

## 锁

同步锁，线程锁，互斥锁，共享锁，排他锁

- **synchronized** 和 **Lock** 互斥锁

## 公平锁

公平锁，非公平锁

- 公平锁：先进先出
- 非公平锁：先进后出

乐观锁 **CAS** 例

`java.util.concurrent` 包里有类 CAS 乐观锁

CAS 乐观锁实现

- 乐观锁V版
- 乐观锁A版
- 乐观锁B版

## 死锁

- 死锁V版
- 死锁A版

**ABA**死锁版A版B版B版A版死锁

- 死锁B版

## ReentrantLock

可重入锁，线程锁，互斥锁，共享锁，排他锁

- 可重入锁
- 线程锁

## 示例

```
Lock l = new ReentrantLock(true);
```

## 公平锁

公平锁，线程锁，互斥锁，共享锁，排他锁

- 公平锁
- 线程锁

## 示例

```
Lock l = new ReentrantLock(false);
```

## 线程池

- synchronized  ReentrantReadWriteLock.WriteLock  Lock

A horizontal row of seven empty rectangular boxes, likely intended for a child to draw or write in.

A horizontal row of fifteen empty rectangular boxes, intended for children to write their names in, likely as part of a classroom activity.

- `ReentrantReadWriteLock`
  - `ReentrantReadWriteLock.ReadLock` 读锁

\_\_\_\_\_

- ReentrantLock  $\sqsubset$  synchronized 互斥锁
  - 顺序一致性

## Synchronize

- 
  - 

11

A decorative horizontal bar consisting of a series of small, evenly spaced rectangular boxes, likely a separator or a decorative element at the bottom of the page.

- CAS

三

A horizontal row of 20 small, empty rectangular boxes arranged in a single row.

- CAS
  - CAS
  - - CAS
    - JVM
  - CAS

5

- JVM 通过 CAS 比较并置換 Mark Word
  - JVM 通过 CAS 比较并置換 Mark Word 时，如果发现 ID 不同，则直接忽略
  - 通过 CAS 比较并置換 Mark Word 时，如果发现 ID 不同，则直接忽略

4

-  CAS

- ဗိုလ်ချုပ်အတွက်
    - မြန်မာစာတမ်း
  - ဗိုလ်ချုပ်အတွက်**CPU**၏
    - မြန်မာစာတမ်း
    - မြန်မာစာတမ်း
    - မြန်မာစာတမ်း

5 / 5

- JVM上锁操作通过调用 **Lock Record** 来完成copy操作，Mark Word 为0x00
  - JVM通过CAS操作完成\*\* Mark Word 为0x00 Lock Record 为0x00 Lock Record 为 owner 为0x00的  
Mark Word \*\*
  - 通过CAS操作完成上锁
  - 释放锁时JVM会更新 Mark Word 为0x01
    - 通过CAS操作完成0x00 Lock Record 为0x00 Lock Record 为 owner 为0x00的
    - 通过CAS操作完成0x01 Lock Record 为0x00 Lock Record 为 owner 为0x00的

4

## Mutex

- **ObjectMonitor**
  - **C++ 用の ObjectMonitor 実装**
    - **ObjectMonitor.h**
    - **ObjectMonitor.cpp**
      - **ObjectMonitor::ObjectMonitor**
      - **ObjectMonitor::lock**
      - **ObjectMonitor::unlock**
      - **ObjectMonitor::tryLock**
      - **ObjectMonitor::wait**
      - **ObjectMonitor::notify**
      - **ObjectMonitor::notifyAll**

4



## ObjectMonitor

1. `如果当前线程的 _recursions==0 则直接返回`
  2. `将当前线程加入 _EntryList 队列中`
- `如果当前线程的 _WaitSet 不为空`
    1. `从 _WaitSet 中移除该线程`
    2. `调用 notify() 方法将该线程加入 _EntryList 队列中`