

Linux系统内核定时器机制详解（上）（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_144891.htm

7.6.2 动态内核定时器机制的原理 Linux是怎样为其内核定时器机制提供动态扩展能力的呢？其关键就在于“定时器向量”的概念。所谓“定时器向量”就是指这样一条双向循环定时器队列（对列中的每一个元素都是一个timer_list结构）：对列中的所有定时器都在同一个时刻到期，也即对列中的每一个timer_list结构都具有相同的expires值。显然，可以用一个timer_list结构类型的指针来表示一个定时器向量。显然，定时器expires成员的值与jiffies变量的差值决定了一个定时器将在多长时间后到期。在32位系统中，这个时间差值的最大值应该是0xffffffff。因此如果是基于“定时器向量”基本定义，内核将至少要维护0xffffffff个timer_list结构类型的指针，这显然是不现实的。另一方面，从内核本身这个角度看，它所关心的定时器显然不是那些已经过期而被执行过的定时器（这些定时器完全可以被丢弃），也不是那些要经过很长时间才会到期的定时器，而是那些当前已经到期或者马上就要到期的定时器（注意！时间间隔是以滴答次数为计数单位的）。基于上述考虑，并假定一个定时器要经过interval个时钟滴答后才到期（ $interval = expires - jiffies$ ），则Linux采用了下列思想来实现其动态内核定时器机制：对于那些 $0 < interval < 255$ 的定时器，Linux严格按照定时器向量的基本语义来组织这些定时器，也即Linux内核最关心那些在接下来的255个时钟节拍内就要到期的定时器，因此将它们按照各自不同的expires值组织成256个定时器向量。而

对于那些256 interval 0xffffffff的定时器，由于他们离到期还有一段时间，因此内核并不关心他们，而是将它们以一种扩展的定时器向量语义（或称为“松散的定时器向量语义”）进行组织。所谓“松散的定时器向量语义”就是指：各定时器的expires值可以互不相同的一个定时器队列。具体的组织方案可以分为两大部分：（1）对于内核最关心的、interval值在[0, 255]之间的前256个定时器向量，内核是这样组织它们的：这256个定时器向量被组织在一起组成一个定时器向量数组，并作为数据结构timer_vec_root的一部分，该数据结构定义在kernel/timer.c文件中，如下述代码段所示：

```
/* * Event timer code */ #define TVN_BITS 6 #define TVR_BITS 8 #define TVN_SIZE (1 <math>100</math>Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com
```