

# Synchronization / locks

## Scenarios

1. system call + system call — disable-pre-emption || single CPUs

↳ pre-emptible kernels!

↳ pre-emption can occur in any process context (including execution in system call)

### issues

- multi-CPU & multiple process context.

2. system call + interrupt handlers

↳ disable interrupts while in system call

perf. cost

dropping interrupts

with multi-CPU this does not work.

- interrupts

↳ execution in interrupt context.

3. LOCKS

(i) blocking locks  
spinning

↳ spinlocks, read-write locks...

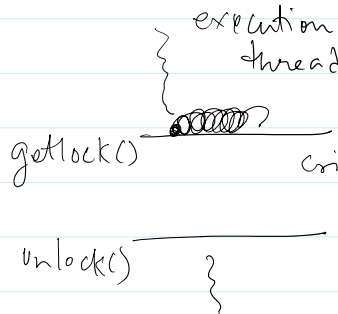
(ii) async. locks

wait — wakeup locks.  
sleep

- condition variables, semaphores

(iii) lockless designs

per-CPU data structures



system call  
Kernel thread  
interrupt handler

read-modify updates  
shared state

```
getlock();
getcandy();
dohappydance();
eatcandy();
releaselock();
```

```
getlock();
getcandy();
eatcandy();
releaselock();
```

RCU, RCU — versioning  
is a  
# Lock ⇒ memory object / variable.