14th Feb Assignment

February 27, 2023

1 Assignment 13

Q1.What is multithreading in python? Why is it used? Name the module used to handle threads in python

Ans.Multithreading refers to concurrently executing multiple threads by rapidly switching the control of the CPU between threads.

Multithreading allows the execution of multiple parts of a program at the same time. These parts are known as threads and are lightweight processes available within the process. So multithreading leads to maximum utilization of the CPU by multitasking.

The module used to handle threads in python is multiprocessing

Q2.Why threading module used? Write the use of the following functions-

Ans.Threading in python is used to run multiple threads (tasks, function calls) at the same time.

1.activeCount()

[2]:
```python
import threading
```

[3]:
```python
def test(id):
    print('This is my test id %d'%id)
thread={threading.Thread(target=test,args=(i,)) for i in [10,1,3,7,7,8,3,6]}
```

[4]:
```python
threading.activeCount()
```

```
/tmp/ipykernel_139/2205834880.py:1: DeprecationWarning: activeCount() is
deprecated, use active_count() instead
  threading.activeCount()
```

[4]:
```
8
```

2.currentThread()

[5]:

```
#it returns the current Thread object active at the moment.
def test(id):
    print('This is my test id %d'%id)
thread={threading.Thread(target=test,args=(i,)) for i in [10,1,3,7,7,8,3,6]}
```

[6]:

```
threading.currentThread()
```

```
/tmp/ipykernel_139/1511990422.py:1: DeprecationWarning: currentThread() is
deprecated, use current_thread() instead
  threading.currentThread()
```

[6]:

```
<_MainThread(MainThread, started 139928167413568)>
```

3.enumerate()

[7]:

```
def test(id):
    print('This is my test id %d'%id)
thread={threading.Thread(target=test,args=(i,)) for i in [10,1]}
```

[8]:

```
threading.enumerate()
```

[8]:

```
[<_MainThread(MainThread, started 139928167413568)>,
 <Thread(IOPub, started daemon 139928096884288)>,
 <Heartbeat(Heartbeat, started daemon 139928021235264)>,
 <Thread(Thread-3 (_watch_pipe_fd), started daemon 139927996057152)>,
 <Thread(Thread-4 (_watch_pipe_fd), started daemon 139927987664448)>,
 <ControlThread(Control, started daemon 139927979271744)>,
 <HistorySavingThread(IPythonHistorySavingThread, started 139927970879040)>,
 <ParentPollerUnix(Thread-2, started daemon 139927551473216)>]
```

Q3. Explain the following functions

1.run()

run() method is an inbuilt method of the Thread class of the threading module in Python. This method is used to represent a thread's activity.

2.start()

start() method is an inbuilt method of the Thread class of the threading module in Python.It is used to start a thread's activity.

3.join()

The . join() method delays a program's flow of execution until the target thread has been completely read.

4.isAlive()

is_alive() method is an inbuilt method of the Thread class of the threading module in Python. It uses a Thread object, and checks whether that thread is alive or not, ie, it is still running or not

Q4.Write a python program to create two threads. Thread one must print the list of squares and thread two must print the list of cubes

[38]:

```
#thread1 for list of squares
def squares(i):
    print(i*i)

thread1=[threading.Thread(target=squares,args=(i,)) for i in range(10)]
```

[39]:

```
for i in thread1:
    i.start()
```

```
0
1
4
9
16
25
36
49
64
81
```

[42]:

```
#thread2 for list of cubes
def cubes(i):
    print(i*i*i)

thread2=[threading.Thread(target=cubes,args=(i,)) for i in range(10)]
```

[43]:

```
for i in thread2:
    i.start()
```

```
0
1
8
27
64
125
216
343
512
729
```

Q5. State advantages and disadvantages of multithreading

Ans. Advantages of multithreading:

- Multithreading in Python streamlines the efficient utilization of resources as the threads share the same memory and data space.
- It also allows the concurrent appearance of multiple tasks and reduces the response time. This improves the performance.

Disadvantages of multithreading:

- Difficulty of writing code
- Difficulty of debugging
- Difficulty of managing concurrency
- Difficulty of testing
- Difficulty of porting existing code

Q6. Explain deadlocks and race conditions.

Ans.Deadlock- It is a condition in which more than one process occur at a same time and one process is holding a resource and wait for resource holding by another process and vice-versa.

Race condition- A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.