Hardware and OS Support

The following code snippet is an example of indeterminate program
Code:

```
COUNT = 0

def calculate_count(arg):
  print("{}:begin".format(arg))
  global COUNT
  for _ in range(1000000):
      COUNT = COUNT + 1
  print("{}:ends".format(arg))
```

To solve this problem, we have used locks. The following snippet demonstrates the usage of locks.
Code:

```
COUNT = 0

def calculate_count(arg):
  LOCK.acquire()
  print("{}:begin".format(arg))
  global COUNT
  for _ in range(1000000):
      COUNT = COUNT + 1
  print("{}:ends".format(arg))
  LOCK.release()
```

We can see that the difference between two snippets is usage of functions: LOCK.acquire() and LOCK.release(). It is established that for concurrent execution for more than two threads without race condition, we need hardware support. Explore and find out the support given by hardware and OS and what happens behind the scenes in the above mentioned functions.