Assignment

PW SKILLS

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

Answer.

Multithreading is a threading technique in Python programming to run multiple threads concurrently by rapidly switching between threads with a CPU help (called context switching).

It enables efficient utilization of the resources as the threads share the data space and memory. Multithreading in Python allows the concurrent and parallel occurrence of various tasks. It causes a reduction in time consumption or response time, thereby increasing the performance.

To use multithreading, we need to **import the *threading module*** in Python Program. A start() method is used to initiate the activity of a thread.

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

1. **activeCount()**
2. **currentThread()**
3. **enumerate()**

Answer:

threading module is used for creating, controlling and managing threads in python.

1. **activeCount():** It returns the number of active threads in the current thread's thread group.
2. **currentThread():** It returns the number of thread objects in the caller's thread control.
3. **enumerate():** It is a built-in function in python that allows you to keep track of the number of iterations (loops) in a loop.

Q3. Explain the following functions:

1. **run()**
2. **start()**
3. **join()**
4. **isAlive()**

Answer:

1. **run():** run() method executes any target function belonging to a given thread object that is now active.
2. **start():** start() method is an inbuilt method of the Thread class of the threading module, it is used to start a thread's activity
3. **join():** join() method takes all items in an iterable and joins them into one string.
4. **isAlive():**is_alive() method returns True if the thread is still running and False , otherwise

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.

Answer:

```
import threading

def SquareCube(n):

    for i in range (1,n):

        s=i*i

        c=i*i*i

        print("The Square of the number is %d and the cube of the number is %d" %(s,c))

thred = [threading.Thread(target=SquareCube , args = (i, )) for i in range (1,5)]

for t in thred:

    t.start()
```

Output:

The Square of the number is 1 and the cube of the number is 1

The Square of the number is 1 and the cube of the number is 1

The Square of the number is 4 and the cube of the number is 8

The Square of the number is 1 and the cube of the number is 1

The Square of the number is 4 and the cube of the number is 8

The Square of the number is 9 and the cube of the number is 27

The Square of the number is 1 and the cube of the number is 1

The Square of the number is 4 and the cube of the number is 8

The Square of the number is 9 and the cube of the number is 27

The Square of the number is 16 and the cube of the number is 64

Q5. State advantages and disadvantages of multithreading.

Answer: Advantages of Multithreading:

1. Multithreading in an interactive application may allow a program to continue running even if a part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user.
2. Processes may share resources only through techniques such as- Message Passing and Shared Memory.
3. Allocating memory and resources for process creation is a costly job in terms of time and space. Since, threads share memory with the process it belongs, it is more economical to create and context switch threads.
4. The benefits of multi-programming greatly increase in case of multiprocessor architecture, where threads may be running parallel on multiple processors.

Disadvantages of Multithreading:

a Python process cannot run threads in parallel but it can run them concurrently through context switching during I/O bound operations.

Q6. Explain deadlocks and race conditions.

Answer: A race condition occurs when two threads use the same variable at a given time. Deadlock exists when two threads seek one lock simultaneously. This situation will stop both threads from processing or executing the functions.