35-subprocess

April 29, 2016

1 subprocess module

- run external programs
- also known as an 'exec'
- way to run programs in the 'external world'
- often quite useful
- documentation somewhat complex, but easy to use

In [1]:
```
# delete file if it's there

import os
path = '/tmp/subp'
os.remove(path)
```

In [2]:
```
# simplest form - just run a command
# touch will create an empty file if none exits,
# or change the last access date of an existing file
# exit code (0 is happy) is returned

import subprocess

subprocess.call(['touch', path])
```

Out[2]:
```
0
```

In [3]:
```
[os.access(path, os.F_OK), os.stat(path)]
```

Out[3]:
```
[True,
 os.stat_result(st_mode=33206, st_ino=16887087, st_dev=16777219, st_nlink=1, st_uid=501, st_gid=0
```

In [4]:
```
# return the stdout command produces
# can pick up stderr as well

subprocess.check_output(['echo', 'run', 'as', 'a', 'subproc'], universal_newlines=True)
```

Out[4]:
```
'run as a subproc\n'
```

- linux/mac has a command line topological sort

In [5]:
```
# supply stdin, and read stdout
# 3 comes before 8, 3 before 10, ...

pairs = [[3, 8], [3, 10], [5, 11], [7, 8], [7, 11], [8, 9], [11, 2], [11, 9], [11, 10]]
input = ''.join( [ ('%d %d\n' % (l, r)) for l, r in pairs ])
print(input)
out=subprocess.check_output(['tsort'], input=input, universal_newlines=True)
out.split()
```

```
3 8
3 10
5 11
7 8
7 11
8 9
11 2
11 9
11 10
```

Out[5]: ['7', '5', '3', '11', '10', '8', '2', '9']

In [6]:
```
# with universal_newlines false, input/output is binary
# note input is a byte array

subprocess.check_output(["sed", "-e", "s/foo/bar/"],
                        input=b"when in the course of fooman events\n")
```

Out[6]: b'when in the course of barman events\n'

In [7]:
```
# run under a shell - can do pipes, redirects

subprocess.check_output(['tsort|wc'], input=input, shell=True, universal_newlines=True)
```

Out[7]: '   8   8   18\n'

2 os.system

- very simple, not much control

In [8]:
```
# works on a mac

import os
os.system('say macs have a text to speech system built in')
```

Out[8]: 0

In [ ]: