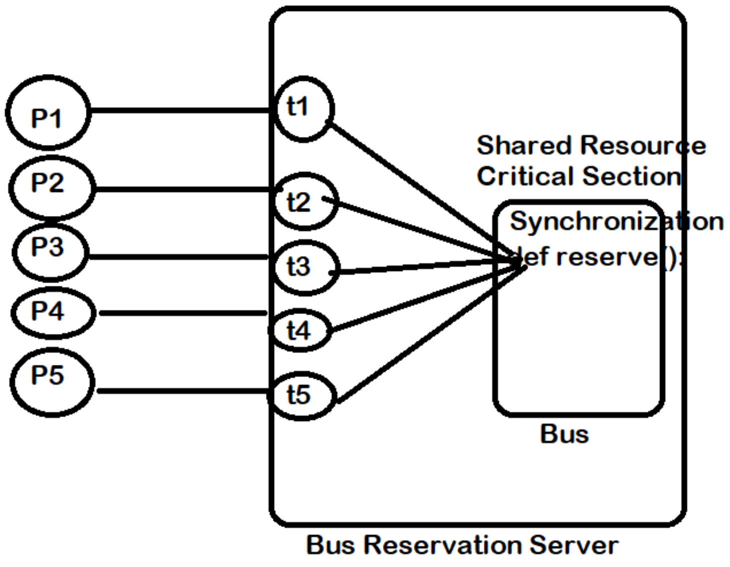
Thread Synchronization

Thread synchronization is defined as a mechanism which ensures that two or more concurrent threads do not simultaneously execute some particular program segment known as critical section.

Critical section refers to the parts of the program where the shared resource is accessed.



Synchronization is a process of acquiring the lock of the object or lock critical section of program.
Synchronization allows developing thread safe applications.

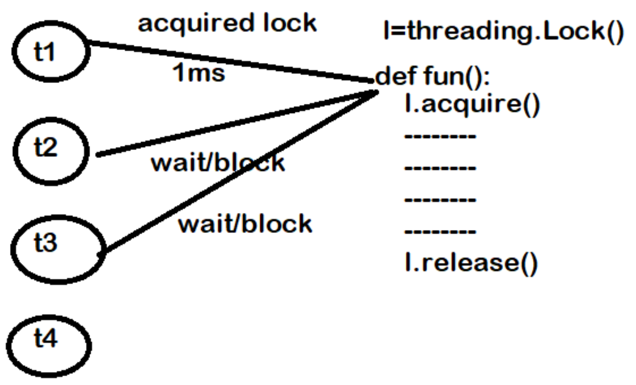
Lock object

class threading.Lock

The class implementing primitive lock objects. Once a thread has acquired a lock, subsequent attempts to acquire it block, until it is released; any thread may release it.

acquire(blocking=True, timeout=-1)
Acquire a lock, blocking or non-blocking.

release()
Release a lock. This can be called from any thread, not only the thread which has acquired the lock.



Example:

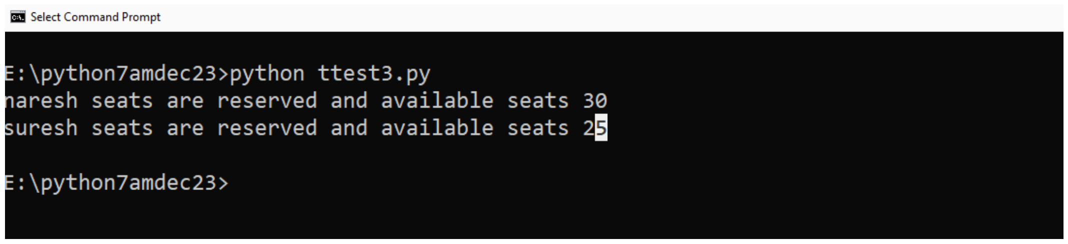
```
import threading

class Bus:
    def __init__(self):
        self.__seats=40
        self.__l=threading.Lock()
    def reserve(self,name,s):
        self.__l.acquire()
        for i in range(s):
            self.__seats=self.__seats-1
        print(f'{name} seats are reserved and available seats
{self.__seats}')
        self.__l.release()

class ReserveThread(threading.Thread):
    def __init__(self,b,name,s):
        super().__init__()
        self.__b=b
        self.__name=name
        self.__s=s
    def run(self):
        self.__b.reserve(self.__name,self.__s)


# Reservation Server
bus1=Bus()
t1=ReserveThread(bus1,"naresh",10)
t2=ReserveThread(bus1,"suresh",5)
t1.start()
t2.start()
```

Output



```
E:\python7amdec23>python ttest3.py
naresh seats are reserved and available seats 30
suresh seats are reserved and available seats 25

E:\python7amdec23>
```

Garbage Collection

In python memory management is automatic.

Memory management consists of two things

1. Allocation of Memory
2. De-Allocation of Memory

Allocating memory is process of creating objects.
De-Allocating memory is process of deleting objects.

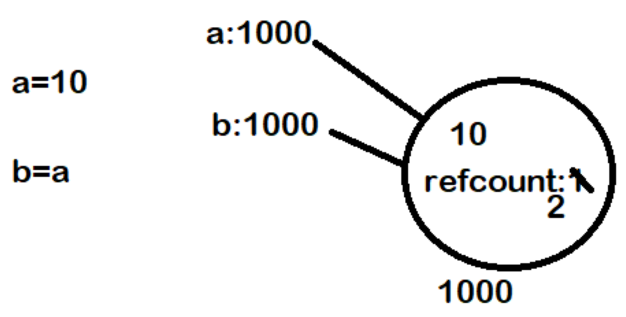
Memory allocating is done by programmer but memory de-allocation done by python virtual machine service called garbage collector.

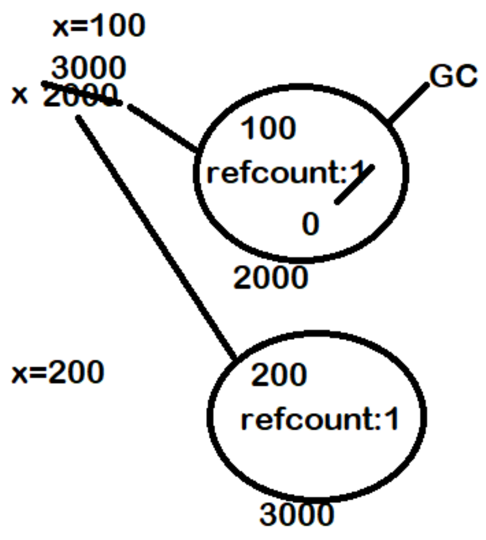
The objects can be deleted in two ways

1. Using del keyword
2. Using garbage collection

Garbage collection is a process of identifying unreferenced objects and removing or deleting.

What is unreferenced object?
An object which is not bind with any reference variable is called unreferenced object (OR) an object whose reference count is zero is called unreferenced object or unused object.





How to find reference count of object?
"sys" module provides a function called getrefcount(), which returns reference count of object.

Example:

```
import sys
class Employee:
    def __init__(self):
        print("Employee object is created...")




emp1=Employee()
emp2=emp1
c=sys.getrefcount(emp1)
print(c)
emp1=None
c=sys.getrefcount(emp2)
print(c)
```

Output

```
Employee object is created...
3
2
```