

synchronization

- spin locks, non-blocking locks, mutex
 - spinning
 - 0/1 condition non-spinning lock.
- condition variables used to index blocked processes

decide block/runnable status of processes

```

if (condition) // not favourable
    wait (variable)
    sleep

if (work done)
    change condition
    wakeup
    signal (variable)
    wakeup
    
```

```

mutex_lock (mutex * m)
{
    spinlock (m->s);
    while (m->L)
        sleep (m->id, m->s);
    m->L = 1;
    spin_unlock (m->s);
}
    
```

```

mutex_unlock (mutex * m)
{
    spinlock (m->s);
    m->L = 0;
    sleep wakeup (m->id);
    spin_unlock (m->s);
}
    
```

```

struct mutex {
    
```

```

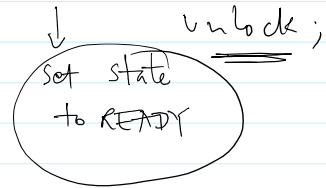
sleep (m->id, m->s) {
    looking xv6 == add to queue (m->id, curproc);
    state = BLOCKED
    spin_unlock (m->s);
    scheduler (); // got of the CPU
    spin_lock (m->s);
}
    
```

```

int L; // lock
int id; // variable
spinlock s;
    
```

```

wakeup (m->id)
{
    sleep remove from queue (m->id);
}
    
```



ptable.lock

serializable
the PCB list state

producer - consumer

bufsize = current size of produced elements.

```

producer () {
    
```

```

consumer () {
    
```

```
producer() {
```

```
    spinlock (S);
```

```
    buf size ++;
```

```
    wakeup (NO BUF);
```

```
    spin unlock (S);
```

```
}
```

```
consumer() {
```

```
    spinlock (S);
```

```
    while (if) (buf size == 0)  
        sleep (NO BUF, S);
```

```
    buf size --;
```

```
    spinunlock (S);
```

```
}
```