

3. Test-and-Set

Hardware test-and-set instruction (atomic exchange):

```
int TestAndSet(int *ptr, int new);
```

returns the old value and simultaneously updates to the new value.

```
1  typedef struct __lock_t {
2      int flag;
3  } lock_t;
4
5  void init(lock_t *lock) {
6      // 0: lock is available, 1: lock is held
7      lock->flag = 0;
8  }
9
10 void lock(lock_t *lock) {
11     while (TestAndSet(&lock->flag, 1) == 1)
12         ; // spin-wait (do nothing)
13 }
14
15 void unlock(lock_t *lock) {
16     lock->flag = 0;
17 }
```

Figure 28.3: A Simple Spin Lock Using Test-and-set

Correct: yes; Performance: bad (spinning); Fairness: no

Fix performance --> yield; Fix fairness --> queue