|  |
| --- |
| (d) TO SIMULATE DEADLOCK SITUATION \*/ |
|  |  |
|  | #include<iostream> |
|  | #include<thread> |
|  | #include<mutex> |
|  | using namespace std; |
|  | std::mutex m1; |
|  | std::mutex m2; |
|  | std::mutex m3; |
|  | void thread1() { |
|  | m1.lock(); |
|  | m2.lock(); |
|  | m3.lock(); |
|  | cout<<"Critical section of Thread Thread One\n"; |
|  | m1.unlock(); |
|  | m2.unlock(); |
|  | m3.unlock(); |
|  | } |
|  | void thread2() { |
|  | m2.lock(); |
|  | m1.lock(); |
|  | m3.lock(); |
|  | cout<<"Critical section of Thread Thread Two\n"; |
|  | m2.unlock(); |
|  | m1.unlock(); |
|  | m3.unlock(); |
|  | } |
|  | void thread3() { |
|  | m3.lock(); |
|  | m1.lock(); |
|  | m2.lock(); |
|  | cout<<"Critical section of Thread Thread Three\n"; |
|  | m3.unlock(); |
|  | m1.unlock(); |
|  | m2.unlock(); |
|  | } |
|  | int main() |
|  | { |
|  | thread t1(thread1); |
|  | thread t2(thread2); |
|  | thread t3(thread3); |
|  | t1.join(); |
|  | t2.join(); |
|  | t3.join(); |
|  | return 0; |
|  | } |