Python中系统命令

1：os.system

例如：查看cpu信息

>>> import os

>>> os.system("cat /proc/cpuinfo")

2：os.popen

该方法不但执行命令还返回执行后的信息对象，是通过一个管道文件将结果返回

>>> output=os.popen("cat /proc/cpuinfo")

>>> print(output.read())

3：commands模块（返回状态模块）

Commands.getoutput(cmd)：返回命令的执行结果，没有返回值

**commands.getstatusoutput(cmd)  返回一个元组（status，output）**

**status代表的shell命令的返回态，如果成功的话是0；output是shell的返回的结果**

>>> import commands

>>> commands.getstatusoutput("ls /bin/ls")

(0, '/bin/ls')===》（这里0表示命令的返回状态，后面返回命令的执行结果）

4：模块subprocess

Subprocess.call: 执行命令，并返回执行状态，其中shell参数为False时，命令需要通过列表的方式传入，当shell为True时，可直接传入命令

subprocess.check\_call()：用法与subprocess.call()类似，区别是，当返回值不为0时，直接抛出异常

在当前目录下创建一个subprocesstest文件夹

a=subprocess.Popen("mkdir subprocesstest",shell=True,cwd='.')

## 5：shutil

（1）：将文件内容考贝到另一个文件中（**shutil.copyfileobj(fsrc, fdst[, length])**）

>>> import shutil

>>> shutil.copyfileobj(open('old.xml','r'),open('new.xml','w'))

这里把old.xml中的内容考贝到new.xml中

（2）： shutil.move(source,destination):将source移动到destination中，如果destination 指向一个文件夹，那么 source 文件将被移动到 destination 中，并且保持其原有名字，如果 destination 指向一个文件，那么 source 文件将被移动并重命名

（3）：**shutil.copyfile(src, dst)考贝文件**

**>>> import shutil**

**>>> shutil.copyfile('f1.log','f2.log')**

**（4）：shutil.copymode(src, dst)：**仅拷贝权限。内容、组、用户均不变

（5）：**shutil.copystat(src, dst)：**仅拷贝状态的信息，包括：mode bits, atime, mtime, flags  
（6）：**shutil.copy(src, dst)：**拷贝文件和权限  
**（7）：shutil.copy2(src, dst)：**拷贝文件和状态信息

**（8）：shutil.copytree(src, dst, symlinks=False, ignore=None)：**递归的去拷贝文件夹

**Src:原文件夹**

**Dst:目录文件夹**

**Symlinks:** 如果 ` symlinks ` 为真， 源文件夹中的符号链接将会被保留，但是原链接的元数据**不会**被复制。如果值为假或被省略，则链接文件指向文件的内容和元数据复制到新文件夹树中

**例如：**

**>>> import shutil**

**>>> shutil.copytree("f1","f2")**

**上面的例子中，将f1文件考贝到f2文件包括子文件，其实相当于复制一份f1命名为f2,目标文件夹应该不存在**

**(9):** **shutil.rmtree(path[, ignore\_errors[, onerror]]):递归的去删除文件**

**删除整个文件夹，包括子文件夹**

**（10）：shutil.make\_archive(base\_name, format,...)：**创建压缩包并返回文件路径，例如：zip、tar

**base\_name:要压缩成的文件名称**

**format:压缩成的格式（zip,tar,tar.gz,rar）**

**root\_dir:要压缩的文件夹路径**

**案例：**

**>>> import shutil**

**>>> ret=shutil.make\_archive("abc",'zip',root\_dir="/home/gaoyuxia/Test")**

**将/home/gaoyuxia/Test目录下的所有文件压缩成abc.zip**

**将文件压缩后放在指定路径：**

**>>> ret=shutil.make\_archive("/home/gaoyuxia/abc",'zip',root\_dir="/home/gaoyuxia/Test")**

**6:ZipFile**

f=zipfile.ZipFile(file, mode="r", compression=ZIP\_STORED, allowZip64=False)

压缩：

f=zipfile.ZipFile(file, mode="r", compression=ZIP\_STORED, allowZip64=False)

**>>> import zipfile**

**>>> with zipfile.ZipFile("test.zip",mode="w") as zipf:**

**... zipf.write("a.txt")**

**... zipf.write("b.txt")**

**...**

**>>> zipf=zipfile.ZipFile("test.zip")**

**>>> print zipf.namelist()**

**['a.txt', 'b.txt']**

**解压：**

f.extract(*directory*)和f.exractall(*directory*)

**>>> import zipfile**

**>>> zipf=zipfile.ZipFile("test.zip")**

**>>> zipf.extractall("test/")**

**将test.zip解压到test目录下**

1:zipfile.is\_zipfile(filename)   
判断一个文件是不是压缩文件 (如果是压缩文件返回True,如果不是压缩文件返回False)

2:ZipFile.namelist() :返回压缩包中的文件列表

（1）：先得到ZipFile

f=zipfile.ZipFile(“test.zip”)

(2)：得到压缩包中的文件

f.namelist()

3 ZipFile.open(name[, mode[, password]]) ：打开压缩文档中的某个文件

>>> f=zipfile.ZipFile("test.zip")

>>> f.open("a.txt").read() #读取文件的内容

**7:TarFile**

**压缩：（压缩一个文件夹下的所有文件）**

**>>> import os**

**>>> import tarfile**

**>>> \_\_author\_\_="Administrator"**

**>>> def main():**

**... cwd=os.getcwd()**

**... tar=tarfile.open("test.tar","w:gz")**

**... for root,dir,files in os.walk(cwd):**

**... for file in files:**

**... fullpath=os.path.join(root,file)**

**... tar.add(fullpath)**

**...**

**>>> if \_\_name\_\_=='\_\_main\_\_':**

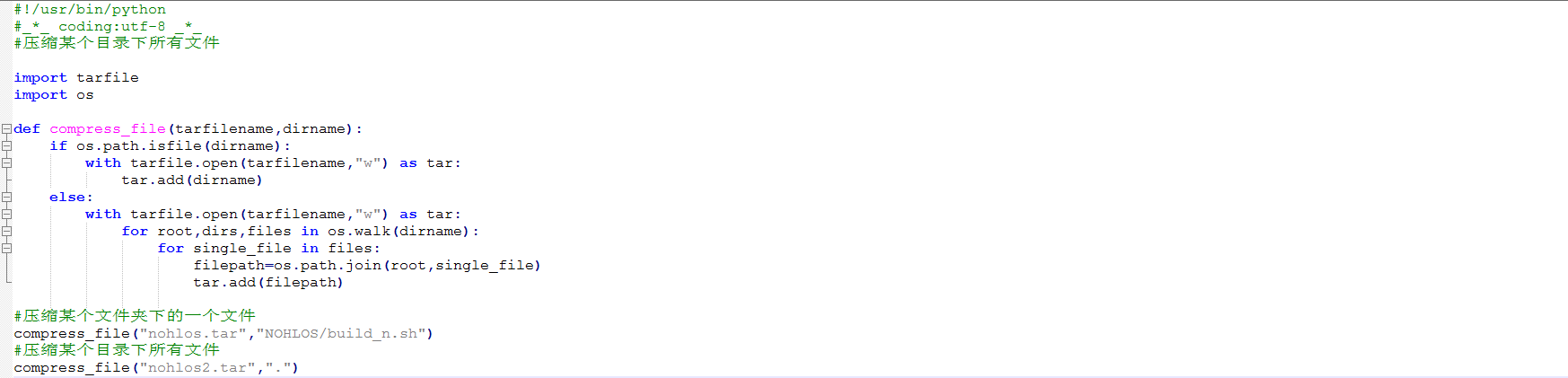
**... main()**

**解压：**

**>>> with tarfile.open("a.tar","r") as tar:**

**... print(tar.getmembers()) #查看压缩包内文件成员**

**... tar.extractall() #解压全部文件**



8：time模块

#------将结构化时间转成字符串时间strftime

print(time.strftime("%Y---%m-%d %X",time.localtime()))

#------将字符串时间转成结构化时间strptime

print(time.strptime("2016:12:24:17:50:36","%Y:%m:%d:%X"))