No Busy-wait.

no busy-wait

```
Timestep: 0
```

T1

// (1)

// (2)

// (3)

// (4)

// (5)

// (6)

// (7)

T2

```
free free
```

shared resource

void acquire(int* lock) {

if (*lock == BUSY) {

go_to_sleep();

guard = FREE;

*lock = BUSY;

guard = FREE;

} else {

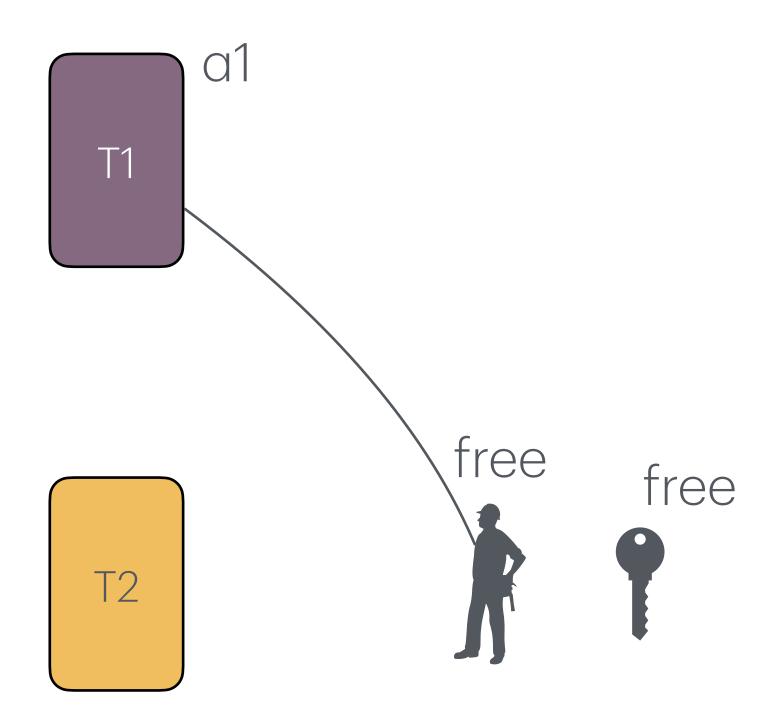
while (test&set(guard));

put_thread_on_wait_queue();

ТЗ

```
Timestep: 1
```

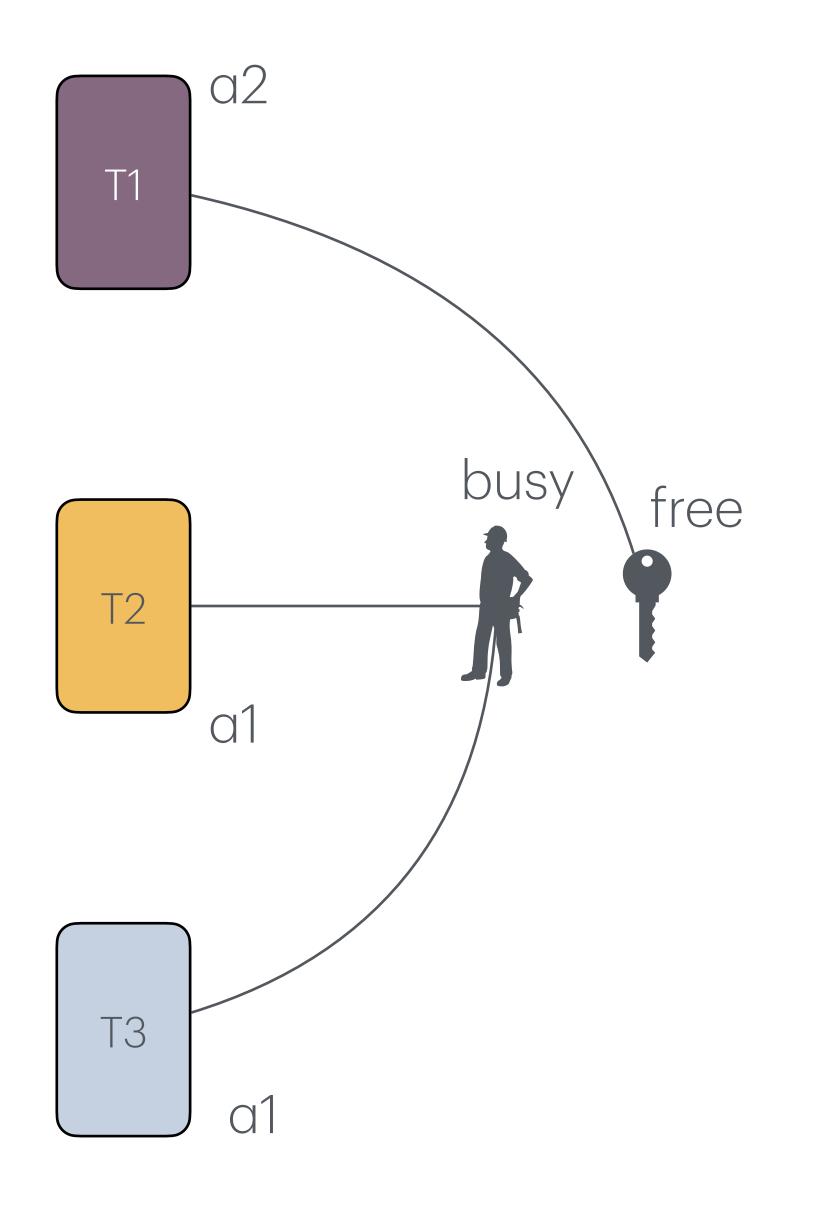
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



ТЗ

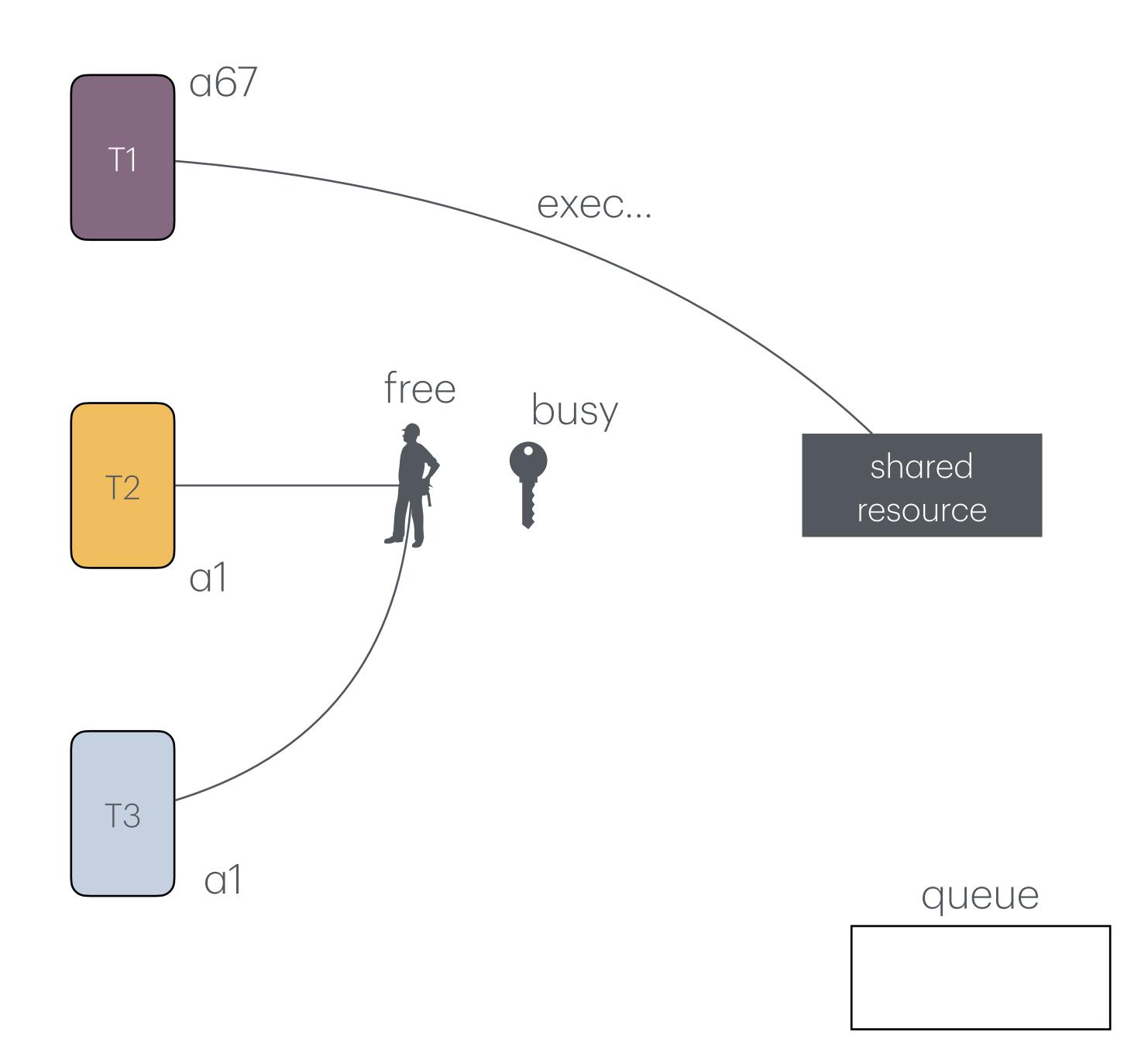
```
Timestep: 2
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



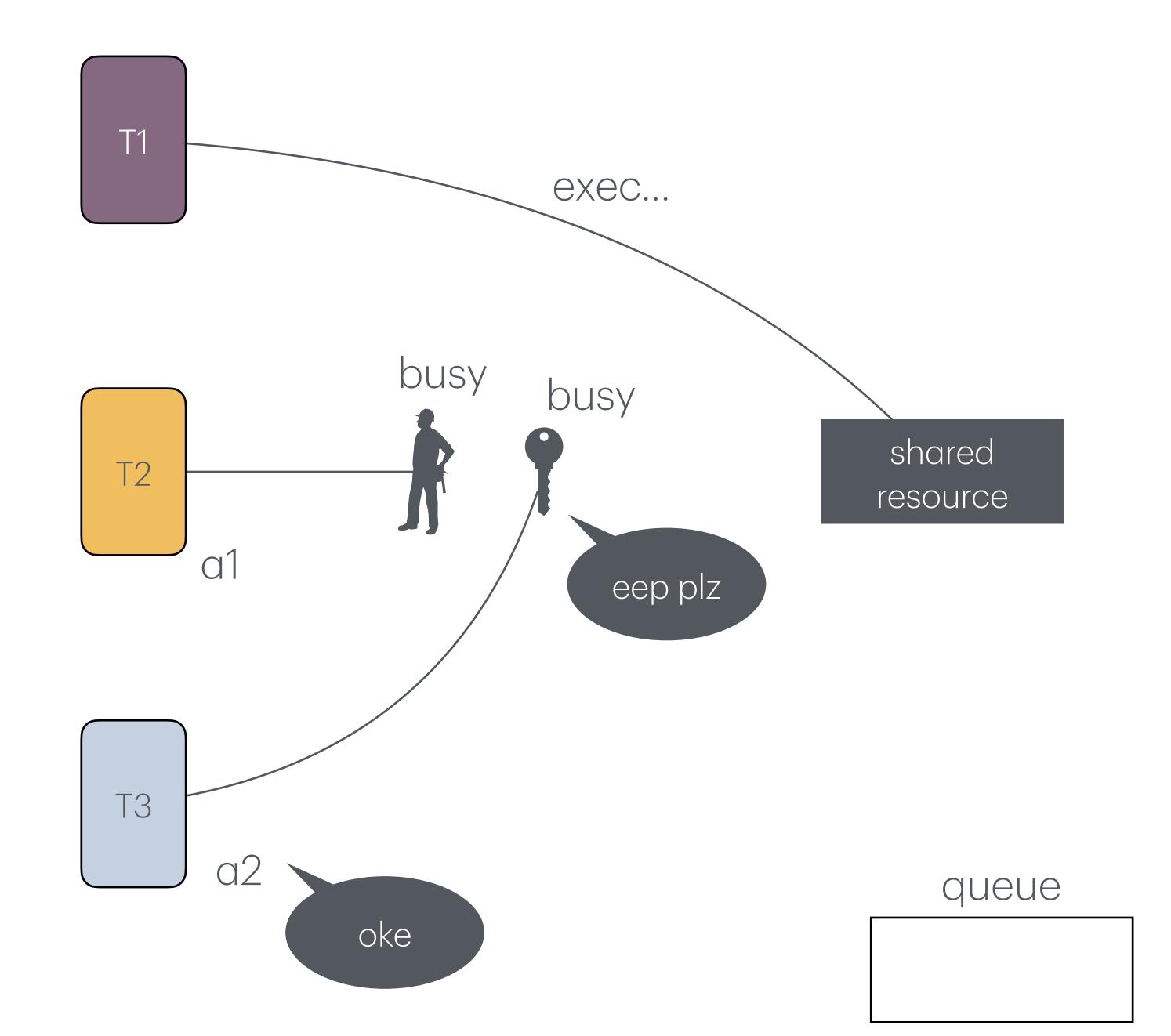
```
Timestep: 3
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 4
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 5
```

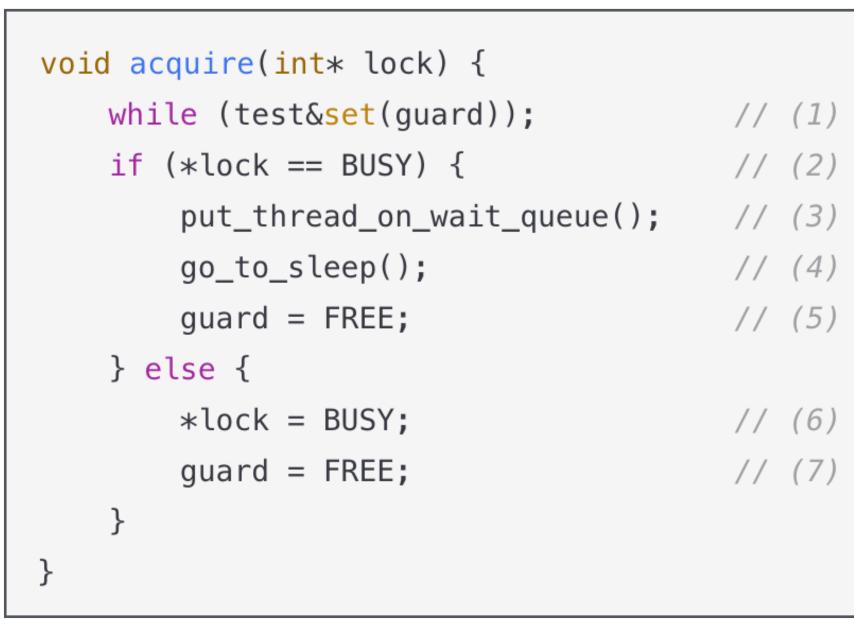
shared

resource

```
exec...
free
        busy
```

a1







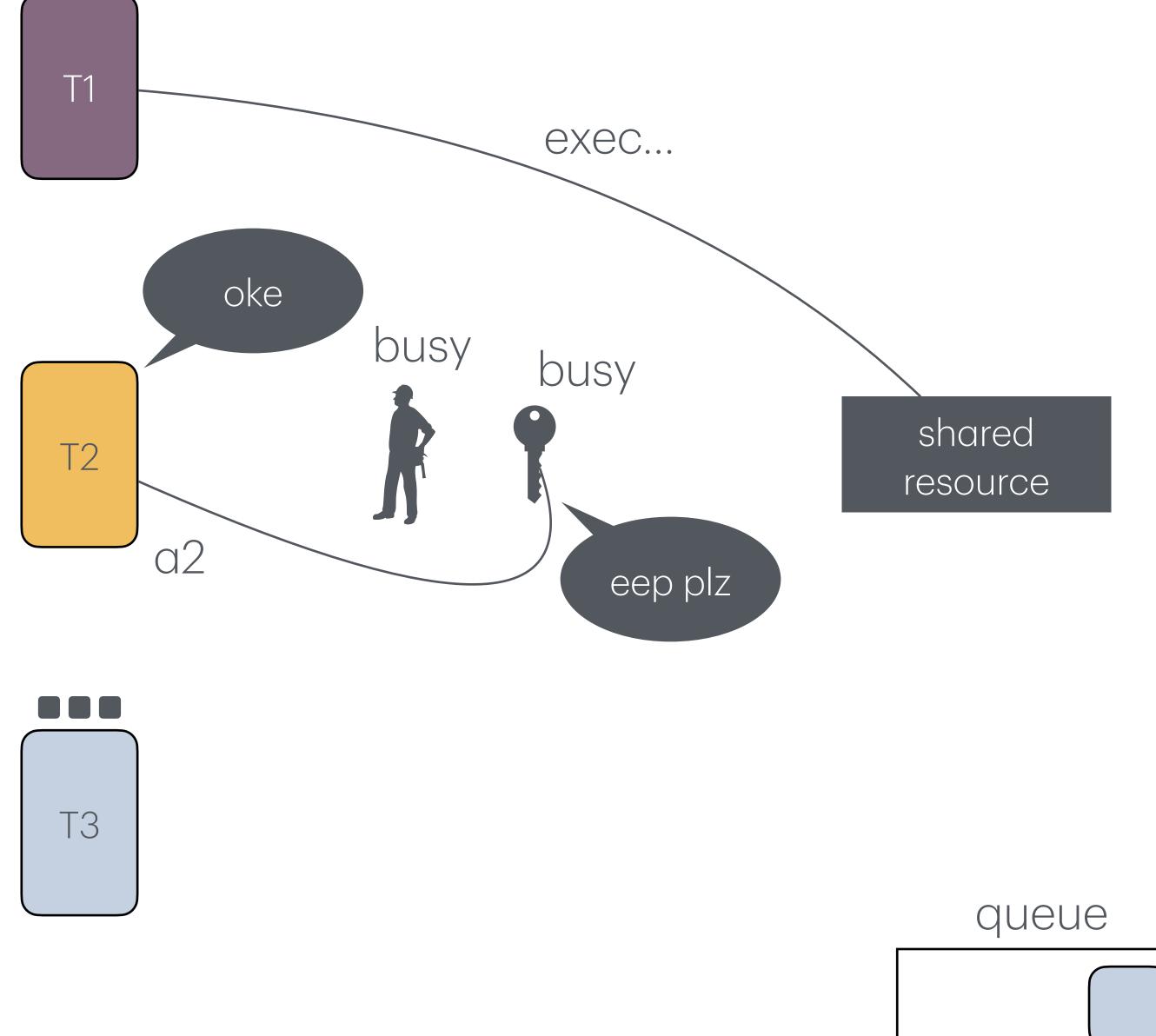
```
Timestep: 6
```

```
while (test&set(guard));
                                      // (1)
    if (*lock == BUSY) {
                                      // (2)
        put_thread_on_wait_queue();
                                      // (3)
        go_to_sleep();
                                      // (4)
        guard = FREE;
                                       // (5)
    } else {
        *lock = BUSY;
                                      // (6)
        guard = FREE;
                                       // (7)
                                                                         oke
                                                               T2
                                                                    a2
void release(int* lock) {
   while (test&set(guard));
                                       // (1)
   if (any_thread_on_wait_queue()) {
                                       // (2)
                                                             take_thread_off_wait_queue();
                                      // (3)
       place_thread_on_ready_queue();
   } else {
                                                              T3
       *lock = FREE;
```

// (6)

void acquire(int* lock) {

guard = FREE;



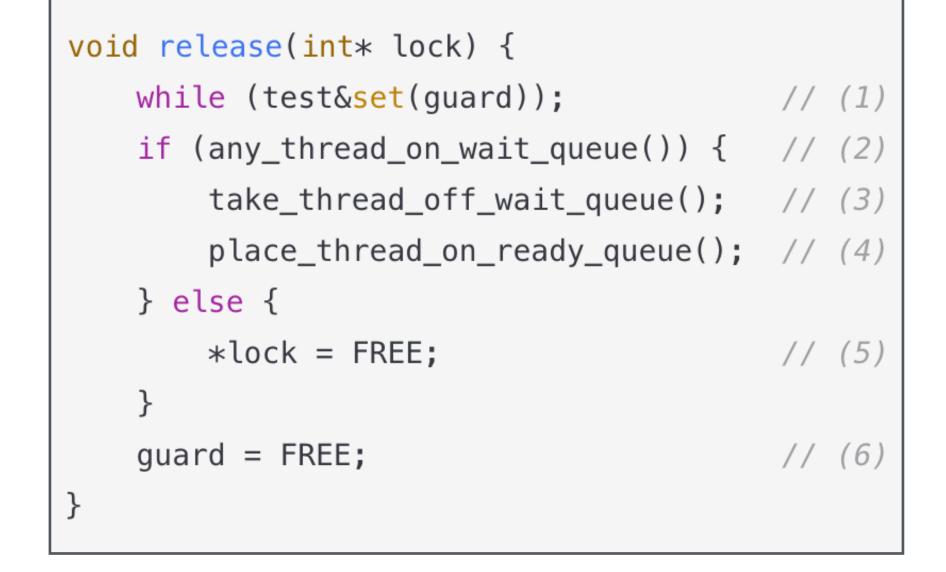
```
Timestep: 7
```

shared

resource

```
void acquire(int* lock) {
   while (test&set(guard));
                                     // (1)
   if (*lock == BUSY) {
                                     // (2)
       put_thread_on_wait_queue();
                                     // (3)
       go_to_sleep();
                                     // (4)
                                      // (5)
                                                                                            exec...
                                     // (6)
                                      // (7)
                                                                                 free
                                                                                            busy
```

a345



guard = FREE;

*lock = BUSY;

guard = FREE;

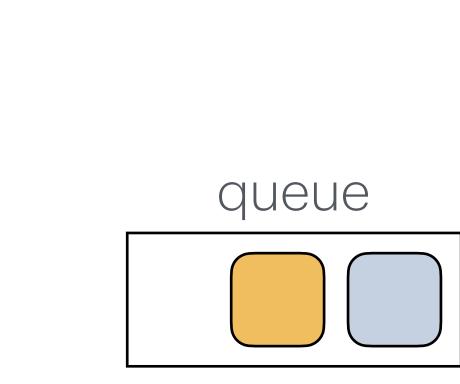
} else {



```
Timestep: 8
```

```
// (1)
                             // (2)
put_thread_on_wait_queue();
                             // (3)
                             // (4)
                             // (5)
                                                                                  exec...
                             // (6)
                             // (7)
                                                                       free
                                                                                 busy
                                                                                                         shared
                                                                                                        resource
```







void acquire(int* lock) {

if (*lock == BUSY) {

go_to_sleep();

guard = FREE;

*lock = BUSY;

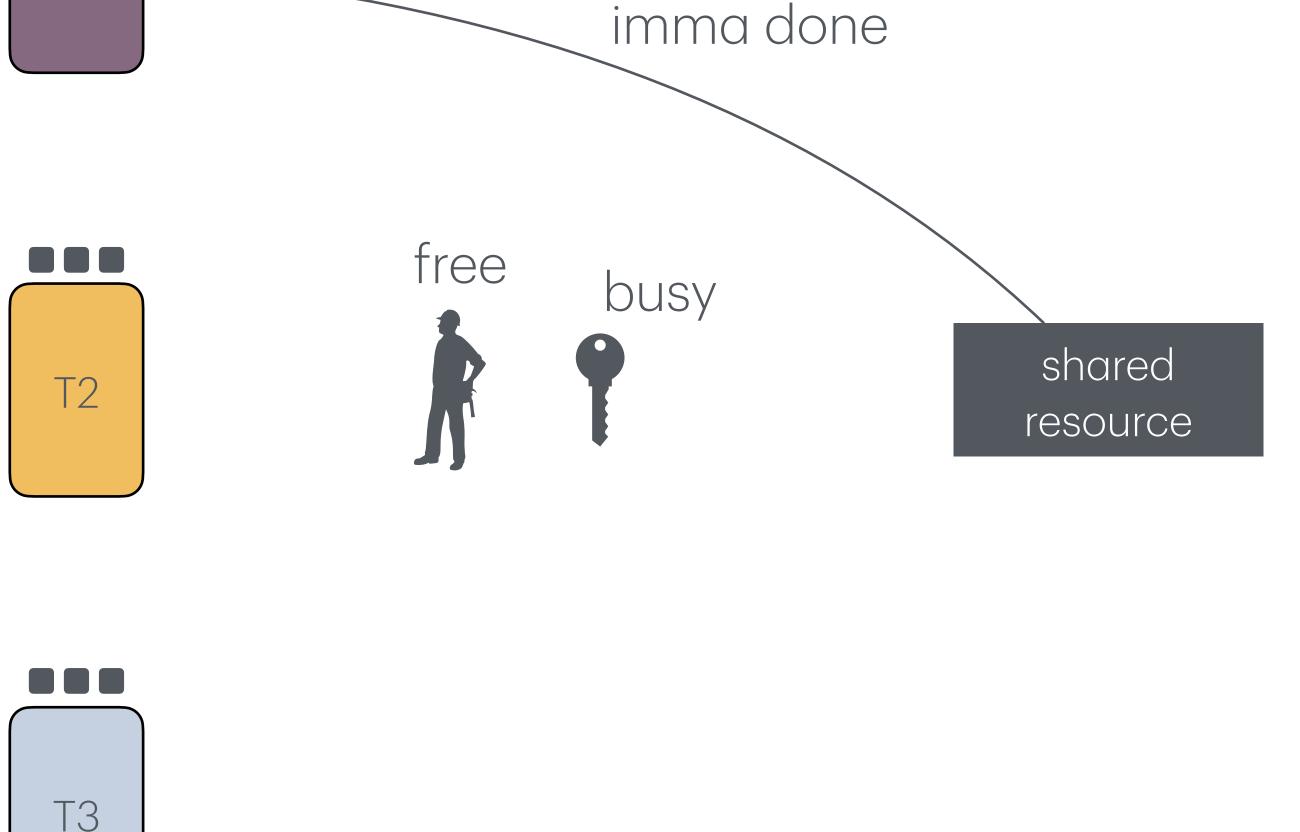
guard = FREE;

} else {

while (test&set(guard));

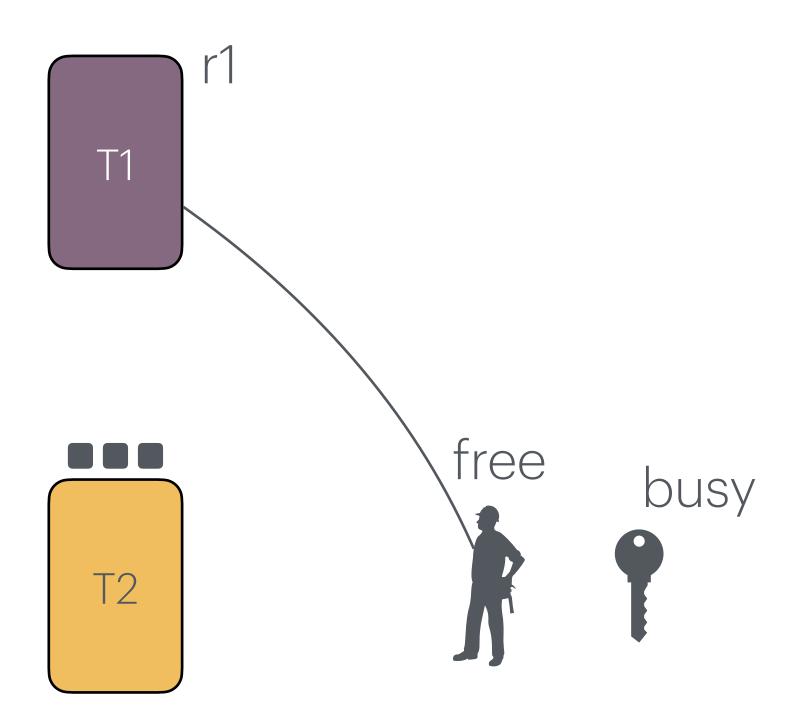
```
Timestep: 9
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```

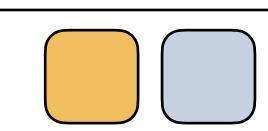


```
Timestep: 10
```

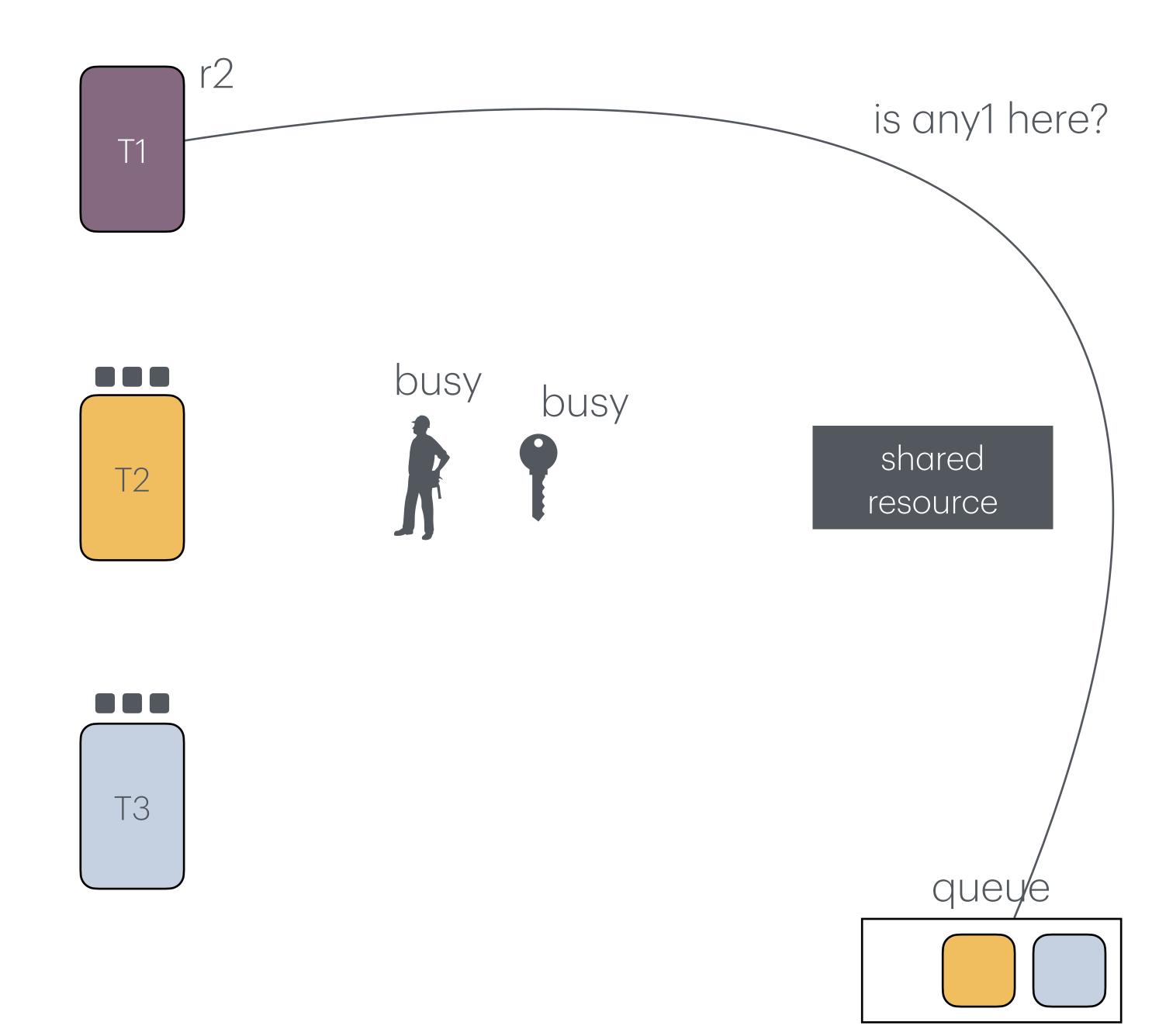
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                      // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```







```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
void acquire(int* lock) {
                                       // (1)
   while (test&set(guard));
   if (*lock == BUSY) {
                                       // (2)
       put_thread_on_wait_queue();
                                       // (3)
       go_to_sleep();
                                       // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                       // (6)
       guard = FREE;
                                       // (7)
```

void release(int* lock) {

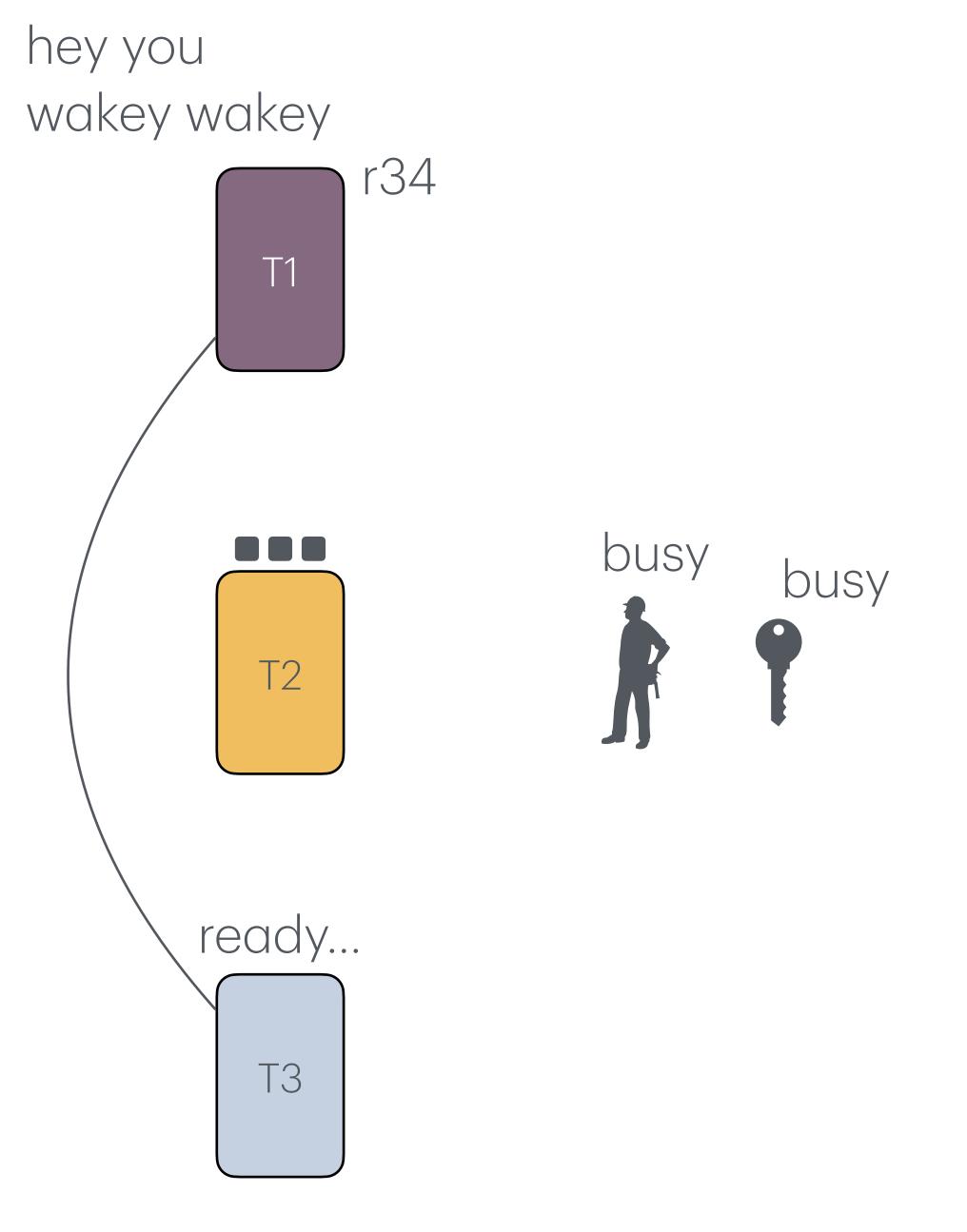
} else {

guard = FREE;

while (test&set(guard));

*lock = FREE;

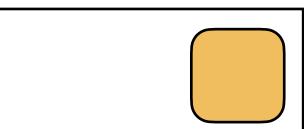
```
// (1)
if (any_thread_on_wait_queue()) {
                                   // (2)
   take_thread_off_wait_queue();
                                   // (3)
   place_thread_on_ready_queue();
                                    // (6)
```



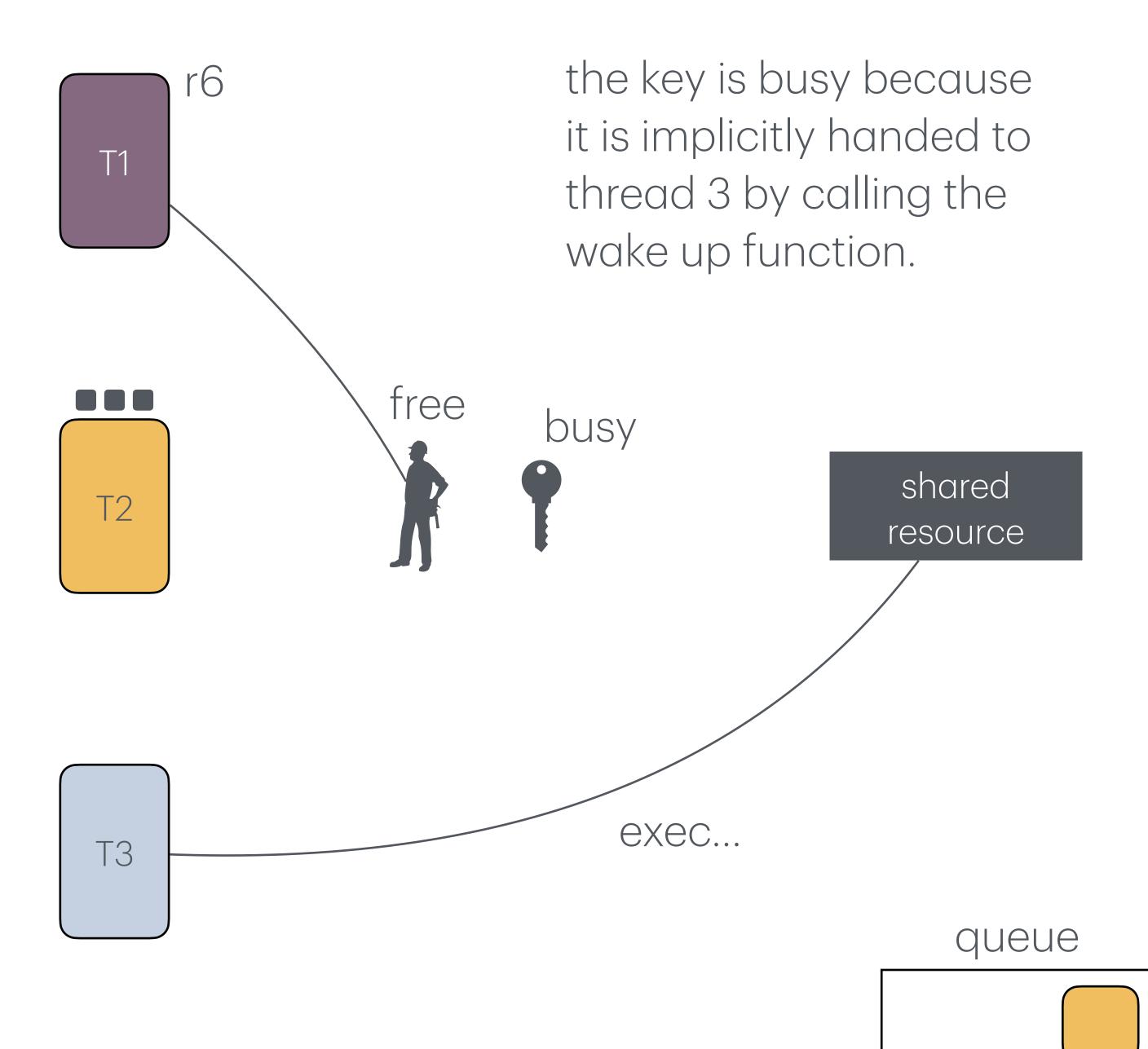
Timestep: 12

resource

shared

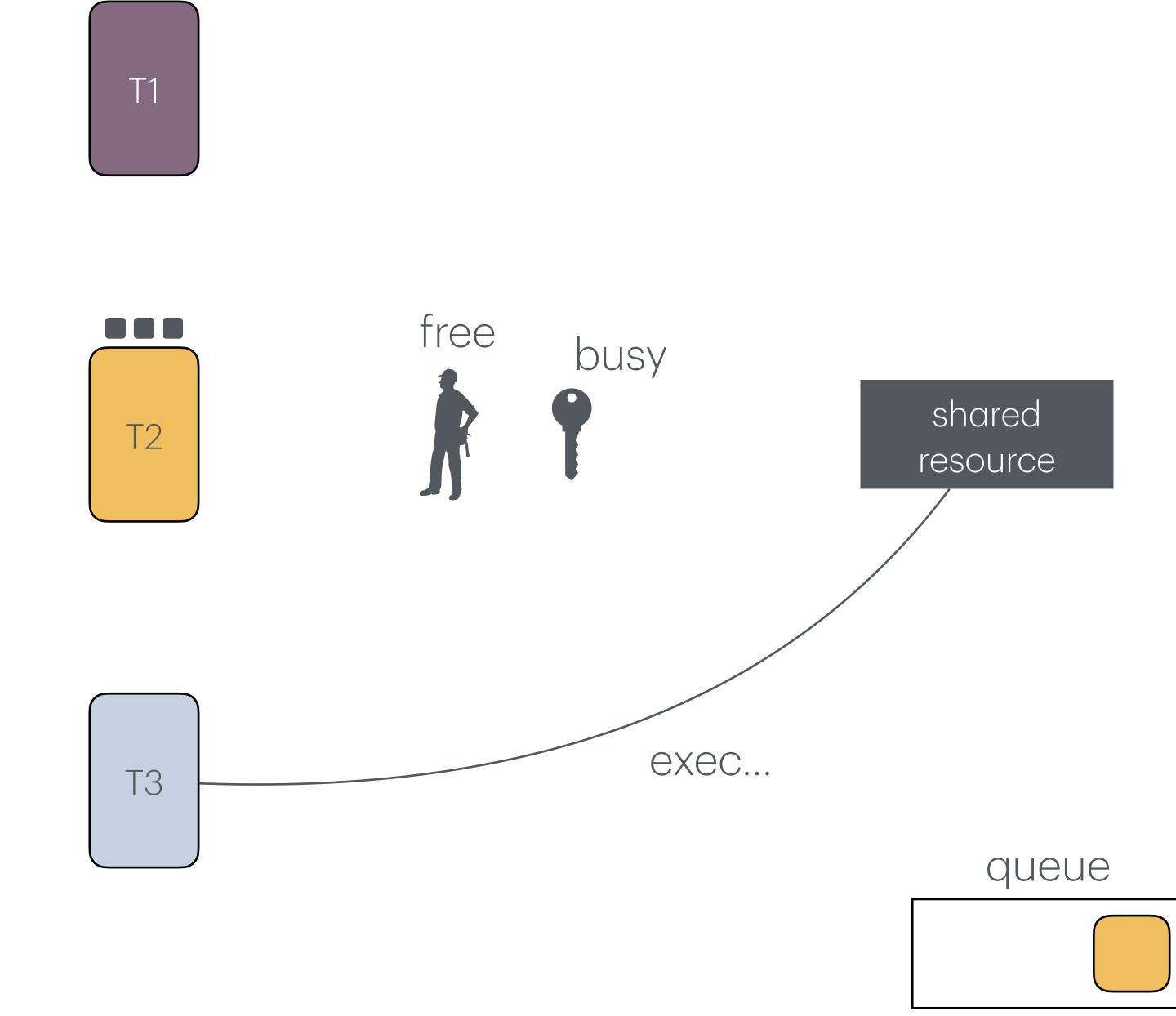


```
void acquire(int* lock) {
   while (test&set(guard));
                                       // (1)
   if (*lock == BUSY) {
                                       // (2)
                                      // (3)
       put_thread_on_wait_queue();
       go_to_sleep();
                                       // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                       // (6)
       guard = FREE;
                                       // (7)
```



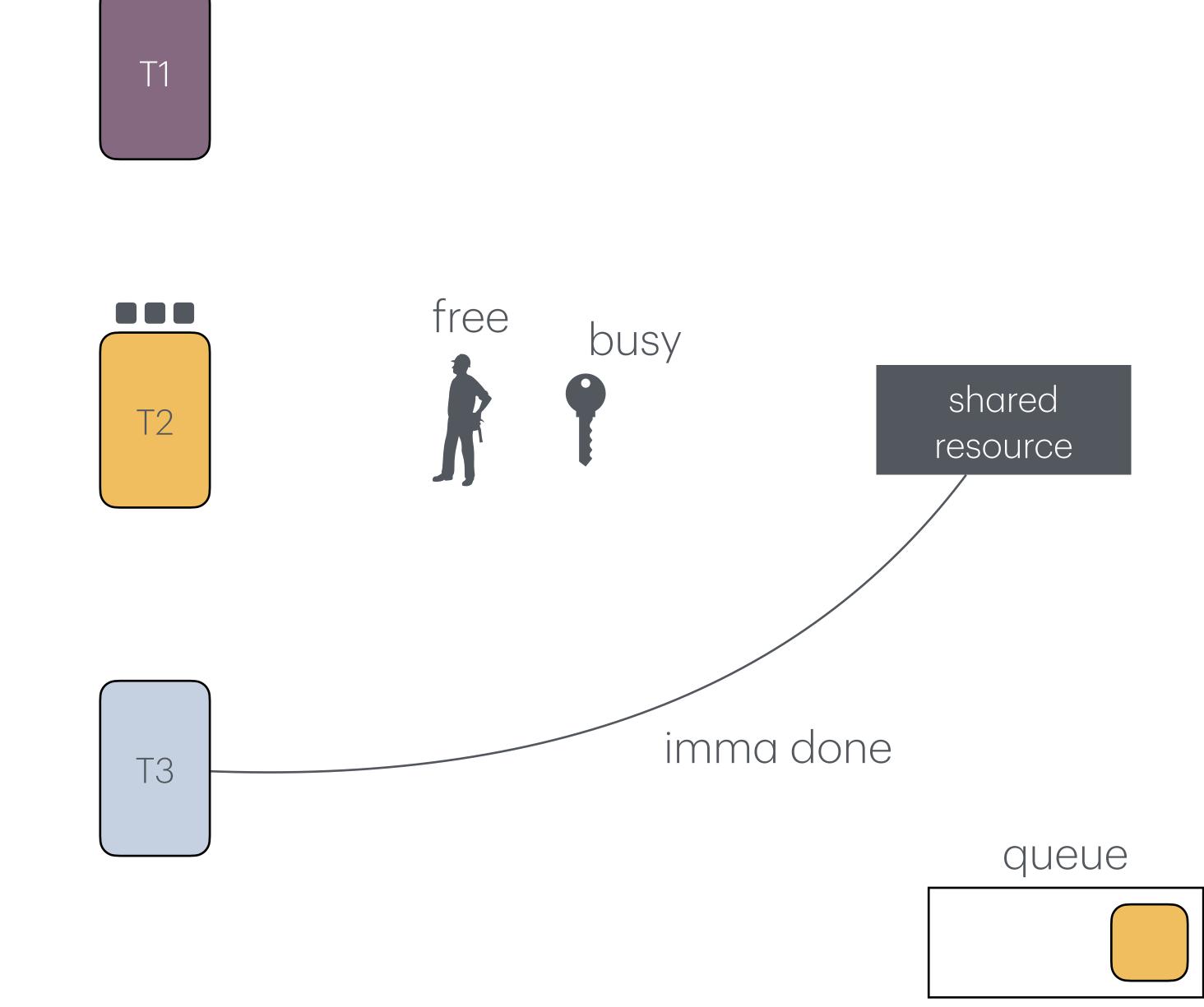
```
Timestep: 14
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



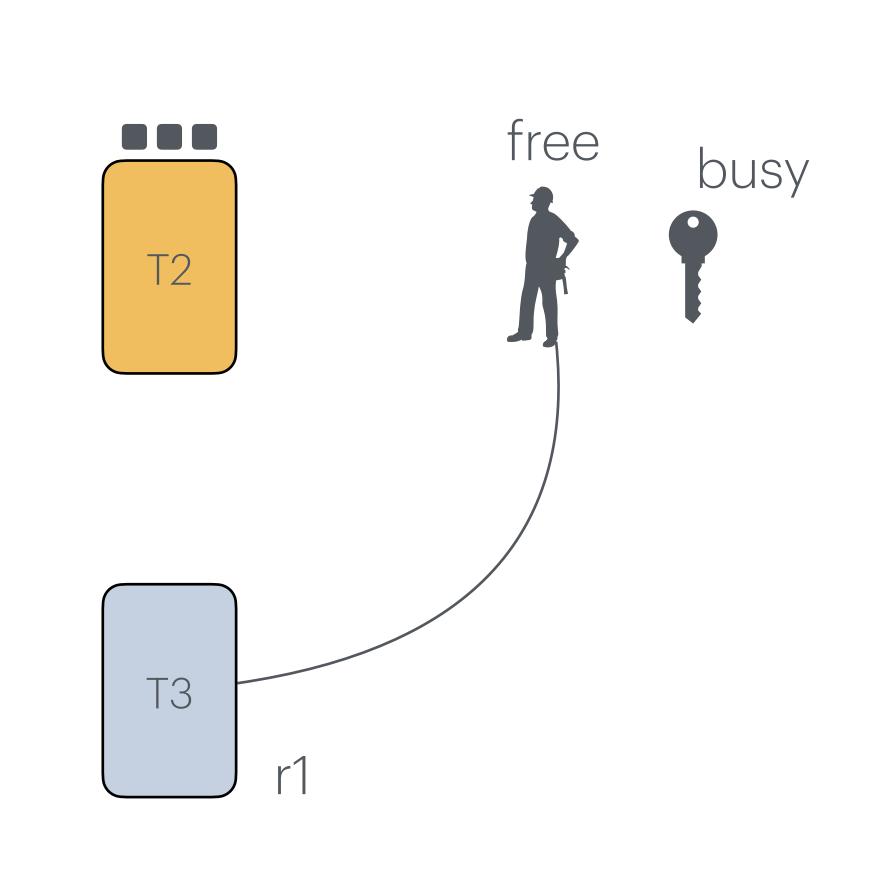
```
Timestep: 15
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 16
```

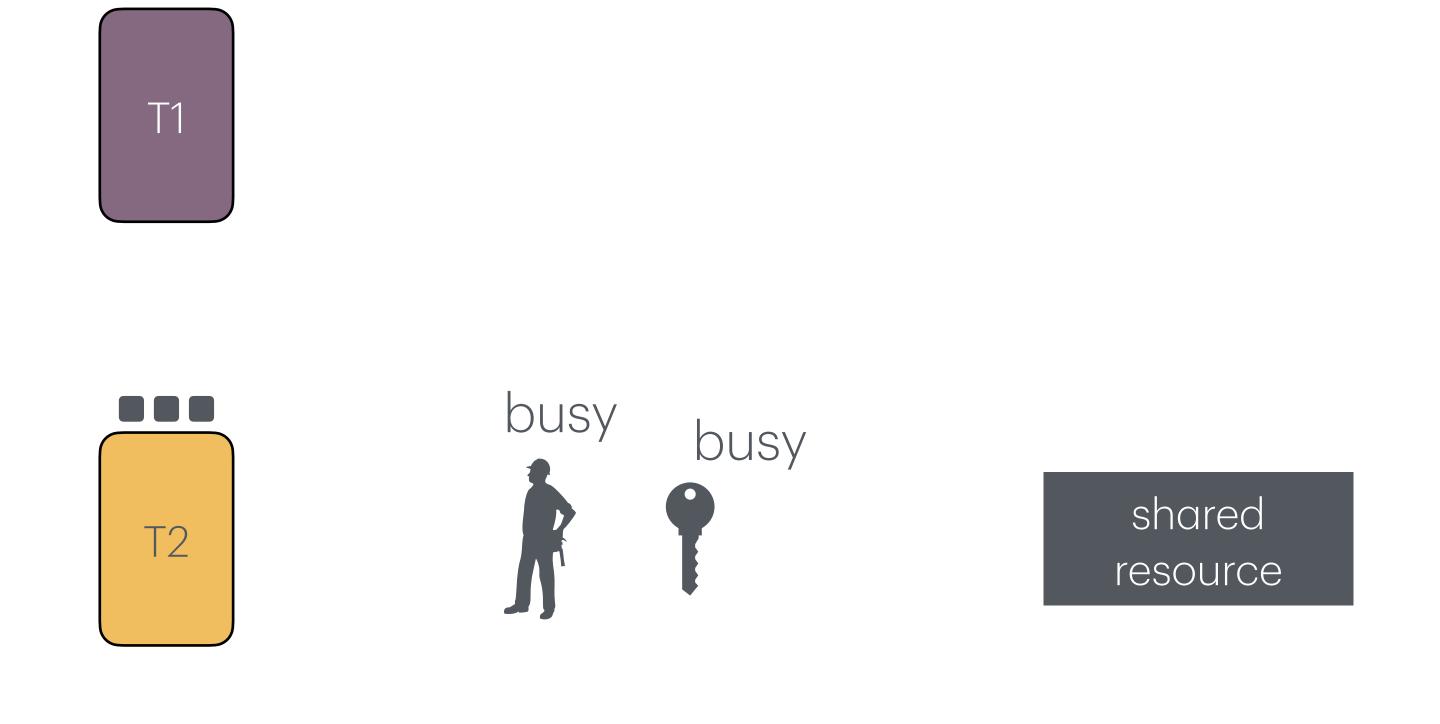
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 17
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                       // (1)
   if (*lock == BUSY) {
                                       // (2)
       put_thread_on_wait_queue();
                                       // (3)
       go_to_sleep();
                                       // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                       // (6)
       guard = FREE;
                                       // (7)
```

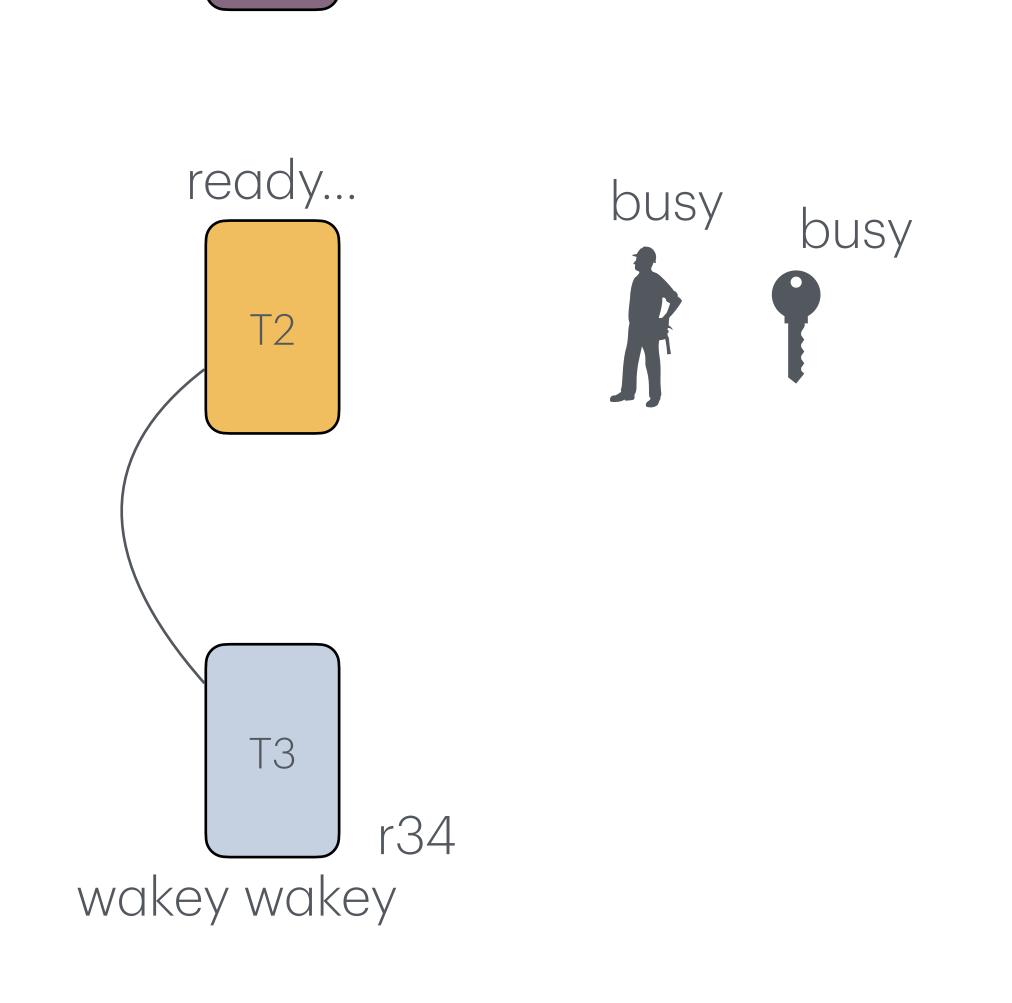






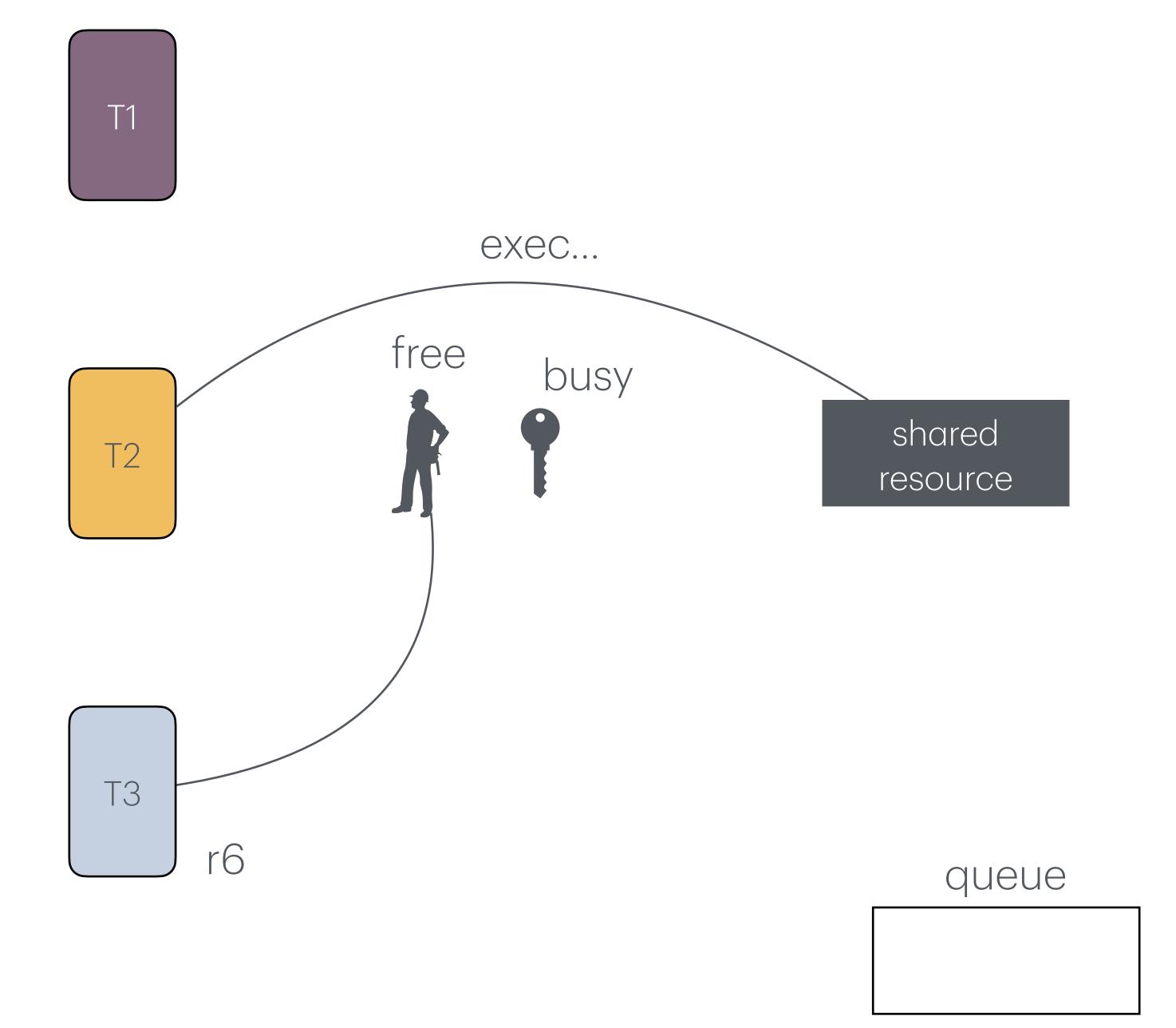
```
Timestep: 18
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



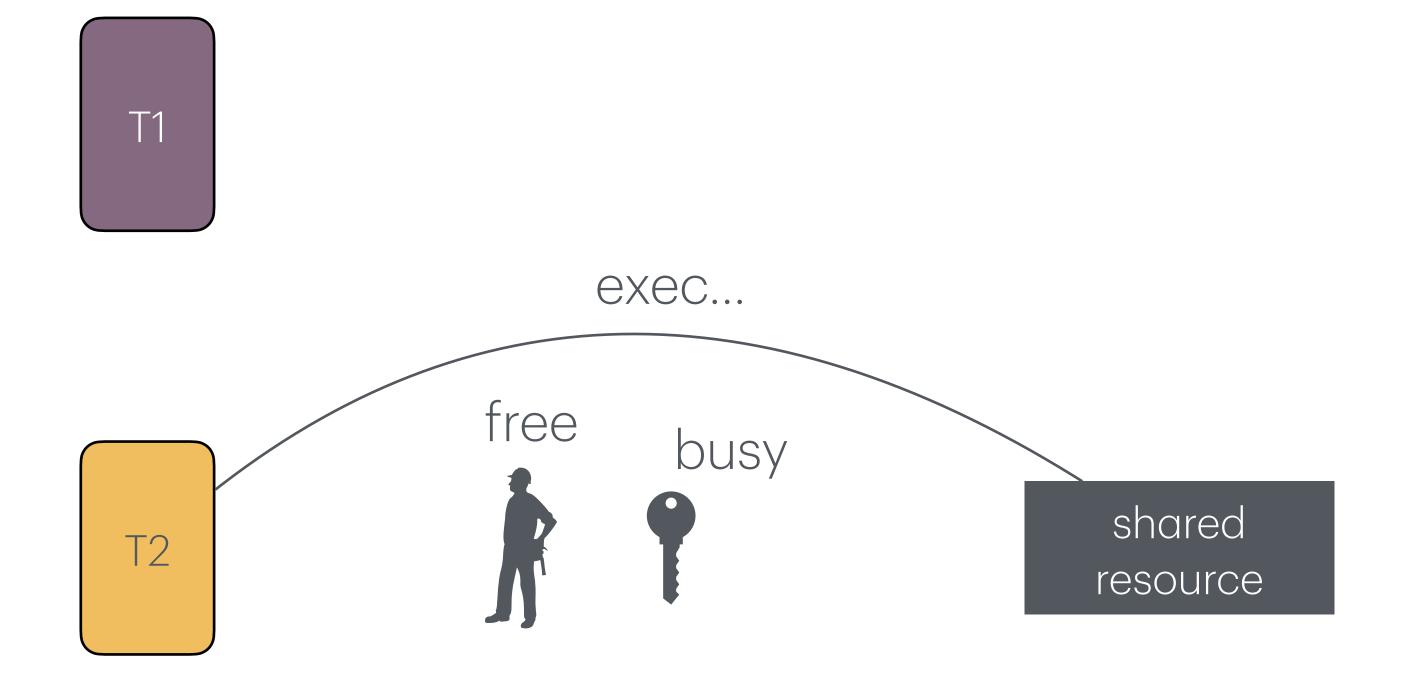
```
Timestep: 19
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 20
```

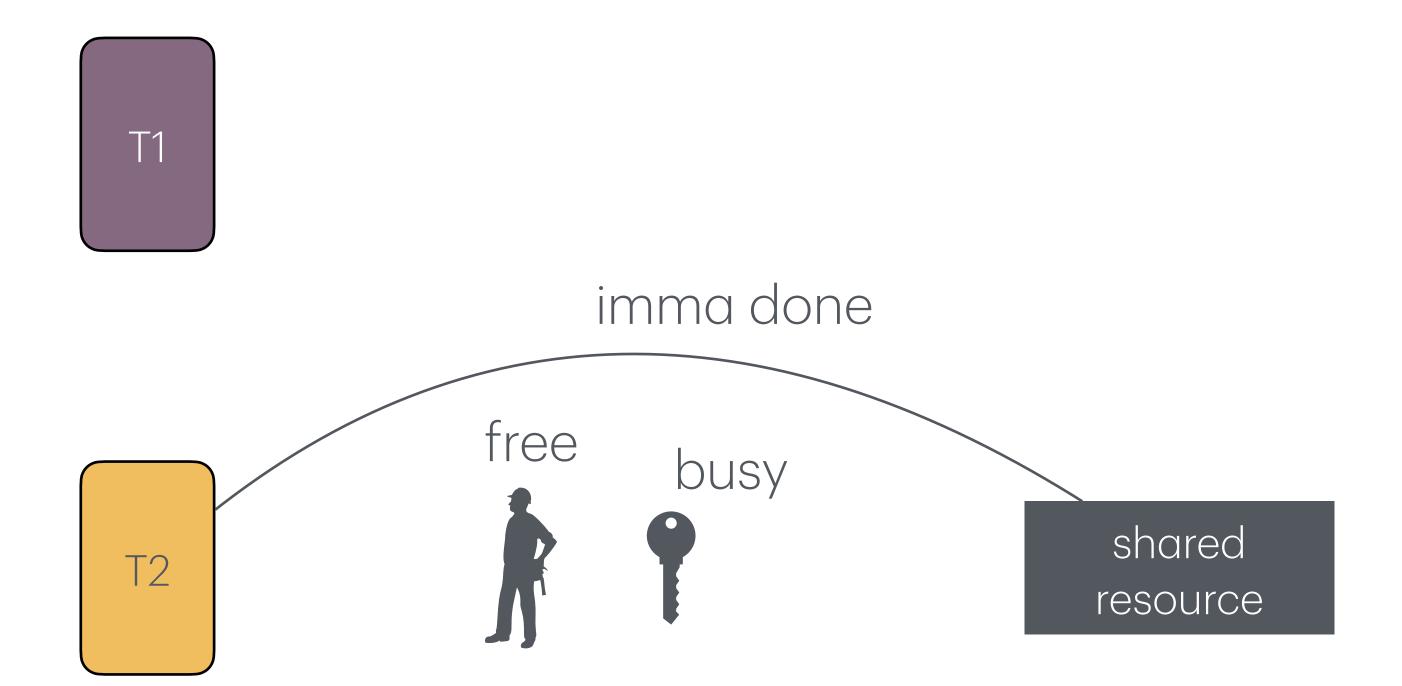
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```





```
Timestep: 21
```

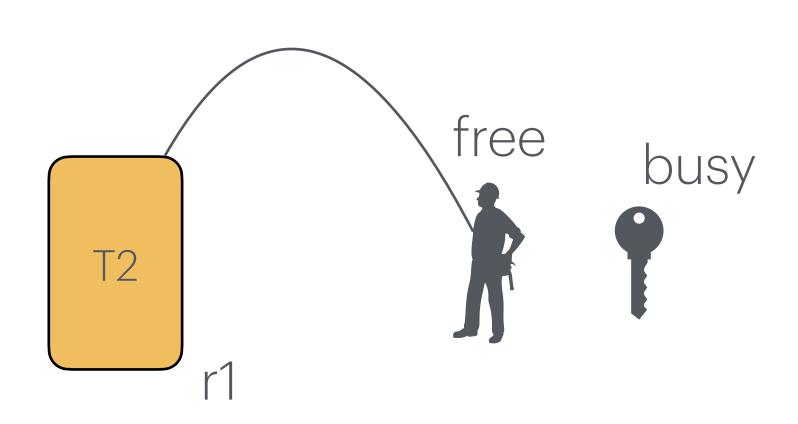
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```





```
Timestep: 22
```

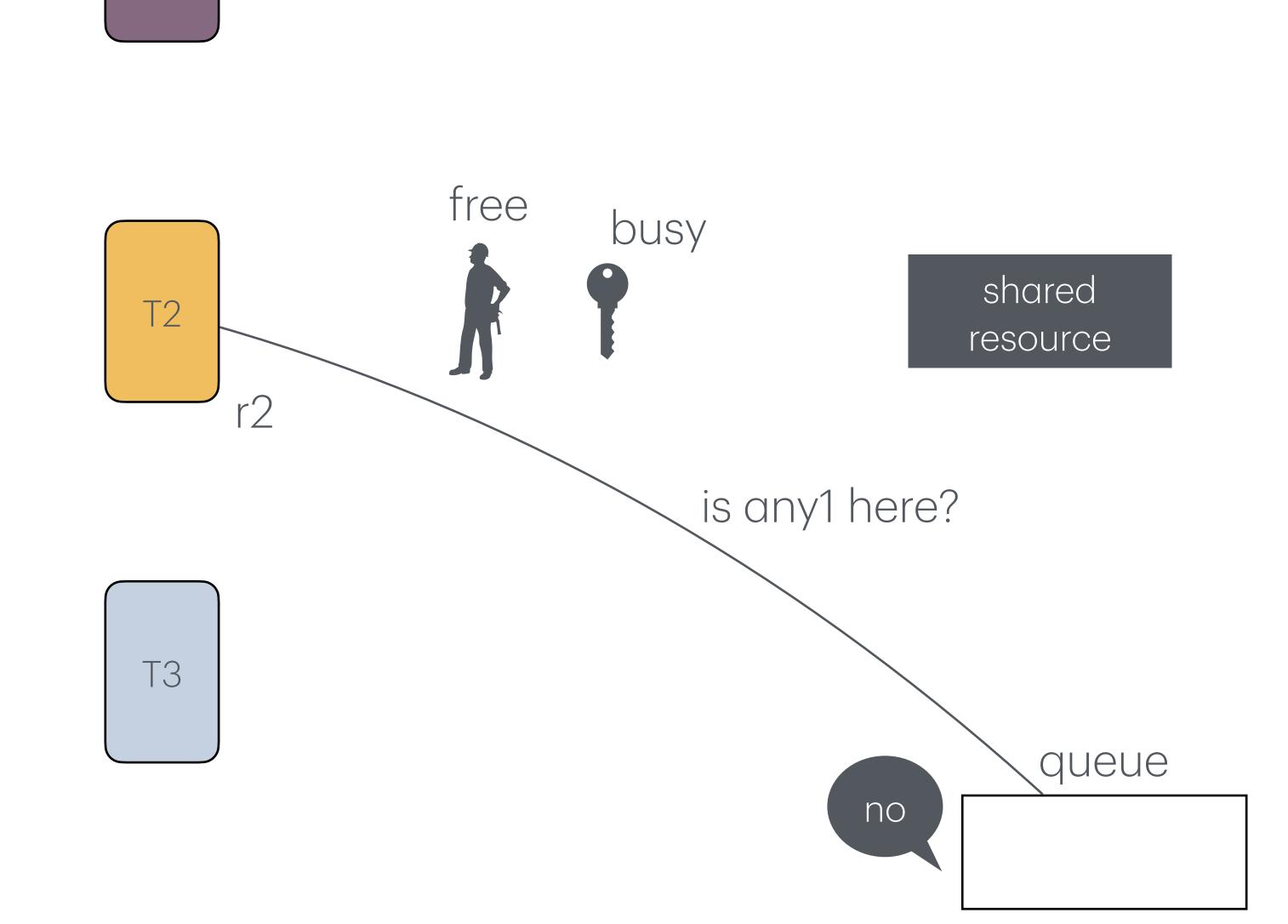
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



ТЗ

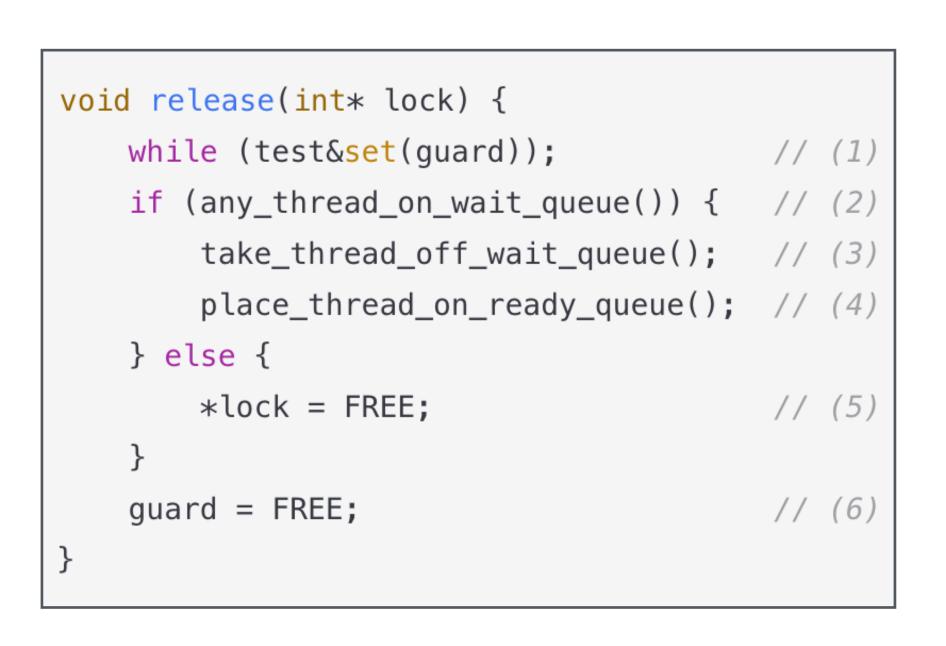
```
Timestep: 23
```

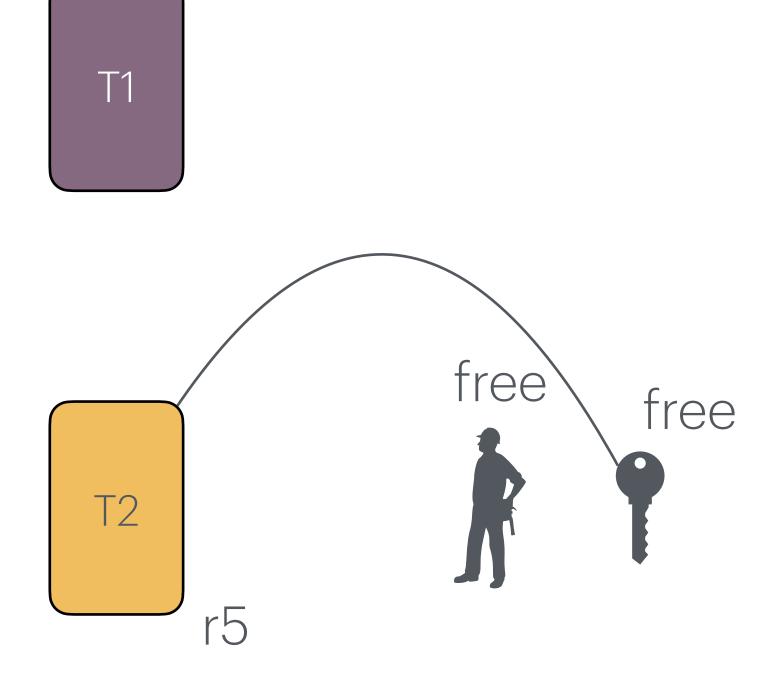
```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



```
Timestep: 24
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```

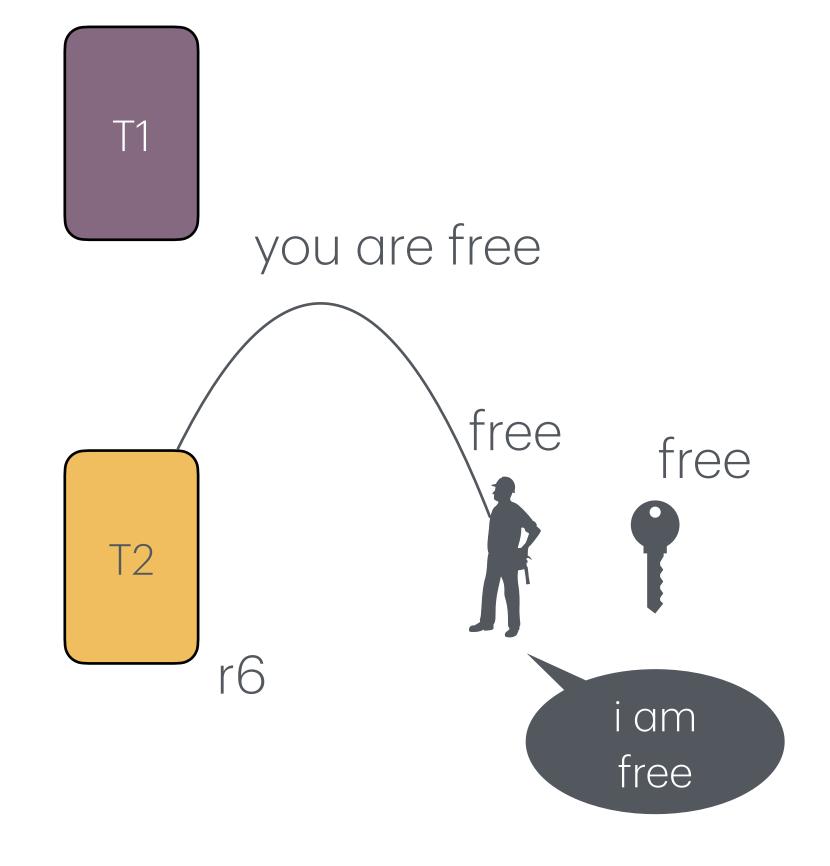




ТЗ

```
Timestep: 25
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                      // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```



ТЗ

```
Timestep: 26
```

```
void acquire(int* lock) {
   while (test&set(guard));
                                      // (1)
   if (*lock == BUSY) {
                                      // (2)
       put_thread_on_wait_queue();
                                      // (3)
       go_to_sleep();
                                      // (4)
       guard = FREE;
                                       // (5)
   } else {
       *lock = BUSY;
                                      // (6)
       guard = FREE;
                                       // (7)
```

```
T2 free
```

T3

shared resource