

用 Python 语言分析银行自助设备交易流水文件

何惠英,付少波,安居,付兰芳

(军事交通学院 基础部 天津 300161)

摘要:无人值守银行自助设备的运行状况是一项非常重要的内容,维护者必须能够及时分析并处理设备所出现的各种故障。自助设备交易流水文件详细记录了设备的运行状况,为了分析该文件,基于当前流行的 Python 语言,并配合 wxPython GUI 包为软件开发平台,开发了自助取款设备运行状况分析程序,该软件具有开发周期短、界面友好、操作简单的优点。

关键词:python;流水记录文件;分析工具;ATM

中图分类号:TP311

文献标识码:A

文章编号:1004-373X(2010)05-131-04

Journal Log Files of ATM Based on Python

HE Huiying, FU Shaobo, AN Ju, FU Lanfang

(Academy of Military Transportation, Tianjin, 300161, China)

Abstract: ATM's running is very important, vindicator must analyse and deal with each fault in time and generally. ATM's running program is developed, based on popular Python language with wxPython GUI toolkit. This method has short development period and friendly interface and simple operation.

Keywords: python; journal log files; analysis tool; ATM

银行自助设备为用户的银行交易提供更多的方便,可以 24 h 为用户提供服务。目前随着银行卡发行量的增多,银行自助取款设备的使用频率在逐步提高。但因自助取款设备无人值守的特点,设备的运行状况便成了一个非常重要的关注点。目前除了实时的监控系统之外,流水记录文件的分析便是机器运行状况的主要参考内容。从机器流水记录中可以分析机器的运行状况、各周期交易量、正常及异常的交易明细等。大多自助设备的交易量很大,每日可达几百笔业务,这样,多台机器经过几个周期运行之后的数据量很大,这无疑给设备的故障分析工作带来了很大的难度。因此,专为流水记录文件写一个分析软件便显得非常迫切。基于该背景,介绍如何使用 Python 语言开发分析流水记录文件工具的过程。

1 Python 语言介绍

Python 是免费的解释性语言^[1],具有面向对象的特性,可以运行在多种操作系统之上,它继承了传统编译语言的强大性和通用性,同时也借鉴了简单脚本和解释语言的易用性。Python 具有清晰的结构、简洁的语法以及强大的功能,可以完成从文本处理到网络通信等各种工作,并且其自身已经提供了大量的模块来实现各

种功能,除此以外还可以使用 C/C++ 来扩展,甚至还可以将其嵌入到其他语言中。

Python 语言特点^[2]:

(1) 自由软件:遵循 GPL 协议,不需要支持任何费用,也不用担心版权问题,可阅读其源代码。

(2) 跨平台:Python 最初是在 MAC 操作系统下实现的,有很强的移植性,可运行在多种流水行操作系统之上,如 Linux, Windows, MAC 等。

(3) 功能强大:可以使用在多个领域,如系统编程,帮助用户完成繁琐的日常工作;科学计算,它简洁的语法可以像使用计算器一样来完成科学计算;快速原型,它省去了编译调试的过程,可以快速地实现系统原形;Web 编程,使用它可以编写 CGI,而现在流行的 Web 框架也可以使用 Python 实现。

(4) 可扩展:通过使用 C/C++ 可以对 Python 进行扩展;Python 也可以嵌入到 C/C++ 编写的程序之中。在某些情况下,它可以作为动态链接库的替代品在 C/C++ 中使用。

(5) 易学易用:语法简单;使用变量时无需事先声明;使用 Python 不必关心内在的使用,它会自动地分配、回收内存;Python 提供了强大的内置对象和方法,如本文使用的 open() 方法;使用 Python 可以减少其他编程语言的复杂性,例如在 C 语言中使用数十行代码实现的排序,而在 Python 中,可以使用列表的排序函

数轻易完成。

2 流水记录文件特性

流水记录文件是记录机器运行状态的 log 文件, 包含机器运行时的各种硬件状态、网络状态、用户操作记录、管理员操作记录等内容, 是分析机器运行状况的重要依据之一。

(1) 文件类型: 以日期命名的纯文本格式文件。

(2) 文件内容: 交易时机器状态及相关交易信息; 操作员对机器的操作记录; 待机时机器当前状态; 网络连接状态; 机器重启时的相关信息等。

(3) 文件关键字: 插卡日期、卡号、交易类别、交易金额、交易成功标志、交易失败标志、失败原因、网络故障等。

(4) 文件位置: 位于 ATM 机器硬盘内, 部分文件同步于银行后台主机。

3 基于 Python 语言开发的分析流水记录文件工具

3.1 开发原理

通过 Python 语言的内置函数 `open()` 来打开流水文件^[3], 返回一个文件对象, 然后用 `for` 循环遍历这个流水文件对象, 通过字符串匹配的方式来查找主要关键字并保存相关信息, 然后计算生成所需要的各种字符串信息或列表供分析使用。

3.2 环境介绍

开发语言: Python 2.5.4

界面框架: wxPython 2.8 - win32 - unicode - 2.8.10.1 - py25;

运行环境: 所有 Unix 衍生系统 (Linux, MacOS X, Solaris, FreeBSD 等), Win32 家族 (Windows NT, 2000, XP 等), 早期平台: MacOS 8/9, Windows 3.x, DOS, OS/2, AIX^[4]。

3.3 程序实现

该程序采用面向对象的编程方法来对各个模块的功能进行实现, `JrnAnalyse.pyw` 来初始化程序, 并调用 `mainframe` 模块进行程序界面显示, 在 `mainframe` 模块中调用 `jrn` 模块进行流水文件分析, `jrn` 模块返回分析结果给 `mainframe` 模块进行显示或是存盘。程序框图如图 1 所示。

3.3.1 Python 实现处理文件

文件也可以看作是 Python 中的数据类型。当使用 Python 的内置函数 `open` 打开一个文件时, 返回一个文件对象。其原型如下所示^[5]:

`open(filename, mode, bufsize)`

其参数含义如下:

`filename`: 要打开的文件名。

`mode`: 可选参数, 文件打开模式。

`bufsize`: 可选参数, 缓冲区大小。

其中 `mode` 可以是 "r" 表示读方式打开文件; "w" 表示以写方式打开文件; "b" 表示以二进制方式打开文件。

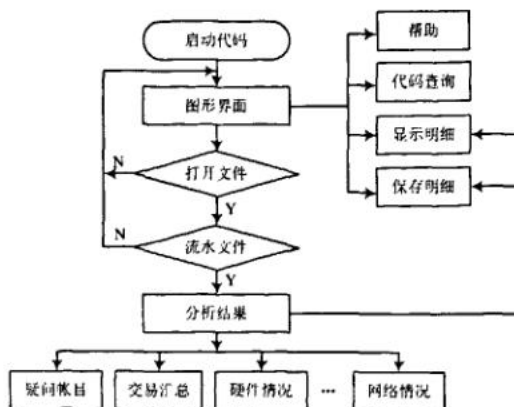


图 1 程序框图

`open()` 成功执行并返回一个文件对象之后, 所有对该文件的后续操作都将通过这个“句柄”进行。文件方法可以分为四类: 输入, 输出, 文件内移动, 以及杂项操作。以下介绍要用到的几个最基本的方法:

`file.write()`: 向文件中写入字符串, 使用此方法来保存成功交易明细或疑问帐信息^[6]。

`file.close()`: 关闭打开的文件

对所打开的文件对象进行遍历来查找所需要的信息, 是进行流水记录分析的重要过程, 示例如下:

```

file=open("filename","rU")
# U 参数可解决文件结束符不统一的问题, 如\r会替换
# 为\n, 这样在用 for 循环的时候就统一了行结束标志。
for line in file: # 使用 for 循环对文件对象 file 进行遍历,
                  # line 为 filename 中的每一行数据。
    print line    # 此例为打印 line 数据到屏幕
  
```

3.3.2 wxPython 实现 GUI 界面显示

wxPython 是跨平台工具库 wxWidgets 的封装。wxWidgets 库是由 C++ 编写的, 它类似于 Windows 的 MFC^[7]。wxWidgets 提供了对多种操作系统的支持。由于它良好的可移植性, wxPython 也具备了跨平台的能力。在 Python 中使用 wxPython 可以编写具有跨平台能力的 GUI 脚本。

这里介绍能够显示图形界面的几个主要步骤:

```

# -*- coding:utf-8 -*-
import wx
class MyApp(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None, title='Hello, wxPython!')
        frame.Show()
        return True
app = MyApp()
  
```

```
app.MainLoop()
```

开头指定使用 utf-8 字符编码,可显示中文字符串。首先导入 wx 模块,即 wxPython。然后继承 wx 模块中的 App 类创建了一个 MyApp 类^[8]。在 MyApp 类中重载了 OnInit 方法。OnInit 方法是窗口初始化的一部分,在该方法中创建了一个窗口框架,并显示该窗口。一般来讲,OnInit 方法最后应返回 True。最后两行,将 MyApp 类实例化成 app 对象,然后调用其 MainLoop 方法进入消息循环。

3.3.3 程序结构

(1) 程序启动文件:JrnAnalyse.pyw

```
# -*- coding:utf-8 -*-
"""
atmtool.py: A tool for analyse atm journal files
"""
# 文件注释

import wx                # 导入 wxpython 模块
import mainframe          # 导入 mainframe 模块
class AtmApp(wx.App):    # 定义启动类
def __init__(self, redirect=False, filename=None):
    # 重载生成对象时初始化方法
wx.App.__init__(self, redirect, filename)
def OnInit(self):        # 重载窗口初始化方法
    self.main = mainframe.CreatMainframe(None)
    self.main.Show()
    self.SetTopWindow(self.main)
    return True
if __name__ == "__main__": # 作为独立程序启动
    application = AtmApp()
    application.MainLoop()
```

(2) 界面显示模块:mainframe.py

在该模块中定义了各种显示组件、界面组件操作方法;各方法列表如下:

```
def __init__(self, parent): # 定义并初始化各种显示组件
def opendir(self):         # 定义打开流水文件方法
def runJrn(self):          # 定义调用流水分析模块方法
def On_Quit(self, event):  # 定义退出程序方法
def On_Help(self, event):  # 定义帮助文件方法
def On_About(self, event): # 定义关于程序方法
def OnItemSelected(self, event): # 定义点击列表方法
def getColumnText(self, index, col):
    # 定义点击选择列表内容方法
def scroll2line(self, current_page):
    # 定义定位到疑问流水位置的方法
def OnOkTrans(self, event): # 定义保存成功交易方法
```

(3) 流水分析模块:jrn.py

该模块进行每个流水记录文件的分析工作。

```
def get_date_card(self, line, jrnfp)[9]:
    # 定义获取用户交易日期卡号方法
def get_cwd_amount(self, line):
    # 定义获取用户取款交易金额方法
def get_dep_amount(self, line, jrnfp)
    # 定义获取用户存款交易金额方法
def total(self, jrns): # 定义计算多周期交易汇总信息方法
```

```
def question(self, jrns): # 定义获取疑问交易方法
def hardware(self, jrns): # 定义获取机器硬件异常方法
if __name__ == "__main__": # 流水模块自省语句判断
```

下面将主要流水模块中的分析情况举例说明如下:
获取用户交易日期和卡号例子:

```
def get_date_card(self, line, jrnfp):
    date=""
    card_no=""
    if 'AcceptCard' in line: # 获取插卡日期
        date=line[12:-2]
        cwd_amount="" # 如果插卡则将 cwd_amount 清空
    elif 'CardType:' in line # 获取插卡卡号
        card_no=line # 将卡号保存在 card_no 变量中
    return date, card_no # 返回插卡日期和卡号
```

获取用户取款交易金额例子:

```
def get_cwd_amount(self, line, jrnfp):
if 'CWD Amount:' in line:
    # 如果关键字-CWD Amount 在本行中
    temp=line.split()[1]
    amount=int(temp.split(':')[1])
    # 将取款金额字符串转成整形后保存
    return amount # 返回取款金额
```

获取用户取款成功标志例子:

```
if 'WITHDRAWAL OK' in line[10]:
    # 如果取款成功标志在本行中
    cwd_ok_amount += amount # 累加取款金额
    cwd_ok_num += 1 # 累加取款笔数
    money_taken=True
    # 只要条件匹配一次,就将取钞标志置真
    succeed_trans += (date.strip() + " " + card.strip() +
    " " + str(cwd_amount) + " cwd succeed\n")
    # 保存成功交易明细
```

获取用户存款交易金额例子:

```
def get_dep_amount(self, line, jrnfp):
    dep_amount=0
    if 'Amount:' in line and 'TELLER:' in line:
        # 如果存款交易关键字在本行中
        temp=line.split()[1]
        dep_amount=int(temp.split(':')[1])
        # 将存款金额字符串转成整形后保存
    return dep_amount # 返回存款金额
```

获取用户存款成功标志例子:

```
if "CASH IN NOTES OK" in line:
    # 如果成功取款标志在本行中
    dep_ok_amount += dep_amount # 累加存款金额
    dep_ok_num += 1 # 累加存款笔数
    cash_stored=True
    # 只要条件匹配一次,就将存钞标志置真
    succeed_trans += (date.strip() + " " + card.strip() +
    " " + str(dep_amount) + " dep succeed\n")
    # 保存成功交易明细
```

获取管理员清机加钞标志:

```
if '---RefillCash---' in line: # 获取加钞标志
    Here do what you want
```

程序运行示例:图 2 是打开了多个流水记录文件后程序的运行显示效果图。



上左:疑问交易明细 上右:疑问交易概况
下左:各周期交易汇总 下右:硬件关键信息

图 2 运行显示效果图

4 结 语

本文所述的基于 Python 语言的流水分析工具在分析自助设备运行状况中得到了很好的应用,大大减少了设备维护者的工作量。在一个交易周期还未结束时,就可提前发现已经出现的疑问帐务,对于由于网络问题而导致服务器记录缺失现象有很大帮助,同时也提高了

作者简介:何惠英 女,1976 年出生,天津人,硕士研究生,助教。主要研究方向为电器可靠性测试、测试分析工具。

(上接第 130 页)

4 结 语

本文提出了一种以 ARM9 处理器为核心,实时自动采集粮仓图像数据,通过 ZigBee 网络进行无线传输的粮仓图像数据采集终端。该终端通过摄像头非接触式采集,布设方便,无需改造现有网络与设备;自带照明电路,可实现 24 h 不间断工作;无线自组网,使用方便,具有广泛的应用价值。

参 考 文 献

- [1] 王艳秋,曾维鲁,岳宇君. ZigBee 技术在仓储管理中的应用[J]. 通信技术,2008(11):205-207.
- [2] 周怡頔,凌志浩,吴勤勤. ZigBee 无线通信技术及其应用探讨[J]. 自动化仪表,2005,26(6):5-9.
- [3] 王盟,马艳峰,徐晓辉,等. 一种无线数据采集及远程监测系统的设计[J]. 现代电子技术,2009,32(1):30-32.

作者简介:雷文礼 男,1982 年出生,助教,硕士。主要研究方向为嵌入式系统。

李建新 男,1963 年出生,教授,硕士。主要研究方向为信号与信息处理、电子电路设计。

张水利 女,1974 年出生,讲师,硕士。主要研究方向为图像数据处理。

参 考 文 献

- [1] Mark Lutz. Learning Python[M]. 3 版. 南京:东南大学出版社,2008.
- [2] [美] Wesley J Chun. Python 核心编程[M]. 2 版. 宋吉广,译. 北京:人民邮电出版社,2007.
- [3] 孙广磊. 征服 Python——语言基础与典型应用[M]. 北京:人民邮电出版社,2007.
- [4] NOEL RAPPIN. wxPython in Action[M]. Manning Publications Co., 2006.
- [5] Alex Martelli. Python in a Nutshell[M]. 2nd Edition. O'Reilly, 2006.
- [6] Magnus Lie Hetland. Beginning Python From Novice to Professional[M]. Second Edition. 2008.
- [7] Tarek Ziadé. Expert Python Programming[M]. Packt Publishing Ltd. 2008.
- [8] Jim Knowlton. Python?: Create - Modify - Reuse[M]. Wiley Publishing Inc., 2008.
- [9] Mark Lutz. Programming Python[M]. 3rd Edition. O'Reilly, 2006.
- [10] David Mertz. Text Processing in Python[M]. Addison Wesley, 2003.

- [4] 逯绍峰,逯跃峰. 基于无线传感器网络的有毒气体监测系统的设计[J]. 现代电子技术,2007,30(23):158-162.
- [5] Chipcon. A True System-on-Chip Solution for 2.4 GHz IEEE802.15.4/ZigBee[EB/OL]. <http://focus.ti.com/lit/dink/cc2430.pdf>, 2007.
- [6] 陈曦. ARM9 嵌入式技术及 Linux 高级实践教程[M]. 北京:北京航空航天大学出版社,2005.
- [7] 孙弋. ARM-Linux 嵌入式系统开发基础[M]. 西安:西安电子科技大学出版社,2008.
- [8] 赵景宏,李英凡,许纯信. ZigBee 技术简介[J]. 电力通信系统,2006,27(10):54-56.
- [9] 孙纪坤,张小全. 嵌入式 Linux 系统开发技术详解——基于 ARM[M]. 北京:人民邮电出版社,2006.
- [10] 杜春雷. ARM 体系结构与编程[M]. 北京:清华大学出版社,2003.
- [11] 刘森. 嵌入式系统接口设计与 Linux 驱动程序开发[M]. 北京:北京航空航天大学出版社,2006.

用Python语言分析银行自助设备交易流水文件

作者: [何惠英](#), [付少波](#), [安居](#), [付兰芳](#)
作者单位: [军事交通学院, 基础部, 天津, 300161](#)
刊名: [现代电子技术](#) 
英文刊名: [MODERN ELECTRONICS TECHNIQUE](#)
年, 卷(期): [2010, 33 \(5\)](#)

参考文献(10条)

1. [David Mertz](#) [Text Processing in Python](#) 2003
2. [Mark Lutz](#) [Programming Python](#) 2006
3. [Jim Knowlton](#) [Python?:Create-Modify-Reuse](#) 2008
4. [Tarek Ziadé](#) [Expert Python Programming](#) 2008
5. [Magnus Lie Hetland](#) [Beginning Python From Novice to Professional](#) 2008
6. [Alex Martelli](#) [Python in a Nutshell](#) 2006
7. [NOEL RAPPIN](#) [wxPython in Action](#) 2006
8. [孙广磊](#) [征服Python—语言基础与典型应用](#) 2007
9. [Wesley J Chun](#); [宋吉广](#) [Python核心编程](#) 2007
10. [Mark Lutz](#) [Learning Python](#) 2008

本文链接: http://d.g.wanfangdata.com.cn/Periodical_xddz.js201005042.aspx