```
1   private class Node {
2     AtomicInteger count;
3     Node parent;
4     volatile boolean sense;
5     int d;
6     // construct root node
7     public Node() {
8       sense = false;
9       parent = null;
10      count = new AtomicInteger(radix);
11      ThreadLocal<Boolean> threadSense;
12      threadSense = new ThreadLocal<Boolean>() {
13        protected Boolean initialValue() { return true; };
14    };
15    }
16    public Node(Node myParent) {
17      this();
18      parent = myParent;
19    }
20    public void await() {
21      boolean mySense = threadSense.get();
22      int position = count.getAndDecrement();
23      if (position == 1) {  // I'm last
24        if (parent != null) { // root?
25          parent.await();
26        }
27        count.set(radix);   // reset counter
28        sense = mySense;
29      } else {
30        while (sense != mySense) {};
31      }
32      threadSense.set(!mySense);
33    }
34  }
```

FIGURE 18.17 Thread-local tree barrier.