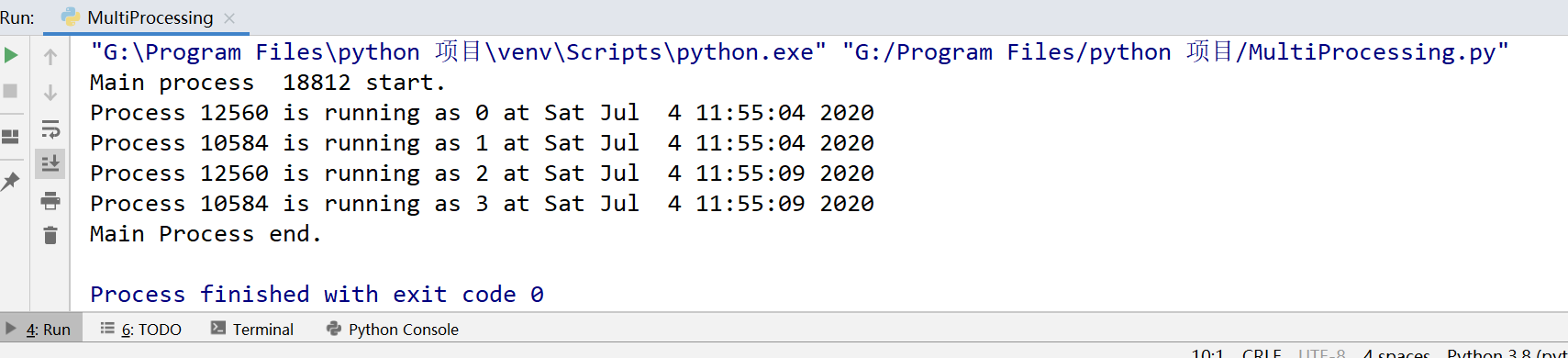
**1、实例化类Process创建一个进程**

**代码：**

**def** text(id):  
 print(**'Child Process'**,id,**'is running.'**)  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 print(**'Main process '**,os.getpid(),**'start.'**)  
 **for** i **in** range(3):  
 p=Process(target=text,args=(i,))  
 p.start()  
 p.join()  
 print(**'Main Process end.'**)

**运行结果：**

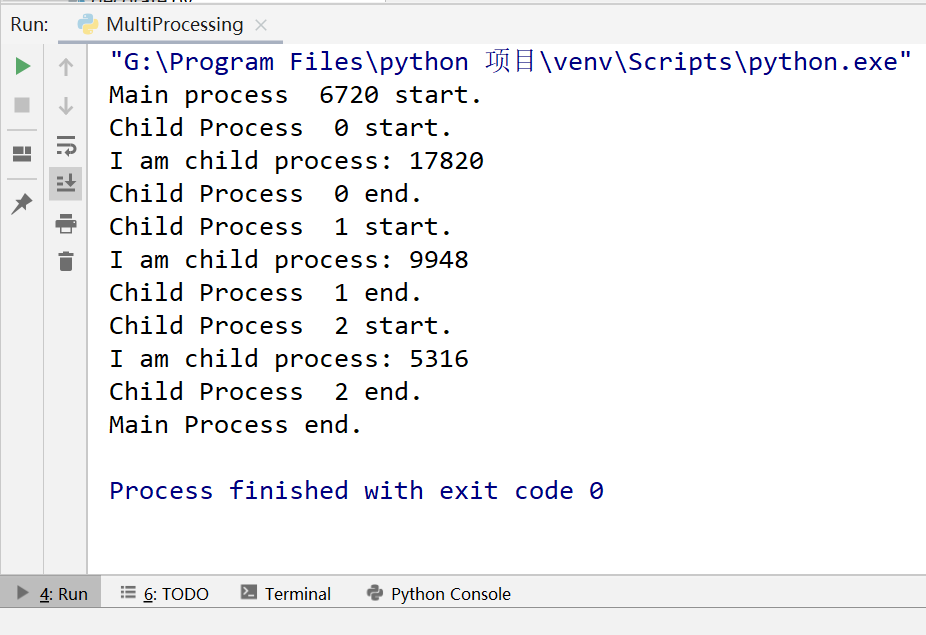


**2、继承Process类创建子进程**

**代码：**

**class** ChildProcess(Process):  
 **def** \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 **def** run(self) :  
 print(**'I am child process:'**,os.getpid())  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 print(**'Main process '**,os.getpid(),**'start.'**)  
 **for** i **in** range(3):  
 print(**'Child Process '**,i,**'start.'**)  
 p=ChildProcess()  
 p.start()  
 p.join()  
 print(**'Child Process '**,i,**'end.'**)  
 print(**'Main Process end.'**)

**运行结果：**

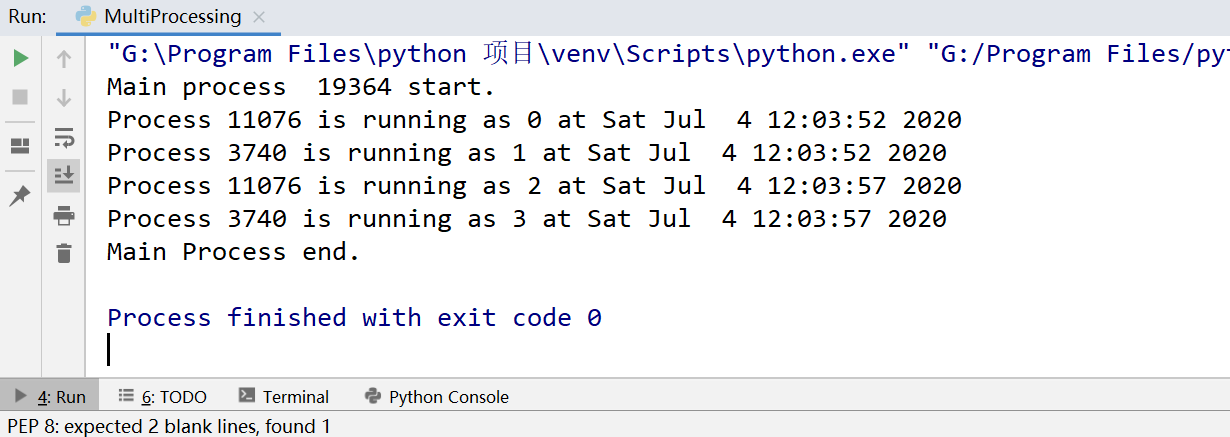
****

**3、进程池函数Pool()用来批量创建子进程**

**代码：**

**def** text2(id):  
 print(**'Process'**,os.getpid(),**'is running as'**,id,**'at'**,ctime())  
 sleep(5) *#停顿五秒***if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 print(**'Main process '**,os.getpid(),**'start.'**)  
 pool=Pool(2)  
 **for** i **in** range(4):  
 pool.apply\_async(text2,args=(i,))  
 pool.close()  
 pool.join()  
 print(**'Main Process end.'**)

**运行结果：**进程池容量为2，所以同时执行两个进程，停顿五秒，再执行同时下面两个进程

****