原 Python多进程 - 实现多进程的几种方式

2018年01月05日 15:31:27 Money多多 阅读数：10614

方式一: os.fork()

```
# -*- coding:utf-8 -*-

"""
pid=os.fork()
    1.只用在Unix系统中有效，Windows系统中无效
    2.fork函数调用一次，返回两次：在父进程中返回值为子进程id，在子进程中返回值为0
"""

import os

pid=os.fork()

if pid==0:
    print("执行子进程，子进程pid={pid},父进程ppid={ppid}".format(pid=os.getpid(),ppid=os.getppid()))
else:
    print("执行父进程，子进程pid={pid},父进程ppid={ppid}".format(pid=pid,ppid=os.getpid()))
```

方式二: 使用multiprocessing模块: 创建Process的实例，传入任务执行函数作为参数

```
# -*- coding:utf-8 -*-

"""
Process常用属性与方法:
    name:进程名
    pid: 进程id
    run()，自定义子类时覆写
    start()，开启进程
    join(timeout=None)，阻塞进程
    terminate(),终止进程
    is_alive()，判断进程是否存活
"""

import os,time
from multiprocessing import Process

def worker():
    print("子进程执行中>>> pid={0},ppid={1}".format(os.getpid(),os.getppid()))
    time.sleep(2)
    print("子进程终止>>> pid={0}".format(os.getpid()))

def main():
    print("主进程执行中>>> pid={0}".format(os.getpid()))

    ps=[]
    # 创建子进程实例
    for i in range(2):
        p=Process(target=worker,name="worker"+str(i),args=())
        ps.append(p)

    # 开启进程
    for i in range(2):
        ps[i].start()

    # 阻塞进程
    for i in range(2):
        ps[i].join()

    print("主进程终止")

if __name__ == '__main__':
```

```
    main()
```

方式二: 使用multiprocessing模块: 派生Process的子类，重写run方法

```python
# -*- coding:utf-8 -*-

import os,time
from multiprocessing import Process

class MyProcess(Process):
    def __init__(self):
        Process.__init__(self)

    def run(self):
        print("子进程开始>>> pid={0},ppid={1}".format(os.getpid(),os.getppid()))
        time.sleep(2)
        print("子进程终止>>> pid={}".format(os.getpid()))

def main():
    print("主进程开始>>> pid={}".format(os.getpid()))
    myp=MyProcess()
    myp.start()
    # myp.join()
    print("主进程终止")

if __name__ == '__main__':
    main()
```

方式四: 使用进程池Pool

```python
# -*- coding:utf-8 -*-

import os,time
from multiprocessing import Pool

def worker(arg):
    print("子进程开始执行>>> pid={},ppid={},编号{}".format(os.getpid(),os.getppid(),arg))
    time.sleep(0.5)
    print("子进程终止>>> pid={},ppid={},编号{}".format(os.getpid(),os.getppid(),arg))

def main():
    print("主进程开始执行>>> pid={}".format(os.getpid()))
    ps=Pool(5)
    for i in range(10):
        # ps.apply(worker,args=(i,))          # 同步执行
        ps.apply_async(worker,args=(i,))  # 异步执行

    # 关闭进程池，停止接受其它进程
    ps.close()
    # 阻塞进程
    ps.join()
    print("主进程终止")

if __name__ == '__main__':
    main()
```