

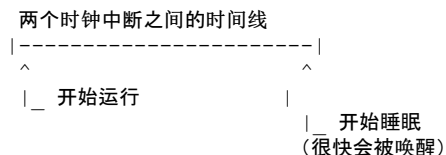
Linux通过`/proc/stat`和`/proc/uptime`导出各种信息, 用户空间工具 如top(1)使用这些信息计算系统花费在某个特定状态的平均时间。例如:

...

大多数情况下``proc/stat``的信息几乎真实反映了系统信息,然而,由于内核采集这些数据的方式/时间特点,有时这些信息根本不可靠。

举例 ---

假设系统有一个进程以如下方式周期性地占用cpu:



在上面的情况下，根据`/proc/stat`的信息(由于当系统处于空闲状态时，时间中断经常会发生)系统的负载将会是0。大家能够想象内核的这种行为会发生在许多情况下，这将导致`/proc/stat`中存在相当古怪的信息：

```

/* gcc -o hog smallhog.c */
#include <time.h>
#include <limits.h>
#include <signal.h>
#include <sys/time.h>
#define HIST 10

static volatile sig_atomic_t stop;

static void sighandler (int signr)
{
    (void) signr;
    stop = 1;
}

static unsigned long hog (unsigned long niters)
{
    stop = 0;
    while (!stop && --niters);
    return niters;
}

int main (void)
{
    int i;
    struct itimerval it = { .it_interval = { .tv_sec = 0, .tv_usec = 1 },
                           .it_value = { .tv_sec = 0, .tv_usec = 1 } };
    sigset_t set;
    unsigned long v[HIST];
    double tmp = 0.0;
    unsigned long n;
    signal (SIGALRM, &sighandler);
    setitimer (ITIMER_REAL, &it, NULL);

    hog (ULONG_MAX);
    for (i = 0; i < HIST; ++i) v[i] = ULONG_MAX - hog (ULONG_MAX);
    for (i = 0; i < HIST; ++i) tmp += v[i];
    tmp /= HIST;
    n = tmp - (tmp / 3.0);

    sigemptyset (&set);
    sigaddset (&set, SIGALRM);

```

```
for (;;) {  
    hog (n);  
    sigwait (&set, &i);  
}  
return 0;  
}
```

参考 ---

- <https://lore.kernel.org/r/loom20070212T063225-663@post.gmane.org>
- Documentation/filesystems/proc.rst (1.8)

谢谢 ---

Con Kolivas, Pavel Machek