

Linux高精度的时序 (sleep , usleep , nanosleep) Linux认证考试 PDF转换可能丢失图片或格式 , 建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Linux_E9_AB_98_E7_B2_c103_645332.htm 首先, 我会说不保证你在使用者模式 (user-mode) 中执行的行程 (process) 能够精确地控制时序因为 Linux 是个多工的作业环境. 你在执行中的行程 (process) 随时会因为各种原因被暂停大约 10 毫秒到数秒 (在系统负荷非常高的时候). 然而, 对于大多数使用 I/O 埠的应用而言, 这个延迟时间实际上算不了什麼. 要缩短延迟时间, 你得使用函式 nice 将你在执行中的行程 (process) 设定成高优先权(请参考 nice(2) 使用说明文件) 或使用即时排程法 (real-time scheduling) (请看下面). 如果你想获得比在一般使用者模式 (user-mode) 中执行的行程 (process) 还要精确的时序, 有一些方法可以让你在使用者模式 (user-mode) 中做到`即时排程的支援. Linux 2.x 版本的核心中有软体方式的即时排程支援. 详细的说明请参考 sched_setscheduler(2) 使用说明文件. 有一个特殊的核心支援硬体的即时排程. (Sleeping) : sleep() 与 usleep() 现在, 让我们开始较简单的时序函式呼叫. 想要延迟数秒的时间, 最佳的方法大概是使用函式 sleep(). 想要延迟至少数十毫秒的时间 (10 ms 似乎已是最短的 延迟时间了), 函式 usleep() 应该可以使用. 这些函式是让出 CPU 的使用权 给其他想要执行的行程 (processes) (`自己休息去了), 所以没有浪费掉 CPU 的时间. 细节请参考 sleep(3) 与 usleep(3) 的说明文件. 如果让出 CPU 的使用权因而使得时间延迟了大约 50 毫秒 (这取决于处理器与机器的速度, 以及系统的负荷), 就浪费掉 CPU 太多的时间, 因为 Linux 的排程器 (scheduler) (单就 x86 架构而言) 在将控制权发

还给你的行程 (process) 之前通常至少要花费 10-30 毫秒的时间. 因此, 短时间的延迟, 使用函式 `usleep(3)` 所得到的延迟结果通常会大於你在参数所指定的值, 大约至少有 10 ms.

`nanosleep()` 在 Linux 2.0.x 一系列的核心发行版本中, 有一个新的系统呼叫 (system call), `nanosleep()` (请参考 `nanosleep(2)` 的说明文件), 他让你能够 休息或延迟一个短的时间 (数微秒或更多). 如果延迟的时间 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