# Linux kernel lock

## -- rcu

# 16 locks:

1. atomic
2. barrier
3. spinlock
4. rwlock
5. seqlock
6. completion
7. mutex
8. semaphore
9. rw_semaphore
10. rcu_classic
11. rcu_preempt
12. rcu_tree
13. bkl
14. percpu
15. interrupts
16. preempt

# rcu_classic:

```
struct rcu_hlkw {
        int a;
        int b;
} *global_rcu_hlkw_p;
```

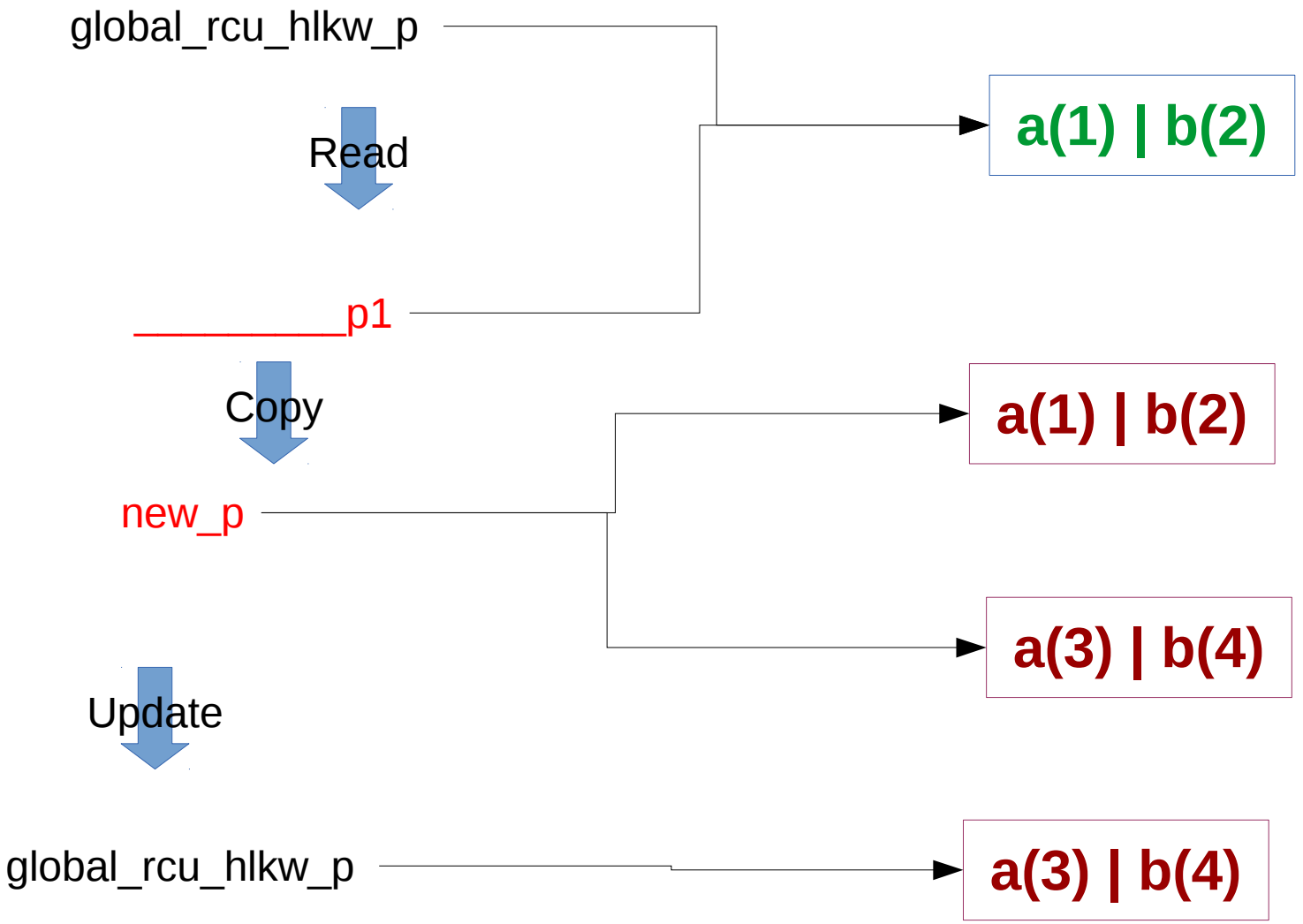
```
int reader(void) {
        int sum;
        struct rcu_hlkw *rcu_p;

        rcu_read_lock();
        rcu_p=rcu_dereference(global_rcu_hlkw_p);
        sum = rcu_p->a + rcu_p->b;
        rcu_read_unlock();
        return sum;
}
```

```
DEFINE_SPINLOCK(rcu_hlkw_mutex);
void writer(int new_a, int new_b){
        struct rcu_hlkw *new_p;
        struct rcu_hlkw *old_p;
        new_p = kmalloc(sizeof(struct rcu_hlkw), GFP_KERNEL);
        spin_lock(&rcu_hlkw_mutex);
        old_p = global_rcu_hlkw_p;
        *new_p = *old_fp;
        new_fp->a = new_a;
        new_fp->b = new_b;

        rcu_assign_pointer(global_rcu_hlkw_p, new_fp);
        spin_unlock(&rcu_hlkw_mutex);
        synchronize_rcu();
        kfree(old_p);
}
```

# rcu_classic:


global_rcu_hlkw_p
Read
a(1) | b(2)
p1
Copy
a(1) | b(2)
new_p
a(3) | b(4)
Update
global_rcu_hlkw_p
a(3) | b(4)

# rcu_classic:

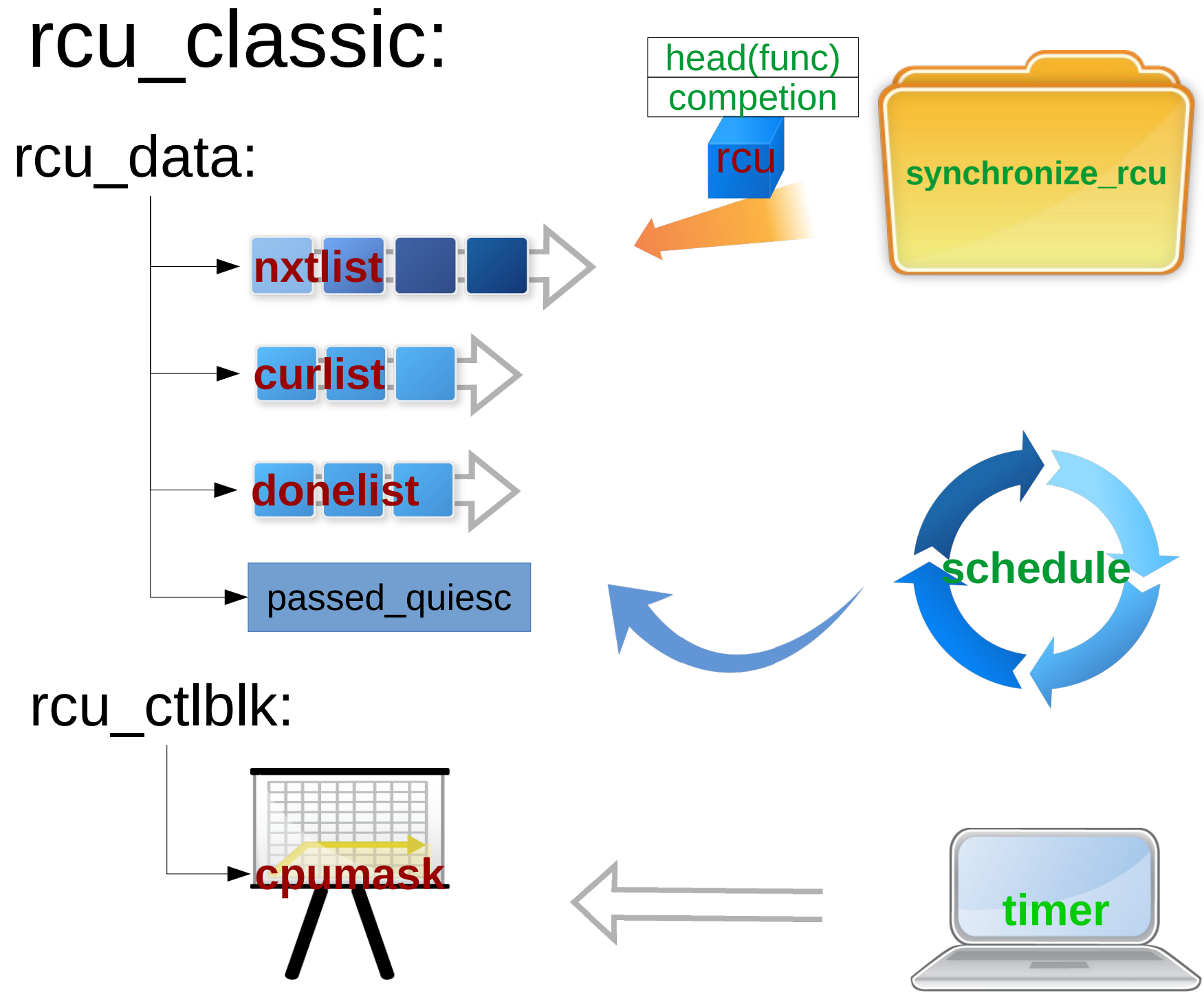
```
1. #define rcu_read_lock()  preempt_disable()
2. #define rcu_dereference(p)     ({ \
                                typeof(p) _________p1 = ACCESS_ONCE(p); \
                                smp_read_barrier_depends(); \
                                (_________p1); \
                                })

3. #define rcu_read_unlock() preempt_enable()
```

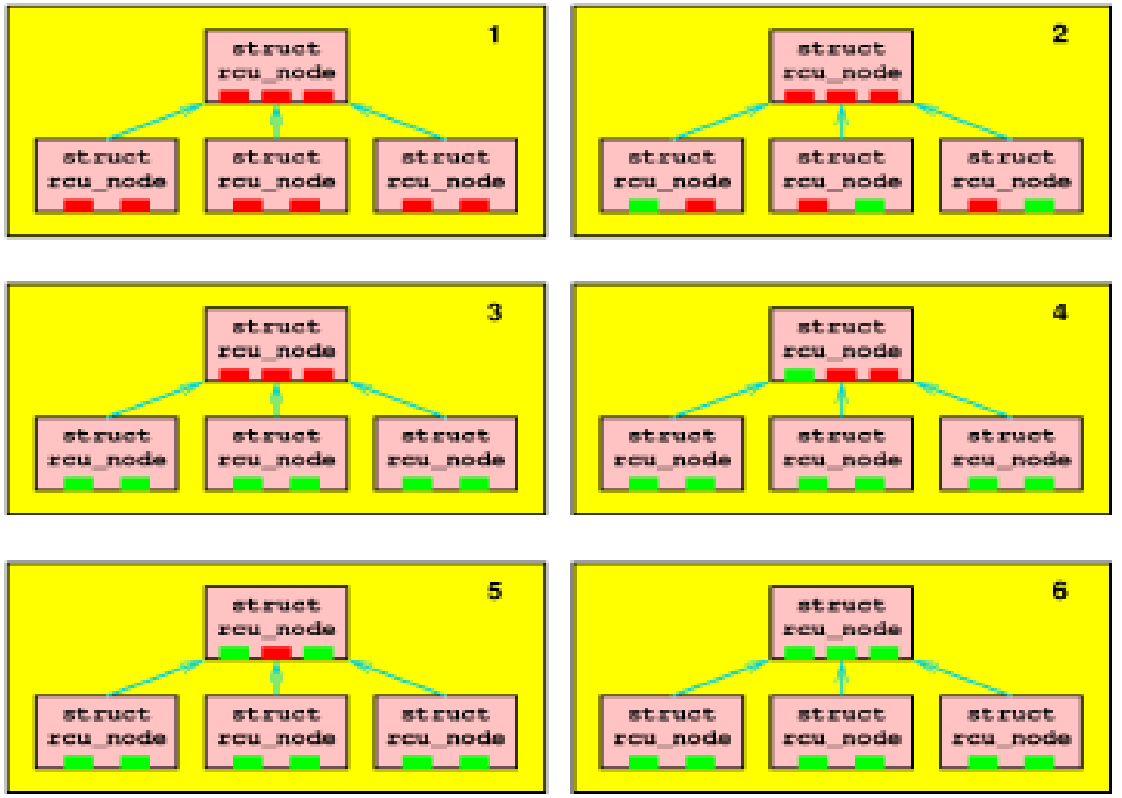
```
4. #define rcu_assign_pointer(p, v)        ({ \
                                        smp_wmb(); \
                                        (p) = (v); \
                                })

5. void synchronize_rcu(void)
{
      struct rcu_synchronize rcu;
      init_completion(&rcu.completion);
      call_rcu(&rcu.head, wakeme_after_rcu);
      wait_for_completion(&rcu.completion);
}
```

# rcu_classic:


head(func)
competion
rcu
synchronize_rcu
rcu_data:
nxtlist
curlist
donelist
passed_quiesc
schedule
rcu_ctlblk:
cpumask
timer

# rcu_tree:


1
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node
2
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node
3
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node
4
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node
5
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node
6
struct rcu_node
struct rcu_node
struct rcu_node
struct rcu_node

# 7 contexts:

1. task
2. trap
3. irq
4. softirq
5. tasklet
6. workqueue
7. locks nested

# rcu vs other locks