Multithreading

30 January 2025 11:02

HW

1. Write a python program
   - menu
1. Show content of given folder
2. Remove a file
3. Remove a folder
4. Rename a file
5. See contents of a particular file
6. Quit

_____

Process = Program in execution

Thread = Path of execution of a program
Path = sequence in which the instructions will run

```
F1(){..
}

F2(){
….
}
……..
…
F30(){
…
}

 main()
 {
        F1();
        F2();
        F22();
        F5();
 }
```

MAIN Thread is the default thread !!

PYTHON also has a main thread ---
py file code sequence is TREATED As PATH of main thread!!!

Why do we more than main thread ?
-------- to execute instructions simultaneously in ROUND ROBIN

_____SINGLE THREAD ----------
TASK - Sequence
job1
job2
job3

Single Thread

CPU	Job1	Job1	Job1	Job1 end	Job2 start	Job2	Job2 end	Job3 start	Job3..

When will job2 start ?When all instructions of job1 are COMPLETED

TASK1 -- Thread1			TASK2 ---Thread2
Job1			Job2
			Job3

Multithreading

CPU	Job1	Job2	Job1	Job2	Job1	Job2 end	Job1	Job3 start	Job1

Thread 1 and Thread 2 are parallel by time slicing

MULTITHREADING --- Job2 NEED not wait for Job1 to complete !!!

```
import threading
import time
def job1():
    x=0
    while x < 10:
        print('job1',threading.current_thread().name)
        x+=1
        time.sleep(3)

def job3(name):
    print("welcome",name)

def job2(message):
    for _ in range(10):
        print(message)
    job3("prachi")


#__main__
print(threading.current_thread().name)

pathofexecution = job1
th1 =threading.Thread(target=job1)
#th1.start() # it is added in ready Q

th2 = threading.Thread(target=job2, kwargs={'message':"good morning"})
#th2.start()
th1.start()
#th1.join()

th2.start()

# for _ in range(30):
#     print(threading.current_thread().name)
```

```
#th1.join()# the code blocks such that it does not proceed till th1 ends

print("GOOD BYE MAIN ENDS")
```

HW --

Write a Python program that has 2 threads along with main thread

1. Calculate factorial --- pass the max-number
   calc factorial of a num - print it and pause for 2sec then print next and next
   you may use generator

2. Write tables to the file (tablerangefrom, tablerangeto , tableupto)
   (2, 30, 5)
   2*1=2
   2*2=4
   2*3=6
   …
   2*5 = 10

   3*1=3
   ….
   3*5=15

   ….
   ….
   30*1=30
   …
   30*5 = 150

   every time one table is added to file print --- table of num is added on the console
   Pause for 2 sec and continue

Main thread will accept the max and tablefrom, tableto and tableupto from user
create and start the 2 threads
Main should print THANK YOU after both threads end