

Synchronization.

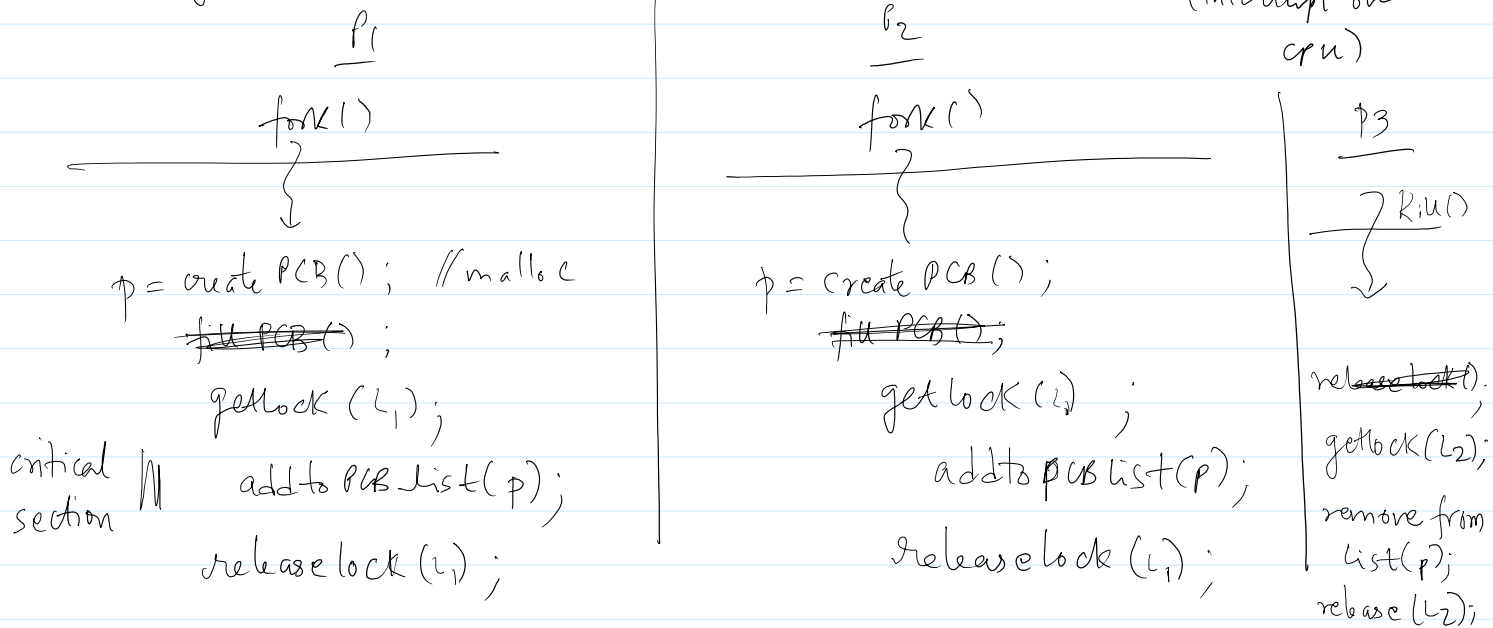
① disable pre-emption
 disable interrupts
 locks

Synchronization mechanisms.

each has applicability in diff. scenarios w/ diff. trade-offs.

⊗ for the most general case ||| all kernel execution scenarios + multi-CPU setup ||| requires... locks + local interrupt disable

② locking example



③ what is lock()? || spin lock | design 0

```

int  
void  
get lock (lock * lock) {  
    while (!) {  
        if (lock->locked == 0) { // available ~ A  
            lock->locked = 1;  
            break; ~ B  
        }  
    }  
}
    release lock (lock * lock)
    
```

Spin on CPU till lock available

lock \rightarrow locked = 0;

④ atomic instructions / atomicity on a set of logical instructions
 \hookrightarrow all or nothing!

* multiple 'C' statements \rightarrow non-atomic

* single 'C' statement \rightarrow non-atomic

* single assembly instruction \rightarrow may be a problem
w/w

\Rightarrow pipelining | fetch-decode-execute.

TSL — test-and-set — atomic

& can test & set memory variables.
reg.