# 任务描述

对Thread类派生出一个新类MyNewThread, MyNewThread可以每秒调用一次run函数

# 要求

机器在任意时间重启都要能够运行以防止停电等情况。

# 程序流程

|  |
| --- |
| 参见work/src/intodb\_pkl\_20190719/readdata.py |
| #-\*- coding: UTF-8 -\*- *import* threading *import* time   *def* second(*key*:str):  print("It is " + str(*key*),time.time())   *class* PeriodThread(threading.Thread):  *def \_\_init\_\_*(self, *firstrange*:list, *secondrange*:list):  threading.Thread.\_\_init\_\_(self)  currentDate = str(time.localtime().tm\_year) + "-" + str(time.localtime().tm\_mon) + "-" + str(time.localtime().tm\_mday) #> "2019-07-11"  print(currentDate)  self.f\_starttime = time.mktime(time.strptime(currentDate + " " + *firstrange*[0] , "%Y-%m-%d %H:%M:%S")) # > 1563785375.0012558  self.f\_endtime = time.mktime(time.strptime(currentDate + " " + *firstrange*[1] , "%Y-%m-%d %H:%M:%S")) # > 1563785475.0012558  self.s\_starttime = time.mktime(time.strptime(currentDate + " " + *secondrange*[0] , "%Y-%m-%d %H:%M:%S")) # > 1563785575.0012558  self.s\_endtime = time.mktime(time.strptime(currentDate + " " + *secondrange*[1] , "%Y-%m-%d %H:%M:%S")) # > 1563785675.0012558   *def* run(self):  DemoThread.run(self,int(time.time()))   *def* start(self):  print("上班啦")  *if* time.time() < self.f\_starttime:  interval = int(self.f\_starttime) - time.time()  time.sleep(interval)  *if* time.time() >= self.f\_starttime *and* time.time() <= self.f\_endtime:  print("上午开盘")  *while* time.time() < self.f\_endtime:  self.run()  time1 = time.time()  time.sleep(int(time.time())+1 - time.time())  *if* int(time.time()) - int(time1) > 1:  print("miss one")  self.run()  print("上午收盘")  *if* time.time() < self.s\_starttime:  interval = int(self.s\_starttime) - time.time()  time.sleep(interval)   *if* time.time() >= self.s\_starttime *and* time.time() <= self.s\_endtime:  print("下午开盘")  *while* time.time() < self.s\_endtime:  self.run()  time1 = time.time()  time.sleep(int(time.time())+1 - time.time())  *if* int(time.time()) - int(time1) > 1:  print("miss one")  self.run()  print("下午收盘")  print("下班啦") *class* DemoThread(PeriodThread):  *def* run(self,*key*):  second(*key*)   *def* main():  # print(type(time.localtime().tm\_mon),time.localtime().tm\_year,time.localtime().tm\_mday)  t = PeriodThread(["16:45:50","16:48:59"],["16:49:30","16:54:50"])  t.start()  time.sleep(10)  # t.join()  *if* \_\_name\_\_=='\_\_main\_\_':  main() |