

用Python做小工具——pypy程序的打包

前言

在前面介绍的pypy打包的过程中，我遇到了一个非常棘手的问题。那就是pypy的标准库中，里面包含了很多我们并不需要的库文件。例如里面的一些测试文件，一些没有引用到的文件等等。单纯靠手工去删除的话是非常的麻烦的。而如果不对这些文件进行删除，便会导致新的问题。那就是当软件被打包成单个文件的时候，需要解压大量的小文件。导致软件的启动时间非常的长。这对一个小程序来说。启动时间长是非常难堪的。于是我便想办法。如何才能把标准库中的不必要文件去掉呢。刚开始我是一个库一个库的删除。这是一个毫无意义的工作。因为要是有了新的需求了。又要重新删除等等。在思考了几个小时后，我有了以下几点思路。

打包思路

第一种方式是自己写一个脚本去分析每个软件的import等等。然后把所有import的包都给找到。[pyinstaller](#)就是使用的这种方式。就需要去写很多程序去分析各种导入的库。而有时候库的导入并不全是py文件。所以这种方法要靠编程的话，可能就像pyinstaller一样，是一个大工程了。这种方式的话也有很大的弊端。就是pyinstaller很容易把用不上的库给打包进去。例如一些并不需要的非py库就给打包进去了。导致的软件也是特别的大。

第一种方法太麻烦了，这时候我就在想第二种方法。既然我没办法去分析每个文件import的哪些包。但是我的软件，标准库，第三方库，等等都在一个文件夹内呀。我可以启动软件，然后观察软件在启动过程中动用了哪些文件。然后我把那些被读取了的文件复制到新的文件夹中去。不就可以将我们所需要的程序打包出来嘛。在网上搜索一番之后，我找到了我所需的软件：Process Monitor(系统进程监视器)。我们只要设置它对我们的工作目录进行监控。然后将软件打开。就能获得到软件运行时候所需的文件了。

操作流程

下面介绍一下具体的工作流程。

首先打开软件，设置两个过滤器，一个过滤器匹配路径，一个路径匹配进程名称。

过滤器实际上是最后一次你退出的 Process Monitor:

显示匹配这些条件的条目:

路径 包含 C:\pypy3.6-v7.3.1-win32 然后 包括

重置 添加 删除

列	关系	值	动作
<input checked="" type="checkbox"/>  进程名称	包含	pypy3.exe	包括
<input checked="" type="checkbox"/>  路径	包含	C:\pypy3.6-v7.3.1-...	包括
<input checked="" type="checkbox"/>  进程名称	是	Procmon.exe	排除
<input checked="" type="checkbox"/>  进程名称	是	Procexp.exe	排除

<https://blog.csdn.net/zengcz123>

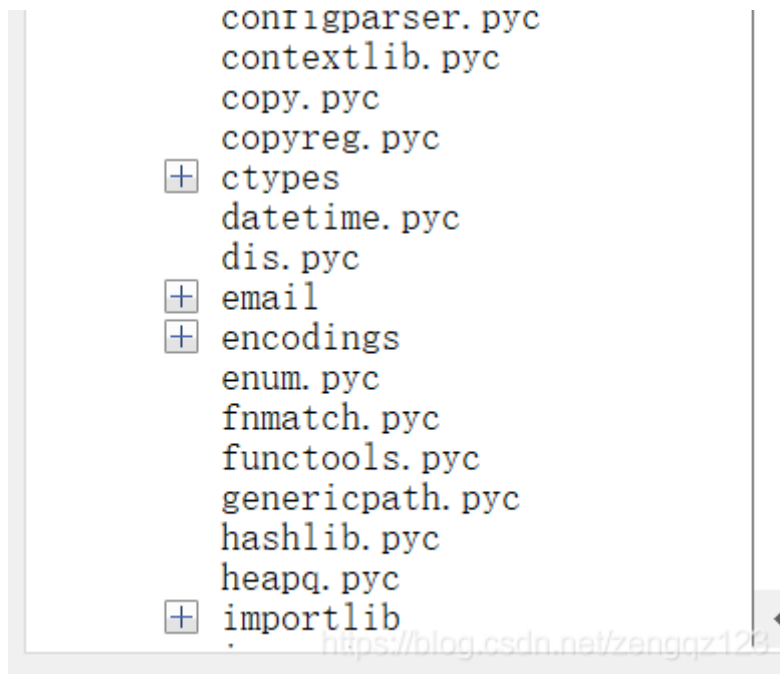
在cmd中启动软件，然后就能观察到许多文件被调用了，这时候点击菜单栏的工具->文件摘要。就能看到文件的使用情况了。

跟踪期间访问的文件:

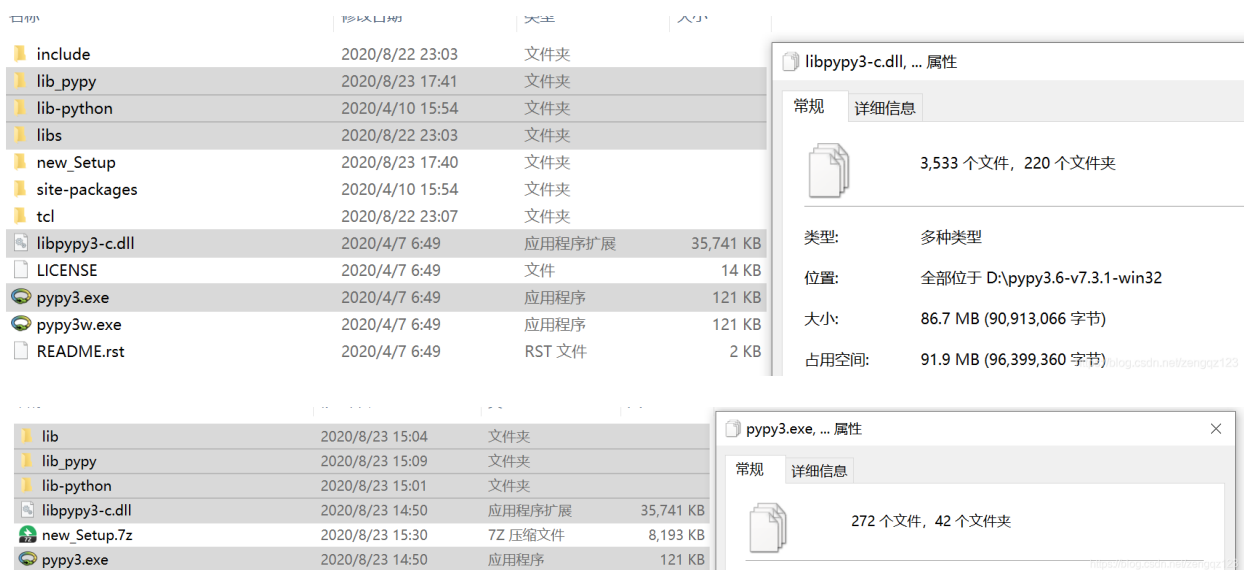
通过路径 通过文件夹 通过扩展

名称

☐ C:
 ☐ pypy3.6-v7.3.1-win32
 .Tk.py
 .Tk.tcl
 .blf.py
 .blf.tcl
 CRYPTBASE.dll
 CRYPTSP.dll
 IPHLPAPI.DLL
 ☒ Modules
 Mswsock.DLL
 TextShaping.dll
 VCRUNTIME140.dll
 lib
 ☐ lib-python
 ☐ 3
 *
 __future__.pyc
 _collections_abc.pyc
 _compat_pickle.pyc
 _compression.pyc
 _sitebuiltins.pyc
 _weakrefset.pyc
 abc.pyc
 ast.pyc
 ☒ asyncio
 base64.pyc
 bisect.pyc
 bz2.pyc
 calendar.pyc
 codecs.pyc
 ☒ collections
 ☒ concurrent



接下来就是将其保存为CSV格式。这时候再写一个脚本，把这些文件复制到新的目录下。然后按照之前的方式打包一下。打包前有3533个文件，而通过这种方式打包为272个文件。解压速度从原来的20秒提升到3秒左右。很大程度的提高了解压速度。同时软件缩小为8mb。甚至比pyinstaller打包出来的软件（10mb）还小。



代码提供:

CSV文件拷贝的源码提供:

```
1 import csv
2 import os
3 from shutil import copy
4
5
6 #打开CSV文件
```

```
7 with open(r"你的CSV文件.CSV",'r',encoding='gb18030',errors='ignore') as f:
8     reader = csv.reader(f)
9     print(type(reader))
10    for row in reader:
11        #打印出每行的第一个参数，并判断是否为文件
12        print(row[0]+"是否为文件: "+str(os.path.isfile(row[0])))
13        #当判断是文件的时候
14        if os.path.isfile(row[0]):
15            #获取文件的路径，并将文件保持到该目录的new下
16            parent = os.path.dirname(row[0])
17            newpath = parent.replace(r"C:\pypy3.6-v7.3.1-win32",r"C:\pypy3.6-
18            #判断文件的路径是否存在，如果不存在，则创建新的路径
19            if not os.path.isdir(newpath):
20                os.makedirs(newpath)
21            #将文件复制过去
22            copy(row[0],row[0].replace(r"C:\pypy3.6-v7.3.1-win32",r"C:\pypy3.
23
```

来源: <https://blog.csdn.net/zengqz123/article/details/108188673>