

Linux操作系统下的软中断问题分析 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\\_E6\\_93\\_8D\\_E4\\_BD\\_c103\\_144944.htm](https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E6_93_8D_E4_BD_c103_144944.htm) 前几天把硬中断完成了，这几天在做软中断的事情。现在对linux2.6.12的软中断机制有了一个基本的认识。在此把它的软中断的一些问题做一个记录。系统中有一个softirq\_action结构，它定义在中，是一个通用结构，还有一个很重要的结构在中申明，这个结构是cpu相关的，比如在i386中这个结构为 typedef struct {unsigned int \_\_softirq\_pending.unsigned long idle\_timestamp.unsigned int \_\_nmi\_count. /\* arch dependent \*/unsigned int apic\_timer\_irqs. /\* arch dependent \*/} \_\_\_\_cacheline\_aligned irq\_cpustat\_t.在arm中这个结构为 typedef struct {unsigned int \_\_softirq\_pending.} \_\_\_\_cacheline\_aligned irq\_cpustat\_t.以上两个结构均再中。在中有几个很不好理解的函数：1：local\_softirq\_pending()下面是这个函数的定义 CODE:#ifndef \_\_ARCH\_IRQ\_STATextern irq\_cpustat\_t irq\_stat[. /\* defined in asm/hardirq.h \*/#define \_\_IRQ\_STAT(cpu, member) (irq\_stat[cpu].member)#endif/\* arch independent irq\_stat fields \*/#define local\_softirq\_pending() \\_\_IRQ\_STAT(smp\_processor\_id(), \_\_softirq\_pending)其中的smp\_processor\_id()如果追下去会很复杂，其实它最后只是一个数字：cpu的号码，单cpu中为0.2.#define \_\_raise\_softirq\_irqoff(nr) do { local\_softirq\_pending() |= 1UL 这个函数实际上是把irq\_stat[cpu\_id](kernel/softirq.c中)结构中的pending的第nr为置位，表示这个cpu有几号软中断要处理，在do\_softirq()中有对irq\_stat[]的pending的相应检查及处理，详

细的处理过程参见源代码及《Linux Kernel Development》第7章。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)