

用 Python 语言跨平台分析处理电子表格

何惠英¹, 王兆合², 王小宇³, 纪蕾⁴

(1. 军事交通学院 天津 300161; 2. 南京合荣欣业信息技术有限公司 江苏 南京 211100;
3. 天津市电力公司东丽供电分公司 天津 300000; 4. 天津市电力公司城东供电分公司 天津 300000)

摘要:在日常工作中经常需要分析和处理电子表格,这就容易涉及很多重复性的工作。有时还要求软件能在不同操作系统下使用。鉴于此背景,基于降低手工处理复杂电子表格重复性操作强度,且不用更改代码便可以在不同操作系统(Windows、Linux、MAC 等等)上运行的目的,采用目前流行的动态语言 Python 编程的方式实现了电子表格的处理任务。通过实际分析几十万条客服工单电子表格,得出了通过编写程序可以大幅度降低人工操作强度,减少工作时间,且能跨平台使用的结论。

关键词: Python; 电子表格; 跨平台; 脚本编程

中图分类号: TP311.1

文献标识码: A

文章编号: 1674-6236(2011)09-0067-04

Analyzing and processing worksheet on multi-platform based on Python

HE Hui-ying¹, WANG Zhao-he², WANG Xiao-yu³, JI Lei⁴

(1. Academy of Military Transportation, Tianjin 300161, China;

2. Nanjing Hiward Technology Co. Ltd, Nanjing 211100, China;

3. Tianjin Electricity Company DongLi Power Supply Subsidiary, Tianjin 300000, China;

4. Tianjin Electricity Company Chengdong Power Supply Subsidiary, Tianjin 300000, China)

Abstract: That is often needed analyzing and processing worksheets in daily work, so it is easy to involve much repetitive works. Sometimes it is also required that the software can be used in different operating systems. In view of this background, this paper based on lowering repeatability operating intensity of manual processing complex worksheets, and running in different operating system (Windows, Linux, MAC, etc.) without changing its codes, adopts popular dynamic program languages Python to realize the worksheets processing tasks. Through the practical analysis about hundred thousand customer service worksheets, it is concluded that using the program described can lower operating intensity, reduce working hours, and can be used in different platforms.

Key words: Python; worksheet; multi-platform; scripting programme

在日常工作中经常需要分析处理一些较复杂的电子表格,如在通信、金融等许多领域需要处理各种情况的客户服务信息,这些信息通常是包含很多项目的电子表格,手工统计起来较繁琐。当然,可以使用 MS Office Excel 办公软件来实现,但是其缺陷是不能跨平台使用,即有时需工作在 Windows 系统下,有时需工作在 Linux 系统下,这样就不能在这两个系统下同时使用相同的 Excel 办公软件。基于此背景,本文介绍如何用 Python 语言实现跨平台分析处理电子表格。

1 开发环境

使用的编程语言为跨平台的编程语言 Python 2.5.4,这样在 Windows 下和 Linux 下都安装 Python 编程环境,就可以执行相同的脚本来生成相同格式的电子表格文件。

读取电子表格模块使用 Python 2.5.4 自带的 CSV 模块,

CSV 模块是处理单个电子表格常用的以逗号分隔的文本文件。使用很广泛且在 Python 中很方便。读取电子表格模块还可以使用 xlrd,目前的最新版本是 0.7.1,可以读取多个 sheet。本文中介绍 CSV 模块的使用。

写入电子表格模块使用第三方的 pyExcelerator 模块,该模块只需要安装 Python 2.4 以上版本就可以正常运行,可工作在 Windows 和 Linux 以及 MAC 下。

1.1 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。Python 可以完成从文本处理到网络通信等各种工作。Python 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

收稿日期:2010-12-08

稿件编号:201012026

作者简介:何惠英(1976—),女,天津人,硕士研究生,讲师。研究方向:电器可靠性测试,基于 Python 语言的测试平台。

使用 C/C++ 来扩展 Python, 甚至还可以将 Python 嵌入到其他语言中。

目前在国外很多大学里, 已经将 Python 作为首选的课程; 在很多软件工程里, Python 语言也在大量的使用, 如著名的 Web 应用服务器 Zope, Django 等等。

Python 语言特点:

1) 自由软件 遵循 GPL 协议, 不需要支付任何费用, 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, 可运行在多种流行操作系统之上, 如 Linux, Windows, MAC 等等。

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。

1.2 CSV 模块介绍

CSV (Comma Separated Values) 格式是在电子表格和数据库中最常用的导入导出格式^[9]。但是没有 CSV 格式标准, 所以许多程序自行定义 CSV 的格式。缺少标准意味着不同的程序产生和处理 CSV 格式数据经常存在着细微的差别。这种差别导致在处理多个 CSV 格式数据时比较麻烦。比如在 Windows 下用 excel 生成的 CSV 格式数据与在 Linux 下用 OpenOffice 生成的 CSV 格式数据存在细微的差别。所以要跨平台处理 CSV 格式数据, Python 中的 CSV 模块便可兼容完成。该模块在 Python 2.3 中引入, 可以忽略读写 CSV 格式数据的细节。

模块用法:

1) 先要导入模块: import csv

2) 使用 csv 的读功能读入打开的 csv 文件: reader = csvreader(open(file, "rb"))^[9], 从而得到一个 Python 的对象 reader, 以后可以使用该对象的现有方法, 来实现对 csv 电子表格数据的处理任务。

1.3 pyExcelerator 模块介绍

pyExcelerator 模块是一个可以生成 Excel 97/2000/XP/2003 和 OpenOffice Calc 兼容表格的模块^[10]。它支持 UNICODE 编码、多种格式化设置、打印等特性。它只需要 Python 2.4 或

以上的环境就可以运行。

模块用法:

1) 先要导入模块: from pyExcelerator import *

注: 导入 pyExcelerator 模块的所有空间属性和方法

2) 生成一个空的工作簿并命名: workbook = Workbook()^[10]

3) 在工作簿中添加表格: baoxiao_sheet = workbook.add_sheet(u'明细')

注: 可以添加多个不同名字的表格(sheet)

4) 在表格中添加数据: baoxiao_sheet.write(0, 0, "测试在 0 行 0 列")

5) 保存工作簿: workbook.save('我的表格.xls')

2 开发过程

2.1 准备工作

1) 安装 Python 编程环境 Python 开发环境为免费的开源项目, 可以到 Python 的主页上下载相应操作系统的安装版本。

以 Windows 系统为例, 执行下载的文件 Python-2.5.4.msi, 按提示安装完成后, 便具备了 Python 编程开发环境。在开始菜单中有交互式的命令行工具 Python 和图形界面开发工具 IDLE。在实际开发过程中可以使用 IDLE, 也可以使用自己熟悉的集成开发工具, 如 PythonWin、Komodo、BlackAdder、UliPad 等等。如果个人不喜欢集成开发工具, 还可以使用文本编辑器写 py 脚本, 如 vim、emacs、UltraEdit、记事本等等。

注: 如果想做成具有 GUI 图形界面外观的程序, 可以使用 Python 内置的库 tkinter, 或是第三方 GUI 库, 如 wxPython、PyQt、PyGtk、win32api、win32MFC 等等。

2) 安装 pyExcelerator 模块 pyExcelerator 模块可以在 <http://sourceforge.net/projects/pyexcelerator/>^[10] 上免费进行下载, 下载后执行 Python /setup.py install 进行安装。安装后可以在 Python 安装目录下的 D:\Python27\Lib\site-packages\pyExcelerator (以 Python 安装在 D 盘根目录下为例) 下看到该模块的相关文件。

3) 准备需要处理的电子表格文件

2.2 编写程序

1) 导出电子表格为 CSV 格式文件

用相应的电子表格软件将要处理的电子表格另存为 CSV 格式, 在 Windows 下可以使用 Microsoft 的 Excel 软件, 在 Linux 下可以使用 OpenOffice 的 Calc 软件, 都非常方便。

2) CSV 格式文件分析

-*- coding: utf-8 -*- # 这行指定使用 utf-8 文件编码以便处理中文或其他非英语文字。

import csv

导入 Python 自带的 csv 模块。通过 Python 内置的 import 命令可以将要使用的模块导入到 Python 的名字空间中, 供以后使用。

fiel = r'c:/myfile.csv'

指定要处理的文件路径 (也可设置变量请用户输入)。
 字符 'r' 表示使用原始字符串, 可以避免多输入 Windows 中的 '\\ 路径。在 Python 中可以都使用 '/' 来表示路径, 即使是在 Windows 系统中。

class CSV_Analyse: # 定义一个名字为 CSV_Analyse 的类
 # class 是 Python 的内置关键字, 可以由用户定义一个类, 后面是类的名字, 通常类名开始用大写字母表示。

def __read_csv(self, file):

定义一个名字为 __read_csv 的方法, 来读取 csv 文件。
 Def 是 Python 内置关键字, 可以由用户定义一个函数或是方法, 在类中称为方法。

使用 csv 的 reader 方法来读取打开的文件对象, 赋值给 reader 变量。

reader = csv.reader(open(file, "rb"))

调用 codes 方法返回表格文件的所有行, 这里使用了列表解析的技术, 比普通的 for 循环速度要快。

return [line for line in self.codes(reader)]

定义一个名字为 codes 的方法, 来将 csv 文件中的中文字符进行编码

def codes(self, reader) []:

使用 for 循环将表格的一行中的所有单元格字符编码转为 unicode 编码。

"Unicode(统一码、万国码、单一码)是一种在计算机上使用的字符编码。它为每种语言中的每个字符设定了统一并且唯一的二进制编码, 以满足跨语言、跨平台进行文本转换、处理的要求。"

for line in reader:

yield [unicode(cell, 'gb18030') for cell in line] # yield 为生成器表达式

def result_list(self, file) # 定义返回结果列表

pm_call = [] # 定义一个空的列表变量

all_rows_no_title = self.__read_csv(file)[1:] # 如不需要表头, 可用这行去掉; # [1:] 指内容分片, 默认从 0 开始, 这里从 1 开始可以去掉表头。

for line in all_rows_no_title: # 用 for 循环遍历表格内容的所有行

if "PM" in line[29]:

进行 if 条件判断处理, 例子的 "PM" 关键字在行的第 29 个位置

pm_call.append([line[35], line[9], line[8], line[1], line[38], line[40], line[29], line[7]])

append 是 Python 列表的追加内容的方法。

else:

else 语句在 if 条件表达式不成立时执行。

other_call.append([line[35], line[9], line[8], line[1], line[38], line[40], line[29], line[7]])

pm_call.sort(lambda x, y: (cmp(x[1], y[1])))

将 pm_call 按第 2 个单元格来排序, sort 是列表的排序方法

other_call.sort(lambda x, y: (cmp(x[1], y[1])))

将 other_call 按第 2 个单元格排序

return pm_call, other_call

返回 pm_call 和 other_call 两个列表, return 是 Python 的关键字, 此处给调用函数或方法返回一个元组 pm_call, other_call。元组是 Python 内置的类型, 和列表相似, 但元组里的元素不能修改。可以用类似 [1:] 的切片方法来取元组的值。注意: 在 codes 方法中的 'gb18030' 为中文编码, 比 'gbk' 和 'gb2312' 能够表示更多的汉字。比如某些生僻字只有在 'gb18030' 中才有。

3) 生成电子表格文件

-*- coding: utf-8 -*- # 这行指定使用 utf-8 文件编码以便处理中文或其他非英语文字。

from pyExcelerator import *

从 pyExcelerator 模块中导入所有属性和方法。form import 也是 Python 导入模块的一种方法, 可以导入所需模块的所有属性和方法。

title = ('Ser.Station', 'Branch', 'End User', 'Serial No.')

定义一个表格的第一行表头元组 (如服务站名称、所属支行、最终用户、设备序列号) 等等

定义生成 xls 表格函数, 参数为方法 result_list 返回的元组

def creat_xls(call_tuple) []:

workbook = Workbook() # 生成一个空的工作簿, 名字为 workbook

mingxi_sheet = workbook.add_sheet(u'明细')

在 workbook 工作簿中添加名字为 '明细' 的 sheet, 并赋值给 mingxi_sheet 变量

baoxiu_sheet = workbook.add_sheet(u'报修')

在 workbook 工作簿中添加名字为 '报修' 的 sheet, 并赋值给 baoxiu_sheet 变量

pm_sheet = workbook.add_sheet(u'PM')

在 workbook 工作簿中添加名字为 'PM' 的 sheet, 并赋值给 pm_sheet 变量

for col_num, item in enumerate(title): # 使用 for 循环写入固定标题头

baoxiu_sheet.write(0, col_num, item)

pm_sheet.write(0, col_num, item)

for row, line in enumerate(call_tuple[0]): # 使用 for 循环写入 PM 的工单

for col, item in enumerate(line[0:-1]):

pm_sheet.write(row+1, col, item)

for row, line in enumerate(call_tuple[1]): # 使用 for 循环

写入其他工单

```
for col,item in enumerate(line[0:-1]):
    baoxiu_sheet.write(row+1,col,item)
for col_num in range(len(title)): # 用 for 循环设置列的
```

宽度

```
baoxiu_sheet.col(col_num).width = 5000
pm_sheet.col(col_num).width = 5000
workbook.save('我的表格.xls')
# 保存成文件, '我的表格.xls' 也可以是用用户输入的变量
```

名

```
4) 调用 2) 和 3) 中的类和函数生成最终电子表格文件
dealcall = CSV_Analyse() # 生成类 CSV_Analyse() 的
```

实例

```
creat_xls(dealcall.result_list(file)) # 最终生成 '我的表格.xls' 文件
```

3 结束语

以上是生成所需电子表格的主要代码, 再加一些自己要处理的详细内容和一些要注意的细节就可以运行在 Windows 和 Linux 操作系统之上。

通过以上描述, 实现了跨平台处理电子表格的功能, 同时可以简化重复的日常电子表格统计工作。在软件使用的实际操作中, 还有一些细节需要注意, 罗列如下: 1) csv 模块只能处理单个表格, 不适合要处理多个表格; 如果需要处理含有多个表格的文件, 可以使用 xlrd 模块。也可以用 pyw32in 的 COM 组件来操作电子表格文件。2) csv 模块目前还不支持

unicode 编码, 需在程序中自行转换; 3) pyExcelerator 模块目前还不能处理特别复杂的样式; 4) 在 Windows 和 Linux 系统中需注意文件路径表示差异, 如“\”和“/”^[10]; 但统一使用“/”, Python 可以自动转换。5) Python 是解释性语言, 在运行时需要 Python 语言环境。

参考文献:

- [1] Mark Lutz. Learning Python 3rd[M]. 南京: 东南大学出版社, 2008.
- [2] Chun WJ. Python 核心编程[M]. 2 版. 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] Rappin N. wxPython in Action [M]. Manning Publications Co., 2006.
- [5] Martelli A. Python in a Nutshell 2nd Edition [M]. O'Reilly, 2006.
- [6] Hetland M L. Beginning Python from novice to professional [M]. Second Edition, 2008.
- [7] Ziade T. Expert Python programming[M]. Published by Packt Publishing Ltd, 2008.
- [8] Knowlton J. Python®: Create-Modify-Reuse[M]. Published by Wiley Publishing, Inc., 2008.
- [9] Lutz M. Programming Python[M]. 3rd Edition. Publisher: O'Reilly, 2006.
- [10] Mertz D. Text processing in Python[M]. Publisher: Addison Wesley, 2003.

(上接第 66 页)

声音仿真研究[J]. 火力与指挥控制, 2008, 33(10): 103-105.
WANG Run-gang, TANG Ke-qun, LIAN Guang-yan, et al. Research on real-time 3D sound simulation based on DirectSound[J]. Fire Control and Command Control, 2008, 33(10): 103-105.

[4] 赵欣, 刘丽. 基于 DirectSound 的飞行训练器音响系统仿真[J]. 计算机仿真, 2006, 23(3): 250-252.

ZHAO Xin, LIU Li. Simulation of FTD sound system with DirectSound[J]. Computer Simulation, 2006, 23(3): 250-252.

[5] 苑春苗, 杨清永. 基于 DirectSound 技术的音频捕获与多路

混音的实现[J]. 计算机工程与设计, 2007, 28(3): 613-614, 634.

YUAN Chun-miao, YANG Qing-yong. Method of audio capture and multi-audio mixing based on DirectSound technology[J]. Computer Engineering and Design, 2007, 28(3): 613-614, 634.

[6] 万刚, 王蒙, 武志强, 等. 虚拟环境中真实感声音的生成[J]. 测绘学院学报, 2004, (1): 50-53.

WAN Gang, WANG Meng, WU Zhi qiang et al. The Creation of Realistic Sound in Virtual Environment [J]. Journal of Institute of Surveying and Mapping, 2004, (1): 50-53.

欢迎订阅 2011 年度《电子设计工程》(半月刊)

国内邮发代号: 52-142 国际发行代号: M2996 订价: 6.00 元/期 144.00 元/年

用Python语言跨平台分析处理电子表格

作者: [何惠英](#), [王兆合](#), [王小宇](#), [纪蕾](#), [HE Hui-ying](#), [WANG Zhao-he](#), [WANG Xiao-yu](#),
[JI Lei](#)

作者单位: [何惠英, HE Hui-ying\(军事交通学院, 天津, 300161\)](#), [王兆合, WANG Zhao-he\(南京合荣欣业信息技术有限公司, 江苏, 南京, 211100\)](#), [王小宇, WANG Xiao-yu\(天津市电力公司东丽供电分公司, 天津, 300000\)](#), [纪蕾, JI Lei\(天津市电力公司城东供电分公司, 天津, 300000\)](#)

刊名: [电子设计工程](#)

英文刊名: [ELECTRONIC DESIGN ENGINEERING](#)

年, 卷(期): 2011, 19(9)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_dzsjgc201109020.aspx