# Linux内核定时器

## 概念

时钟中断由系统的定时硬件以周期性的时间间隔产生，这个间隔（即频率）由内核根据HZ来确定，HZ是一个与体系结构无关的常数，可配置（50-1200），在

X86平台，默认值为1000。每当时钟中断发生时，全局变量jiffies(unsigned long)就加加1，因此jiffies记录了自linux启动后时钟中断发生的次数。驱动程序常利用jiffies来计算不同事件间的时间间隔。因此，对于HZ值为100的系统，1个jiffy等于10ms，而对于HZ为1000的系统，1个jiffy仅为1ms。

## 内核定时器

定时器用于控制某个函数数(定时器处理函数)在未来的某个特定时间执行。内核定时器注册的处理函数只执行一次--不是循环执行的。

内核定时器被组织成双向链表，并使用struct timer\_list结构描述。

struct timer\_list {

struct list\_head entry /\* 内核使用\*/;

unsigned long expires; /\*超 超 时的 的jiffies 值\*/

void (\*function)(unsigned long); /\*超 超 时 处 理 函数 数\*/

unsigned long data; /\*超 超 时 处 理 函 数 参数 数\*/

struct tvec\_base \*base; /\* 内核使用\*/

};

操作定时器的有如下函数：

void init\_timer(struct timer\_list \*timer);

初始化定时器队列结构 。

void add\_timer(struct timer\_list \* timer);

启动定时器 。

int del\_timer(struct timer\_list \*timer);

在定时器超时前将它删除。当定时器超时后，系统会自动地将它删除。