## Python调用系统命令的6种方法

在Python中调用系统命令一般使用os或者subprocess模块，下面介绍Python中最常用的6种调用系统命令的方法。

1、os.system()

该函数返回命令执行结果的返回值，system()函数在执行过程中进行了以下三步操作：

1、fork一个子进程；

2、在子进程中调用exec函数去执行命令；

3、在父进程中调用wait（阻塞）去等待子进程结束。

返回0表示命令执行成功，其他表示失败。

用法：

os.system("command")

2、os.popen()

这种调用方式是通过管道的方式来实现，函数返回是 file read 的对象，对其进行读取read、readlines等操作可以看到执行的输出。

用法：

os.popen("command")

3、subprocess.Popen()

subprocess模块被推荐用来替换一些老的模块和函数，如：os.system、os.spawn\*、os.popen\*等

subprocess模块目的是fork一个新的进程并与之通信，最常用是定义类Popen，使用Popen可以创建进程，并与进程进行复杂的交互。

用法：

child = subprocess.Popen(["cmd","arg1"...])

4、subprocess.call()

执行指定的命令，返回命令执行状态，其功能类似于os.system(cmd)。

用法：

subprocess.call("command")

5、subprocess.run()

Python 3.5中新增的函数，执行指定的命令，等待命令执行完成后返回一个包含执行结果的CompletedProcess类的实例。

用法：

subprocess.run(args, \*, stdin=None, input=None, stdout=None, stderr=None, shell=False, timeout=None, check=False, universal\_newlines=False)

6、subprocess.getstatusoutput()

执行cmd命令，返回一个元组（命令执行状态，命令执行结果输出），其功能类似于commands.getstatusoutput()。

用法：

subprocess.getstatusoutput("command")