Threadlocal类研究

2019-08-28 16:52:18 0 0 0

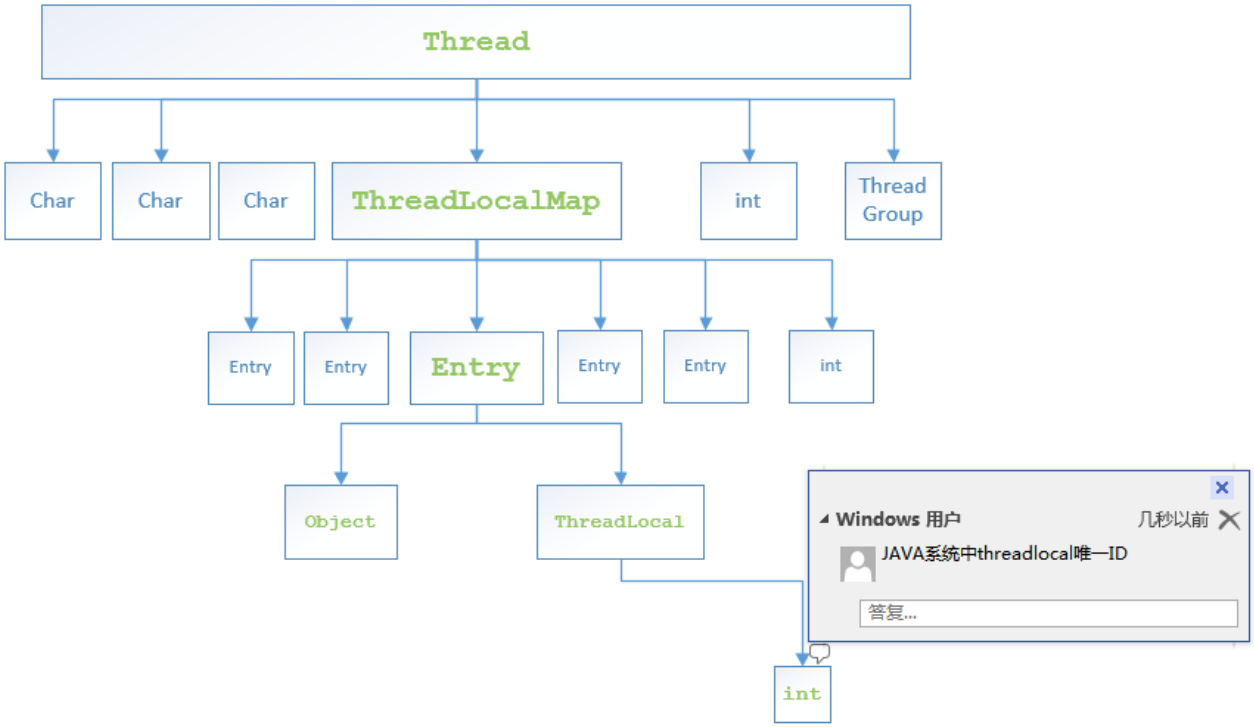
为什么ThreadLocal可以避免数据共享？

```
1.      /**
2.       * Creates a thread local variable.
3.       */
4.      public ThreadLocal() {
5.      }
6.
7.      /**
8.       * Returns the value in the current thread's copy of this
9.       * thread-local variable.  If the variable has no value for the
10.      * current thread, it is first initialized to the value returned
11.      * by an invocation of the {@link #initialValue} method.
12.      *
13.      * @return the current thread's value of this thread-local
14.      */
15.     public T get() {
16.         Thread t = Thread.currentThread();
17.         ThreadLocalMap map = getMap(t);
18.         if (map != null) {
19.             ThreadLocalMap.Entry e = map.getEntry(this);
20.             if (e != null)
21.                 return (T)e.value;
22.         }
23.         return setInitialValue();
24.     }
```

Thread t = Thread.currentThread();

该方法是native方法，底层用c语言实现的，可以获取当前线程的唯一ID。线程运行时好比线程问自己“我是谁？”

线程内部数据结构



一个线程保存的threadlocal越多，线程类占用内存越大。

```
1.  public class ThreadLocalThread implements Runnable{
2.      private final List<ThreadLocal> threadLocals
3.      = Collections.synchronizedList(new ArrayList<ThreadLocal>());  //threadLocal
    s个数越多、Thread越吃内存
4.
5.      @Override
6.      public void run() {
7.
8.      }
9.      public static void main(String[] args) {
10.         //5个线程都非常吃内存
11.         for(int i = 0;i<5;i++){
12.             new Thread(new ThreadLocalThread()).start();
13.         }
14.
15.     }
16.
17. }
```

或者这种

```java
/**
 *
 *线程会创建大量的ThreadLocal。该线程大小会不断升高.
 */
public class ThreadLocaltest {

    private ThreadLocal<String> threadlocal = new ThreadLocal<>();

    public void methed(){
        threadlocal.set(" ");
    }
    public static void main(String[] args) {

        new Thread(new Runnable() {

            @Override
            public void run() {

                while(true){
                    //不会结束、会创建大量的 ThreadLocal
                    new ThreadLocaltest().methed();//每次循环都要创建一个ThreadLocal
                }
            }
        });
    }
}
```

如果系统共有16个Threadlocal，某线程entry数据长度为8.(threadlocal在线程中对应的数组下标与线程中entry数组长度有关)

threadlocal编号	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
数组下标	3	2	1	0	7	6	5	4	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**

编号为1、9、10、11、5、13、6、14的threadlocal依次做set操作。
存放顺序为