

Linux系统进程间隔定时器Itimer(上) (3) 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\\_144905.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_B3_BB_E7_BB_c103_144905.htm) 当用户通过setitimer()系统调用来设置进程的ITIMER\_REAL间隔定时器时，it\_real\_incr被设置成非零值，于是该系统调用相应地设置好real\_timer.expires值，然后进程的real\_timer定时器就被加入到内核动态定时器链表中，这样该进程的ITIMER\_REAL间隔定时器就被启动了。当real\_timer定时器到期时，它的关联函数it\_real\_fn()将被执行。注意！所有进程的real\_timer定时器的function函数指针都指向it\_real\_fn()这同一个函数，因此it\_real\_fn()函数必须通过其参数来识别是哪一个进程，为此它将unsigned long类型的参数p解释为进程task\_struct结构的地址。该函数的源码如下

```
( kernel/itimer.c ) : void it_real_fn(unsigned long __data) { struct task_struct * p = (struct task_struct *) __data. unsigned long interval. send_sig(SIGALRM, p, 1). interval = p->it_real_incr. if (interval) { if (interval > (unsigned long) LONG_MAX) interval = LONG_MAX. p->real_timer.expires = jiffies interval. add_timer(&p->real_timer). } }
```

函数it\_real\_fn()的执行过程大致如下：（1）首先将参数p通过强制类型转换解释为进程的task\_struct结构类型的指针。（2）向进程发送SIGALRM信号。（3）在进程的it\_real\_incr非0的情况下继续启动real\_timer定时器。首先，计算real\_timer定时器的expires值为（jiffies + it\_real\_incr）。然后，调用add\_timer()函数将real\_timer加入到内核动态定时器链表中。7.7.3 itimer定时器的系统调用与itimer定时器相关的syscall有两个：getitimer()和setitimer()。

其中，`getitimer()`用于查询调用进程的三个间隔定时器的信息，而`setitimer()`则用来设置调用进程的三个间隔定时器。这两个syscall都是现在`kernel/itimer.c`文件中。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)