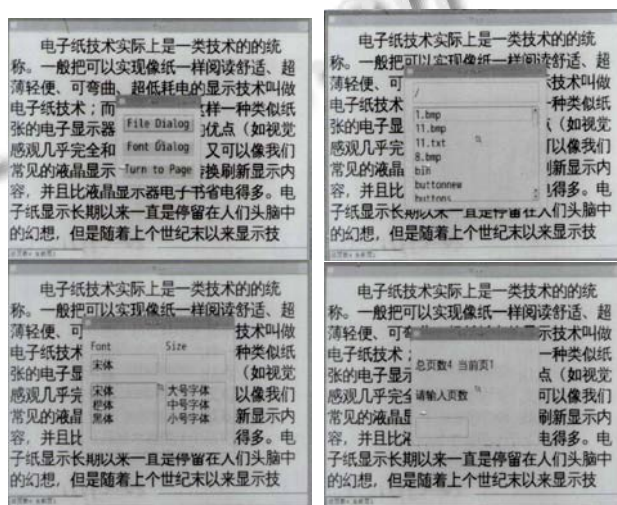


图6 电子纸上显示的电子书图形用户界面

在实验过程中,采用系统的特点如下:(1)利用Linux操作系统设计电子书图形用户界面,充分利用了Linux的开源性、可移植、可裁剪,节省了软件开发的时间,满足了电子书对小系统的要求,可以量身定制系统。(2)考虑到电子纸显示屏屏幕大小限制和电子纸显示屏在显示方面具有视觉效果好的优越性及灰阶少的局限性,以及硬件资源上按键个数的限制,设计的系统界面在满足用户需求的前提下,应尽量简单。

分析实验结果,可知实验中实现的电子书图形用户界面的优越性在于:(1)考虑到硬件设备的成本、硬件设备构造上的美观等以及电子纸显示屏的尺寸,利用对话框实现键盘操作,可以节省硬件上按键的设计以及最大程度地利用阅读界面。(2)利用键盘按钮实现光标在主窗口和对话框间切换,使得在用户需要时实时地显示主窗口或对话框。(3)在显示内容上实现了翻页和指定页阅读的模式,便于用户阅读文档文件。

4 结语

鉴于电子书系统的硬件资源及电子纸显示屏的屏

(下转第90页)

对上述的任意一测试数据,分别采用了3种不同的数据存储结构来组织数据,并针对指定的同一个点进行最近K邻域点(K=2)查找。对于本文的算法,八叉树的深度统一都设置成5。算法运行的环境为Dell Vostro 1710笔记本,CPU主频为2.1GHz,内存为2G,运行结果如表1所示

表1 结构建立及邻域查找耗时

点数	KD树	八叉树	本文算法
936770	0.78s	6.207s	0.241s
3176096	2.321s	25.452s	0.798s
6423463	5.522s	68.357s	1.452s

从表1可以看出,KD树的效率明显高于八叉树,这是由于随着数据量的增加,八叉树的深度也迅速增加,从而导致其检索速度变慢;KD树在数据量适中或较小的情形下具有极快的搜索速度,但随着数据量增加,同样会增加其检索的复杂度,导致速度很快下降;本文算法明显优于前两种算法,搜索速度大幅度提高,主要因为本文算法充分利用了两者的优势:先利用八叉树实现概略检索,再利用KD树实现检索精确定位,这样有效平衡了各环节的数据处理工作量,提高了整体效率。应当指出,这里将3组数据的八叉树深度都设置成5显得过于武断,实际运用应该做适当的测试,针对不同的数据类型及其数据量大小设置更为合适的八叉树深度值。

4 结语

为满足海量点云数据高效管理及组织的需要,本文提出了一种基于八叉树及KD树的混合型点云数据存储结构,该数据结构很好地利用了八叉树和KD树的技术优势,平衡了各环节的工作量,效率明显优于

单纯的八叉树及KD树索引结构,实验也证明了该数据结构能够实现海量数据环境下的快速邻域查找。

存在的问题及进一步工作:如前所述,针对不同的数据量及数据类型,八叉树的深度最优值应该是不同的,但目前只能通过预测试来确定,如何能够实现自动化及智能化设置八叉树的深度值还需进一步加以研究;此外,点云数据的处理效率除了直接和使用的算法有关外,还和数据的读取方式、计算机的物理限制(如内存限制等)相关联,当数据量达到计算机的物理极限无法一次性将数据读入内存时,则必须考虑使用缓存、预取、分级或多分辨率(LOD)等渐进传输的方式来实现数据处理。

参考文献

- 1 廖丽琼,罗德安.地面激光雷达的数据处理及其精度分析.四川测绘,2004,27(4):153-155.
- 2 董道国,薛向阳,罗航哉.多维数据索引结构回顾.计算机科学,2002,29(3):1-6.
- 3 宋扬,潘懋,朱雷.三维GIS中R树索引研究.计算机工程与应用,2004,(14):9-10.
- 4 惠文华,郭新成.三维GIS中的八叉树空间索引研究.测绘通报,2003,1:25-27.
- 5 郑坤,朱良S峰,吴信才,等.3D GIS空间索引技术研究.地理与地理信息科学,2006,22(4):35-39.
- 6 吴涵,杨克俭.基于kd树的多维索引在数据库中的运用.计算机应用,2007,26(9):37-39.
- 7 石波,卢秀山,陈允芳.基于kd-tree的建筑物散乱点云平面分割.测绘科学,2008,33(1):135-136.
- 8 路明月,何永健.三维海量点云数据的组织与索引方法.地球信息科学,2008,10(2):190-194.

(上接第219页)

幕特点,采用Qt图形用户界面开发工具,设计电子书图形用户界面,同时以文本文档显示为例,实现不同编码方式的文本文档的解码及分页显示。系统在满足用户基本阅读需求的前提下,最大程度地利用了电子纸显示屏的显示界面,节省了按键资源。

参考文献

- 1 张勇.电子书-引领数字阅读潮流.科技信息,2010,(28):379.

- 2 曾祥健,姚倩茹.浅谈国内电子书市场现状.财经视点,2010,(7).
- 3 郭晓琳.电子阅读器驱动程序设计与实现.成都:电子科技大学,2006.
- 4 吴蔚.任重道远电子书.中国计算机报,2007,12.
- 5 王丽洁.嵌入式Linux的图形界面技术研究与实现.长沙:国防科技大学,2006.