SUBPROCESS:

# out=os.system("di r")

# print(out)  # 0

# out=os.system("dir1")

# print(out)  # 1

Linux machine:

import subprocess

cmd="ls -l"

sp=subprocess.Popen(cmd,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out}')

print(f'Error is: {err}')

print(f"rc: {rc}")

print(f"{type(out)}")

Output:

OUTPUT IS: b'total 16\n-rw-rw-r-- 1 ubuntu ubuntu 2 Jun 30 01:09 1.txt\n-rw-rw-r-- 1 ubuntu ubuntu 311 Jun 30 01:48 sp1.py\n-rw-rw-r-- 1 ubuntu ubuntu 244 Jun 30 03:19 sp11.py\n-rw-rw-r-- 1 ubuntu ubuntu 327 Jun 30 01:54 sp2.py\n'

Error is: b''

rc: 0

<class 'bytes'>

CODE 2:

(ERROR)

import subprocess

cmd="lss -l"

sp=subprocess.Popen(cmd,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out}')

print(f'Error is: {err}')

print(f"rc: {rc}")

print(f"{type(out)}")

OUTPUT:

$ python3 sp11.py

OUTPUT IS: b''

Error is: b'/bin/sh: 1: lss: not found\n'

rc: 127 (Positive Number)

<class 'bytes'>

3.

Shell=True (String)

New shell is going to create

import subprocess

cmd="ls -l"

sp=subprocess.Popen(cmd,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out.splitlines()}')

print(f'Error is: {err.splitlines()}')

print(f"{type(out.splitlines())}")

Out put as string:

$ python3 sp11.py

OUTPUT IS: ['total 16', '-rw-rw-r-- 1 ubuntu ubuntu 2 Jun 30 01:09 1.txt', '-rw-rw-r-- 1 ubuntu ubuntu 311 Jun 30 01:48 sp1.py', '-rw-rw-r-- 1 ubuntu ubuntu 290 Jun 30 03:29 sp11.py', '-rw-rw-r-- 1 ubuntu ubuntu 327 Jun 30 01:54 sp2.py']

Error is: []

<class 'list'>

Code 4:

cmd="ls -lrt"

print(cmd.split())

['ls', '-lrt']

CODE 5:

Shell=false

So has to give list

CODE 5:

import subprocess

#cmd="ls -lrt"

#cmd="ls -l".split()

cmd=["ls","-lrt"]

sp=subprocess.Popen(cmd,shell=False,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out.splitlines()}')

print(f'Error is: {err.splitlines()}')

print(f"{type(out.splitlines())}")

output:

$ sudo python3 sp11.py

OUTPUT IS: ['total 16', '-rw-rw-r-- 1 ubuntu ubuntu 2 Jun 30 01:09 1.txt', '-rw-rw-r-- 1 ub untu ubuntu 311 Jun 30 01:48 sp1.py', '-rw-rw-r-- 1 ubuntu ubuntu 327 Jun 30 01:54 sp2.py', ' -rw-rw-r-- 1 ubuntu ubuntu 334 Jun 30 04:00 sp11.py']

Error is: []

<class 'list'>

CODE 6:

import subprocess

#cmd="ls -lrt"

#cmd="ls -l".split()

#cmd=["ls","-lrt"]

cmd= ["echo","$HOME"]

sp=subprocess.Popen(cmd,shell=False,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out.splitlines()}')

print(f'Error is: {err.splitlines()}')

print(f"{type(out.splitlines())}")

OUTPUT:

OUTPUT IS: ['$HOME']

Error is: []

<class 'list'>

CODE 8:

True for ENV variables

import subprocess

#cmd="ls -lrt"

#cmd="ls -l".split()

#cmd=["ls","-lrt"]

#cmd= ["echo","$HOME"]

cmd = "echo $HOME"

sp=subprocess.Popen(cmd,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out.splitlines()}')

print(f'Error is: {err.splitlines()}')

print(f"{type(out.splitlines())}")

OUTPUT:

$ python3 sp11.py

OUTPUT IS: ['/home/ubuntu']

Error is: []

<class 'list'>

######################################

TRUE: for ENV, if new shell created

FALSE : no new shell created. But not works for ENV

True: only string

False : List

############################################

CONCEPT:

import subprocess

sp=subprocess.Popen(cmd,shell=True/False,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

print(f'OUTPUT IS: {out}')

print(f'Error is: {err}')

==================================>

if shell=True then your cmd is a string (as your os command)

if shell=False then your cmd is a list

Note: shell=False dont work on your os environment variables

ex: cmd="ls -lrt" ==>shell=True

shell=False ==> cmd="ls -lrt".split() or ['ls','-lrt']

=======================================================================

shell=True always on windows

============================

cmd is a string

----------------------------------------------------------

################################################################################################################

Shell=True

Windows:

A computer screen shot of a black screen

AI-generated content may be incorrect.

CLASS 2:

BASH VERSION Of UNIX

Bash –version

$ bash --version

GNU bash, version 5.2.21(1)-release (x86\_64-pc-linux-gnu)

Code1:

import subprocess

#cmd="ls -lrt"

cmd=["bash","--version"]

sp=subprocess.Popen(cmd,shell=False,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

if rc==0:

print("out is :",out)

else:

print("cmd failed")

OUTPUT:

$ python3 bash\_version.py

out is : GNU bash, version 5.2.21(1)-release (x86\_64-pc-linux-gnu)

Copyright (C) 2022 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law

CODE 2:

import subprocess

#cmd="ls -lrt"

cmd=["bash","--version"]

sp=subprocess.Popen(cmd,shell=False,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal\_newlines=True)

rc=sp.wait()

out,err=sp.communicate()

if rc==0:

for each in out.splitlines():

if "version" in each and "release" in each:

print(each)

else:

print("cmd failed",err)

OUTPUT:

$ python3 bash\_version.py

GNU bash, version 5.2.21(1)-release (x86\_64-pc-linux-gnu)

CODE3 :

if rc==0:

for each in out.splitlines():

if "version" in each and "release" in each:

print(each.split())

OUTPUT:

$ python3 bash\_version.py

['GNU', 'bash,', 'version', '5.2.21(1)-release', '(x86\_64-pc-linux-gnu)']

ubuntu@ip-172-31-83-125:~$ vim bash\_version.py

CODE 4:

if "version" in each and "release" in each:

print(each.split()[3])

OUTPUT:

5.2.21(1)-release

CLASS 3:

JAVA\_VERSION:

1. sudo apt update
2. sudo apt install default-jre
3. java -version

$ java --version

openjdk 21.0.7 2025-04-15

OpenJDK Runtime Environment (build 21.0.7+6-Ubuntu-0ubuntu124.04)

OpenJDK 64-Bit Server VM (build 21.0.7+6-Ubuntu-0ubuntu124.04, mixed mode, sharing)

OUTPUT:

$ python3 java\_version.py

21.0.7