

Linux下本地文件快速搜索工具设计与实现

冯荷飞, 孙 前

(浙江科贸职业技术学院, 浙江 金华 321016)

摘 要: 本地计算机内文件数量逐年增长, 并且文件和目录还处于不断的变化中, 比如新增、删除、修改等, 因而用户想要快速定位到所需文件也是越发困难。随着Linux操作系统的推广, 普通Linux用户更需要一款易用的本地文件快速搜索工具。在业务逻辑模块中建立全盘索引、更新文件索引信息, 采用Tkinter库设计图形用户界面, 以SQLite嵌入式数据库管理索引数据, 并使用inotify接口调用Linux内核特性来监控文件系统的变化。基于此, 本文使用Python语言开发了一款轻量级的本地文件搜索工具, 并能够方便普通Linux用户的快速搜索需求。

关键词: Linux; 本地文件; 搜索; 快速

中图分类号: TP311 **文献标识码:** A

Design and Implementation of the Fast Searching Tool for Local Files in Linux

FENG Hefei, SUN Qian

(Zhejiang Vocational College of Technology and Trade, Jinhua 321016, China)

Abstract: Given that local files in computers increase year by year, and files and catalogues are in continuous changes, such as add, delete, modify, etc., it becomes more difficult for users to fast locate the required documents. With the promotion of the Linux operating system, ordinary Linux users need an easy-to-use fast searching tool for local files. In the business logic module, the paper sets up index, updates file index information, uses Tkinter library to design GUI, manages data with SQLite embedded database, and applies Inotify interface to call Linux kernel characteristics to monitor file system changes. Based on this, this paper uses the Python language to develop a convenient local file searching tool, which can facilitate the rapid search requirements of ordinary Linux users.

Keywords: Linux; local files; search; fast

1 引言(Introduction)

在现今信息化时代, 大数据概念已深入人心, 各种类型信息、资源层出不穷。个人计算机中的文件越来越多, 用户想要快速定位到所需文件也是越发困难。操作系统自带的文件搜索工具搜索速度较慢。因而在Windows平台下出现了一大批由第三方开发的本地文件快速搜索工具, 其中最著名的则是Everything软件^[1]。近年来随着Linux操作系统的大力推广, 众多国内Linux发行版中的中标麒麟、优麒麟、Deepin日益成熟, 不少普通用户也选择了Linux系统。而Linux系统下却没有简单易用的本地快速搜索工具。因此开发一款满足普通用户的轻量级本地文件搜索工具显得十分有必要。

2 Linux下的文件查找方法(The file locating method in Linux)

任何一种操作系统都由成千上万的文件组成, Linux系统更是认为“一切皆文件”。普通的文件、目录、字符设备、块设备、套接字等在Linux中都是以文件被对待。虽然类型不同, 但是对其提供的却是同一套操作接口。在Linux系统中

准确高效地确定一个文件在系统中的具体位置一般来说有三种搜索方法, 即使用文件管理器, 使用find命令, 使用locate命令。

2.1 文件管理器

文件管理器是Linux系统桌面版的必备工具, 常见的有Nautilus、Dolphin、Nemo、Peony。它们都提供了方便的文件管理功能, 包括浏览、复制、剪切、粘贴、删除、搜索等。文件管理器是普通用户最常用的文件搜索工具。但因其搜索文件时需要扫描整个磁盘文件, 因而搜索时间常常会很长。

2.2 find命令

find命令根据文件的属性进行查找, 如文件名、文件大小、所属组、所有者、访问时间、修改时间等。基本语法如下: # find path expression search-term。例如使用find命令根据文件名来查找特定文件时可以用: # find -name "test.file"。该命令会搜索整个目录树来查找名为test.file的文件, 并且会提供其存放位置。find命令有时会花费十几分钟来

查找整个目录树,特别是当系统中有很多文件和目录时搜索时间会更长。

2.3 locate命令

locate是一种比find速度更快的文件搜索方式。它事先构建索引数据库,通过匹配文件信息进而定位符合条件的文件。索引的构建过程需要遍历整个根文件系统,较为耗费资源,因此其常常在系统较为空闲时进行。locate属于非实时查找,使用locate搜索文件的语法为: # locate filename。例如,使用 # locate Tkinter命令可得到如图1所示的搜索结果。



图1 locate命令搜索结果

Fig.1 "Locate" command search results

此外locate有很多选项来过滤输出。如果要更新搜索数据库,可以运行命令: # updatedb。就如同find命令一样,locate查询文件得到的也只是文件的地址。如果想要定位到被查询文件,则还需另外的命令,比如: # nautilus [文件目录]。并且locate的文件索引数据库不是实时更新的,对于一些新文件可能会搜索不到。

3 软件结构设计(Software structure design)

本文所述的软件体量较小,总体结构采用简化的MVC模式。软件分为三个模块,采用将业务逻辑、后台数据与界面显示分离的方式来组织代码。使用模块分离的方式有代码耦合性低、重用性高的优点,并且当软件需求变更、算法改进、界面美化时更易于维护和修改。

3.1 业务逻辑模块

业务逻辑模块(Module.py)用于处理应用程序数据逻辑,它包含了一些函数和全局变量,其中主要函数见表1。这些函数实现了指定目录下文件索引建立、创建数据库、数据入库、数据查询以及文件索引实时更新等功能。

表1 业务逻辑模块中的主要函数

Tab.1 The main function in the business logic module

函数	描述
def create_db():	# 创建sqlite数据库,用于存储文件索引信息
def index_all_files(dir_path):	# 遍历dir_path磁盘目录,并保存得到的文件索引信息
def insert_to_db(file_path,file_name):	# 插入文件索引记录到sqlite数据库
def InitDB():	# 完成初始化工作,通过调用上述三个方法来建立数据库、获取文件索引、索引信息入库
def find_file(file_name):	# 在数据库中查找文件信息,主要工作是使用sql语句在数据库中查询
def updateDB(dir_path):	# 实时监控dir_path目录下的文件变化情况,如有变动立即更新文件索引到数据库中

3.2 后台数据模块

本文使用SQLite数据库进行后台数据管理。SQLite是一个嵌入式开源的关系数据库,其特点是自包容、零配置、无服务端和支持事务。在大多数情况下只要确保SQLite的二进制文件存在即可创建、连接和使用数据库。它能够支持Windows/Linux/Unix等主流操作系统,同时能够跟很多程序语言相结合,并且还有ODBC接口^[2]。

可使用如下python脚本创建SQLite数据库文件^[3]:

```
sqlite_con=sqlite3.connect('fileIndex.db')
sqlite_cur=sqlite_con.cursor()
```

```
sqlite_cur.execute('CREATE TABLE FOO (o_id
INTEGER PRIMARY KEY, file_path VARCHAR(260),
file_name VARCHAR(260))')
```

```
sqlite_con.commit
```

创建SQLite数据文件后就可以正常使用SQL语句来操作数据库了。可使用如下python脚本语句:

```
sqlite_cur.execute('INSERT INTO FOO (o_id,
file_path, file_name)VALUES(NULL,?,?)',[file_path.
decode('utf-8 '),file_name.decode('utf-8 ')])
```

将单个文件的索引信息作为一条记录存入数据库中。

3.3 界面显示模块

Python作为一种脚本语言,开发图形用户界面并不是它的强项。但开发人员仍然有多种选择,包括wxWidgets、PyQt、PyGTK和Tkinter等^[4]。

(1)wxPython是Python语言的一套开源GUI库,它是wxWidgets的Python封装,以Python模块的方式提供给用户。同时wxPython具有非常优秀的跨平台能力,能够运行在Windows系统、绝大多数的Unix或类Unix系统、Macintosh OS上。

(2)PyQt是Qt库的Python版本。它实现了一个Python模块集,有超过300个类,将近6000个函数。它是一个多平台的工具包,可以运行在大部分操作系统上,包括UNIX、Windows和Mac。PyQt采用双许可证,开发人员可以选择GPL或商业许可。

(3)Tkinter是Python的标准GUI库,是一个轻量级的跨平台图形用户界面开发工具。Tkinter是内置库,由一定数量的模块组成,与Python的结合度最好。Tkinter位于一个名为_tkinter的二进制模块中。只要安装好Python,就能通过import Tkinter导入并调用^[5]。

Tkinter的不足之处是缺少可视化界面设计工具,需要通过代码来完成窗口设计和元素布局。IDLE是开发python程序的基本IDE(集成开发环境),而IDLE本身也是用Tkinter编写而成。可见对于简单的图形界面,Tkinter还是能应付自如,因而使用TKinter作为本文程序界面也是最合适的。

4 数据实时更新(Real-time data update)

inotify是一个Linux的内核特性,提供了一种监控文件系

