

Linux系统进程间隔定时器Itimer(上) (1) 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_144906.htm 所谓“间隔定时器 (Interval

Timer, 简称itimer) 就是指定定时器采用“间隔”值 (interval) 来作为计时方式, 当定时器启动后, 间隔值interval将不断减小。当interval值减到0时, 我们就说该间隔定时器到期。与上一节所说的内核动态定时器相比, 二者最大的区别在于定时器的计时方式不同。内核定时器是通过它的到期时刻expires值来计时的, 当全局变量jiffies值大于或等于内核动态定时器的expires值时, 我们说内核定时器到期。而间隔定时器则实际上是通过一个不断减小的计数器来计时的。虽然这两种定时器并不相同, 但却也是相互联系的。假如我们每个时钟节拍都使间隔定时器的间隔计数器减1, 那么在这种情形下间隔定时器实际上就是内核动态定时器 (下面我们会看到进程的真实间隔定时器就是这样通过内核定时器来实现的)。间隔定时器主要被应用在用户进程上。每个Linux进程都有三个相互关联的间隔定时器。其各自的间隔计数器都定义在进程的task_struct结构中, 如下所示 (include/linux/sched.h) :

```
struct task_struct { ..... unsigned long it_real_value, it_prof_value,  
it_virt_value. unsigned long it_real_incr, it_prof_incr, it_virt_incr.  
struct timer_list real_timer. .... } (1) 真实间隔定时器
```

(ITIMER_REAL) : 这种间隔定时器在启动后, 不管进程是否运行, 每个时钟滴答都将其间隔计数器减1。当减到0值时, 内核向进程发送SIGALRM信号。结构类型task_struct中的成员it_real_incr则表示真实间隔定时器的间隔计数器的初始值,

而成员it_real_value则表示真实间隔定时器的间隔计数器的当前值。由于这种间隔定时器本质上与上一节的内核定时器时一样的，因此Linux实际上是通过real_timer这个内嵌在task_struct结构中的内核动态定时器来实现真实间隔定时器ITIMER_REAL的。

(2) 虚拟间隔定时器ITIMER_VIRT：也称为进程的用户态间隔定时器。结构类型task_struct中成员it_virt_incr和it_virt_value分别表示虚拟间隔定时器的间隔计数器的初始值和当前值，二者均以时钟滴答次数为计数单位。当虚拟间隔定时器启动后，只有当进程在用户态下运行时，一次时钟滴答才能使间隔计数器当前值it_virt_value减1。当减到0值时，内核向进程发送SIGVTALRM信号（虚拟闹钟信号），并将it_virt_value重置为初值it_virt_incr。具体请见7.4.3节中的do_it_virt()函数的实现。

(3) PROF间隔定时器ITIMER_PROF：进程的task_struct结构中的it_prof_value和it_prof_incr成员分别表示PROF间隔定时器的间隔计数器的当前值和初始值（均以时钟滴答为单位）。当一个进程的PROF间隔定时器启动后，则只要该进程处于运行中，而不管是在用户态或核心态下执行，每个时钟滴答都使间隔计数器it_prof_value值减1。当减到0值时，内核向进程发送SIGPROF信号，并将it_prof_value重置为初值it_prof_incr。具体请见7.4.3节的do_it_prof()函数。

Linux在include/linux/time.h头文件中为上述三种进程间隔定时器定义了索引标识，如下所示：
#define ITIMER_REAL 0 #define ITIMER_VIRTUAL 1 #define ITIMER_PROF 2

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