Linux操作系统中关于负载的定义 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/242/2021_2022_Linux_E6_93_8D_E4_BD_c103_242686.htm 使用uptime或者top命令，都可以看到一个负载的输出，形如load average: 0.00, 0.03, 0.00，这个负载到底是什么东西呢，man文档里只是一笔带过，没有具体的给出负载的定义。 负载的统计，必然是由内核完成的，因此在内核源码中找答案是再好不过的事情了，找来2.6.21的内核源码，开始探索。 节选部分源码：//kernel/timer.c1254 active_tasks = count_active_tasks().1256 CALC_LOAD(avenrun[0], EXP_1, active_tasks).1257 CALC_LOAD(avenrun[1], EXP_5, active_tasks).1258 CALC_LOAD(avenrun[2], EXP_15, active_tasks).//include/linux/sched.h 110 #define FSHIFT 11 /* nr of bits of precision */ 111 #define FIXED_1 (1 112 #define LOAD_FREQ (5*HZ) /* 5 sec intervals */ 113 #define EXP_1 1884 /* 1/exp(5sec/1min) as fixed-point */ 114 #define EXP_5 2014 /* 1/exp(5sec/5min) */ 115 #define EXP_15 2037 /* 1/exp(5sec/15min) */ 117 #define CALC_LOAD(load,exp,n) \ 118 load *= exp. \ 119 load = n*(FIXED_1-exp). \ 120 load >>= FSHIFT. load(t) = (load(t-1)*exp(i) n(t)*(2048-exp(i))) / 2048 load(t-1)为上次计算出的结果 n(t)为t时刻的活动进程数 计算方式是累加各个CPU的运行队列中running和uninterruptible的值 再乘以2048 计算方式如下： 1946 unsigned long nr_active(void) 1947 { 1948 unsigned long i, running = 0, uninterruptible = 0. 1949 1950 for_each_online_cpu(i) { 1951 running = cpu_rq(i)->nr_running.

1952 uninterruptible = cpu_rq(i)->nr_uninterruptible. 1953 } 1954 1955 if (unlikely((long)uninterruptible 1226 static unsigned long count_active_tasks(void) 1227 { 1228 return nr_active() * FIXED_1. 1229 } exp(1) = 1884 exp(5) = 2014 exp(15) = 2037 exp(i) = 2048 * e^(-1/12/i) 从本质上看负载是完全由过去的一段时间里每个CPU上的活动进程数决定的，但并不是在数值上等同于每秒钟需要进行调度的进程数，具体的计算过程是个比较复杂的过程。 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