深入浅出 down_interruptible 函数

int down_interruptible(struct semaphore *sem)
这个函数的功能就是获得信号量，如果得不到信号量就睡眠，此时没有信号打断，那么进入睡眠。但是在睡眠过程中可能被信号打断，打断之后返回-EINTR，主要用来进程间的互斥同步。

下面是该函数的注释：

```
/**
* down_interruptible - acquire the semaphore unless interrupted
* @sem: the semaphore to be acquired
*
* Attempts to acquire the semaphore. If no more tasks are allowed to
* acquire the semaphore, calling this function will put the task to sleep.
* If the sleep is interrupted by a signal, this function will return -EINTR.
* If the semaphore is successfully acquired, this function returns 0.
*/
```

一个进程在调用 down_interruptible()之后，如果 sem<0，那么就进入到可中断的睡眠状态并调度其它进程运行， 但是一旦该进程收到信号，那么就会从 down_interruptible 函数中返回。并标记错误号为:-EINTR。一个形象的比喻：传入的信号量为 1 好比天亮，如果当前信号量为 0，进程睡眠，直到（信号量为 1）天亮才醒，但是可能中途有个闹铃（信号）把你闹醒。又如：小强下午放学回家，回家了就要开始吃饭嘛，这时就会有两种情况：情况一：饭做好了，可以开始吃；情况二：当他到厨房去的时候发现妈妈还在做，妈妈就对他说：“你先去睡会，待会做好了叫你。”小强就答应去睡会，不过又说了一句：“睡的这段时间要是小红来找我玩，你可以叫醒我。”小强就是 down_interruptible，想吃饭就是

获取信号量，睡觉对应这里的休眠，而小红来找我玩就是中断休眠。

并未详细说明中断信号的发出方式和方法。

使用可被中断的信号量版本的意思是，万一出现了 semaphore 的死锁，还有机会用 ctrl+c 发出软中断，让等待这个内核驱动返回的用户态进程退出。而不是把整个系统都锁住了。在休眠时，能被中断信号终止，这个进程是可以接受中断信号的！比如你在命令行中输入# sleep 10000，按下 ctrl＋c，就给上面的进程发送了进程终止信号。信号发送给用户空间，然后通过系统调用，会把这个信号传给递给驱动。信号只能发送给用户空间，无权直接发送给内核的，那 1G 的内核空间，我们是无法直接去操作的。