# Threading

Wednesday, July 14, 2021 1:30 PM

In computing, a process is an instance of a computer
program that is being executed.

Any process has 3 basic components:

1. An executable program.
2. The associated data needed by the program (variables,
   work space, buffers, etc.)
3. The execution context of the program (State of process)

Thread:
-------
A thread is an entity within a process that can be scheduled for execution
or
A thread is a sequence of instructions within a program that can be
executed independently of other code.

Thread Control Block (TCB) will control all the thread operations

By Default each and every Programming language work on Uni-Threading principle

If u want to do multi threading then we can do it externally

Necessity of achieving Multi threading:
---------------------------------------
1. Whenever there is a wait time in the program then at that
   time CPU will not take any other programs for execution then
   due to idle time of CPU efficiency of program reduces
2. We can reduce idleness of CPU by using Multi Threading

Multi_Threading:
----------------
1. **Multithreading** is defined as the ability of a processor to execute
   multiple threads concurrently.
2. Whenever the First started thread is waiting then at that particular
   Time Second thread will be under execution

*In a simple, single-core CPU, it is achieved using frequent switching between threads. This is termed as* ***context switching****.*
*In context switching, the state of a thread is saved and state of another thread is loaded whenever any interrupt (due to I/O or manually set) takes place. Context switching takes place so frequently that all the threads appear to be running parallely (this is termed as* ***multitasking****).*

Classification of Threads:
-------------------------
Threads are classified into 2 types
    1.Kernel Threads
    2.User_Defined Threads

1.Kernel Threads:
------------------
1. This are the thread which are created by OS inorder to execution purpose
2. OS will not allow user to create Kernel Threads

2.User_Defined Threads:
-----------------------
1. This are the thread which are created by USER inorder to execution purpose

2. we can create Thread by using the treading module of Python
    import threading

3. By using Thread class of threading module  we can create User_Defined Threads
    Syntax for creating User_Defined Threads

    Threadvariablename=threading.Thread(target,args,name)

    target: the function to be executed by thread
    args: the arguments to be passed to the target function

name:name of the created thread

example:
--------
t1=threading.Thread(target=fun1,args=(5,),name='thread1')

Some of the functions of the threading module:
------------------------------------------
Methods of Thread class
----------------------------

1. start() helps u to start the execution of thread

2. join()  helps u to wait untill the termination of thread execution

3. is_alive() -The is_alive() method checks whether a thread is still executing.

4. name() helps u to print the name of the thread

Functions of threading module:
----------------------------------------
1. threading.active_count() -Returns the number of thread objects that are active.
2. threading.enumerate() - Returns a list of all thread objects that are currently active.

```
import time
import threading
def function1(n):
   for i in range(n):
      print('function1 is executing')
      time.sleep(2)


def function2(n):
   for j in range(n):
      print('Executing Function2')
      time.sleep(2)
```

```
t1=time.time()
thread1=threading.Thread(target=function1,args=(5,),name='Ashu')
thread2=threading.Thread(target=function2,args=(5,),name='Nikky')

thread1.start()
thread2.start()

#print(threading.activeCount())
#print(threading.enumerate())
thread1.join()

thread2.join()
print(thread1.is_alive())
#print(threading.main_thread())
#print(thread1.getName())
#print(threading.activeCount())
#print(threading.enumerate())
t2=time.time()
print('time taken is',t2-t1)
```