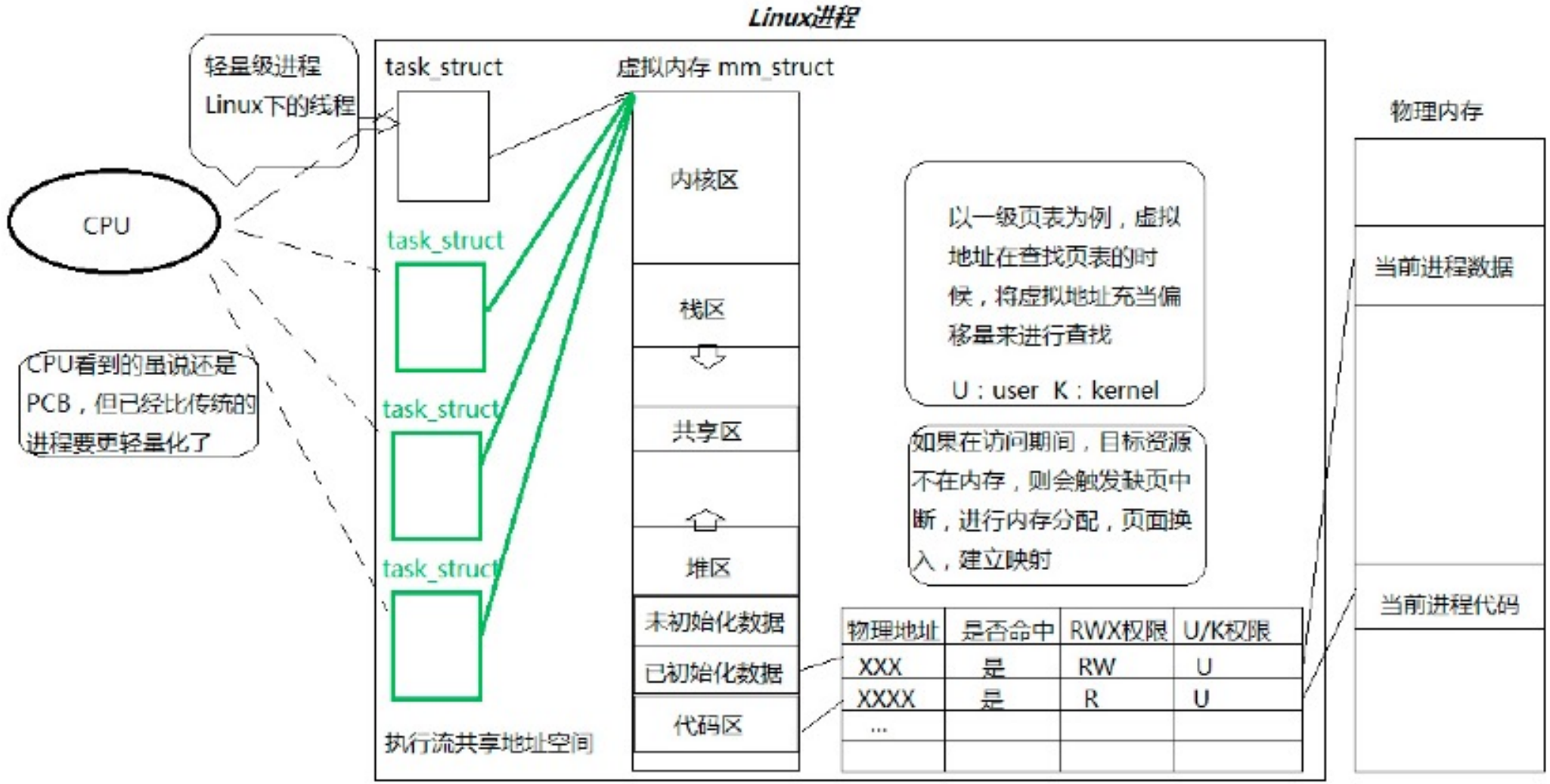
0207信号量



进程的内核数据结构+代码数据

进程是承担分配系统资源的基本实体

信号量

POSIX信号量和SystemV信号量作用相同，都是用于同步操作，达到无冲突的访问共享资源目的。 但POSIX可以用于线程间同步。

信号量本质是一个计数器

1. 什么是信号量
2. 如何理解信号量的使用

初始化信号量

```
#include <semaphore.h>
int sem_init(sem_t *sem, int pshared, unsigned int value);
参数:
    pshared:0表示线程间共享，非零表示进程间共享
    value: 信号量初始值
```

销毁信号量

```
int sem_destroy(sem_t *sem);
```

等待信号量

```
功能: 等待信号量，会将信号量的值减1
int sem_wait(sem_t *sem); //P()
```

发布信号量

```
功能: 发布信号量，表示资源使用完毕，可以归还资源了。将信号量值加1。
int sem_post(sem_t *sem);//V()
```

1. 共享资源 -> 任何一个时刻都只有一个执行流在访问 -> 临界资源、临界区的概念

共享资源互斥：
把这部分资源当作整体看待

但是这个共享资源
是被不同线程访问不同的部分，是不是也可以呢？

如果我们还是用互斥的方式去处理，虽然是正确的，**但是是不是效率太低了？**

只要保证执行流访问的是共享资源的不同部分，就能让执行流同时去访问

电影院的例子：
买票的本质：
叫做资源的预定机制
信号量本质：
是一个计数器，发访问临界资源的时候，休闲申请信号量（sem--，预定资源，P操作），使用完毕信号量资源（sem++，释放资源，V操作）。

问题来了：
1. 你怎么知道一共有多少个资源，还剩多少个？
2. 你怎么保证这个资源就是给你的（程序员编码保证）？我怎么知道我一定可以有一个共享资源呢？（信号量保证）

如何理解信号量的使用

我们申请了一个信号量-> 当前执行流一定具有一个资源，可以被它使用啦 -> 是哪一个资源呢？？需要程序员结合场景，自定义编码完成的！