In [1]:
```
from threading import Thread
from time import sleep
import time


class MyClass(Thread):
    def __init__(self, tName, tTime, tCounter):
        Thread.__init__(self)
        self.tName = tName
        self.tTime = tTime
        self.tCounter = tCounter

    def run(self):
        print("Strating {}".format(self.tName))
        self.myPrint(self.tTime, self.tTime, self.tCounter)
        print("Ending {}".format(self.tName))

    def myPrint(self, tName, tTime, counter):
        while counter:
            print("{} : {}".format(tName, time.ctime()))
            sleep(tTime)
            counter-=1

t1 = MyClass("Thread 1", 1, 5)
t2 = MyClass("Thread 2", 2, 5)

t1.start()
t2.start()

t1.join()
t2.join()

print("Ending main thread")
```

```
Strating Thread 1
1 : Fri Oct 28 13:36:52 2022
Strating Thread 2
2 : Fri Oct 28 13:36:52 2022
1 : Fri Oct 28 13:36:53 2022
2 : Fri Oct 28 13:36:54 2022
1 : Fri Oct 28 13:36:54 2022
1 : Fri Oct 28 13:36:55 2022
2 : Fri Oct 28 13:36:56 2022
1 : Fri Oct 28 13:36:56 2022
Ending Thread 1
2 : Fri Oct 28 13:36:58 2022
2 : Fri Oct 28 13:37:00 2022
Ending Thread 2
Ending main thread
```

In [2]:
```
class MyClass():
    def __init__(self, tName, tTime, tCounter):
        self.tName = tName
        self.tTime = tTime
        self.tCounter = tCounter

    def run(self):
        print("Strating {}".format(self.tName))
        self.myPrint(self.tTime, self.tTime, self.tCounter)
        print("Ending {}".format(self.tName))

    def myPrint(self, tName, tTime, counter):
        while counter:
            print("{} : {}".format(tName, time.ctime()))
            sleep(tTime)
            counter-=1

t1 = MyClass("Thread 1", 1, 5)
t2 = MyClass("Thread 2", 2, 5)

t1.run()
t2.run()

print("Ending main thread")
```

```
Strating Thread 1
1 : Fri Oct 28 13:37:02 2022
1 : Fri Oct 28 13:37:03 2022
1 : Fri Oct 28 13:37:04 2022
1 : Fri Oct 28 13:37:05 2022
1 : Fri Oct 28 13:37:07 2022
Ending Thread 1
Strating Thread 2
2 : Fri Oct 28 13:37:08 2022
2 : Fri Oct 28 13:37:10 2022
2 : Fri Oct 28 13:37:12 2022
2 : Fri Oct 28 13:37:14 2022
2 : Fri Oct 28 13:37:16 2022
Ending Thread 2
Ending main thread
```

In [3]:
```
# Python program to illustrate the concept of threading, importing the threading module

import threading

def print_cube(num):
        """
        function to print cube of given num
        """
        print("Cube: {}\n".format(num * num * num))

def print_square(num):
        """
        function to print square of given num
        """
        print("Square: {}\n".format(num * num))

if __name__ == "__main__":
        # creating thread
        t1 = threading.Thread(target=print_square, args=(10,))
        t2 = threading.Thread(target=print_cube, args=(10,))

        # starting thread 1
        t1.start()
        # starting thread 2
        t2.start()

        # wait until thread 1 is completely executed
        t1.join()
        # wait until thread 2 is completely executed
        t2.join()

        # both threads completely executed
        print("Done!")
```

```
Square: 100

Cube: 1000

Done!
```

In [4]:
```
# Python program to illustrate the concept of threading
import threading
import os

def task1():
        print("Task 1 assigned to thread: {}".format(threading.current_thread().name))
        print("ID of process running task 1: {}".format(os.getpid()))

def task2():
        print("Task 2 assigned to thread: {}".format(threading.current_thread().name))
        print("ID of process running task 2: {}".format(os.getpid()))

if __name__ == "__main__":

        # print ID of current process
        print("ID of process running main program: {}".format(os.getpid()))

        # print name of main thread
        print("Main thread name: {}".format(threading.current_thread().name))

        # creating threads
        t1 = threading.Thread(target=task1, name='t1')
        t2 = threading.Thread(target=task2, name='t2')

        # starting threads
        t1.start()
        t2.start()

        # wait until all threads finish
        t1.join()
        t2.join()
```

```
ID of process running main program: 9104
Main thread name: MainThread
Task 1 assigned to thread: t1
ID of process running task 1: 9104
Task 2 assigned to thread: t2
ID of process running task 2: 9104
```

In [ ]: