

Linux系统内核定时器机制详解（下）（2）PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_Linux\\_E7\\_B3\\_BB\\_E7\\_BB\\_c103\\_144976.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_B3_BB_E7_BB_c103_144976.htm) 7 . 6 . 3 . 5 修改一个定时器的expires值

当一个定时器已经被插入到内核动态定时器链表中后，我们还可以修改该定时器的expires值。函数

`mod_timer()`实现这一点。如下所示（`kernel/timer.c`）：  

```
int mod_timer(struct timer_list *timer, unsigned long expires) { int ret. unsigned long flags. spin_lock_irqsave(&timerlist_lock, flags). return ret. }
```

该函数首先根据参数expires值更新定时器的expires成员。然后调用`detach_timer()`函数将该定时器从它原来所属的链表中删除。最后调用`internal_add_timer()`函数将该定时器根据它新的expires值重新插入到相应的链表中。函数

`detach_timer()`首先调用`timer_pending()`来判断指定的定时器是否已经处于某个链表中，如果定时器原来就不处于任何链表中，则`detach_timer()`函数什么也不做，直接返回0值，表示失败。否则，就调用`list_del()`函数将定时器从它原来所处的链表中摘除。如下所示（`kernel/timer.c`）：

```
static inline int detach_timer (struct timer_list *timer) { if (!timer_pending(timer)) return 0. list_del(&timerlist_lock, flags). ret = detach_timer(timer). timer->list.next = timer->list.prev = NULL. spin_unlock_irqrestore(&timerlist_lock, flags). return ret. }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)