

自分の状況では、この例がもっともフィットしました。
[試験運用中なLinux備忘録](#)

[O'REILLY](#)

[subprocessにおけるエラー報告ダイアログ抑止の参考](#)

py2exe内で外部プロセスを使うプログラムA.exeを作成、
プログラムBからプログラムAを外部プロセスとして呼ぶ場合、
一定量まで標準出力を呼んでからまとめて出す、ようなおかしな状況に。

readlineのバッファ量の設定と思ったのだが
[Python Matrix Jp](#)

プログラムA内でsys.stdout.flush()やsys.stdout.close()をよぶ、といった解決法がありそう。
[subprocess -popen - reading stdout from child - hangs](#)

[西尾泰和のはてなダイアリー\(Pythonで子プロセスの出力を読む\)](#)

[西尾泰和のはてなダイアリー\(子プロセスに標準入力と標準出力を中継\)](#)

子プロセス以下をすべてkillしたい

subprocess.Popenに
preexec_fn= os.setsid
とオプション指定すると、サブプロセスがプロセスグループ (Process Group) のトップになるため
os.killpg(pgid, signal.SIGTERM)
したときに、親プロセスを巻き込まずにkillすることができます。

プロセス情報を操作するのはosモジュールの方にあります。
[プロセスのパラメタ](#)

Pythonではありませんが、Unixのプロセスについて分かりやすく説明されています。
[プロセス制御](#)

spaceのあるパスでうまくいかない

```
path = 'C:/Documents and Settings/user'  
cmd = ['C:/Program Files/TortoiseSVN/bin/SubWCRev.exe', path]  
p = subprocess.Popen(  
    cmd,  
    shell=True,  
    stdin=subprocess.PIPE,  
    stdout=subprocess.PIPE,  
    stderr=subprocess.STDOUT  
)
```

このpathの方にスペースがあると、コマンド自体、C:/Programまでと判断されてしまう？
解決法は、shell=Falseです。