

Linux内核的一些算法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022_Linux_E5_86_85_E6_A0_c103_252964.htm

1.Linux 调度 I/O消耗型和处理器消耗型进程（保证及时的IO响应和高效的CPU利用率）负载平衡：每颗CPU上有一个进程队列，当CPU的负载差大于25%时运行负载平衡程序。（从最忙的CPU上取下一个优先级最高且CPU亲和力最小的进程到负载最轻的CPU进程队列中）。为什么要取优先级最高的进程呢？因为把优先级高的进程平均分配到各个CPU上去才能使系统的性能达到最高。CPU的亲和力：主要是高速缓存或是其它原因，如某个进程只能在某个CPU上运行。

2.中断处理程序与线程的不同之处。

- 1、中断中不能睡眠。
- 2、中断上下文具有较严格的时间限制，因为它打断了正在运行的线程。
- 3、中断处理程序不具有自己的栈，它共享被中断线程的栈，因此在分配一个任务栈要注意比实际需求的大一点，以防中断发生时，中断程序使用栈时溢出。

3.自旋锁：主要特点，与等待一个信号量不同的是它在得不到想要的资源时进程不会睡眠。

4.引入读写自旋锁。读写自旋锁减少了锁的粒度。

5.slab层：优化动态内存分配，（将相同大小的动态内存分配和释放放到一个连续的内存区，这样就不会产生内存碎片，因为每次分配刚好是一个单位，每次释放也刚好是一个单位，比如用在进程控制块的动态分配与释放中）。

6.磁盘块I/O操作：使用Linux电梯算法，在磁盘底层操作时将磁盘块IO请求按磁道排序，以减少磁头寻迹时间而优化磁盘IO操作。

7.使用页高速缓存和页回写，优化文件系统。

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