Linux下如何实现秒以下精确定时与休眠 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/462/2021_2022_Linux_E4_B8_8B_E5_A6_c103_462197.htm Linux 中提供的休眠函数是sleep和alarm ，但是他们仅仅提供以秒为单位的休眠，这中休眠有些进程显然太长了，那么怎样才能使进程以更小的时间分辨率休眠呢？ 我知道的方法有2种，下面就做分别介绍。 第一种方法是使用定时器，Linux提供的定时器函数是： int setitimer(int which, const struct itimerval *value, struct itimerval *ovalue). which指定那种定时器。Linux提供3种定时器： TIMER_REAL: 准确定时器，超时会发出SIGALRM信号； TIMER_VIRTUAL: 虚拟定时器，只记进程时间，所以会根据进程执行时间而变化，不能实现准确定时，超时发出SIGVTALRM信号； TIMER_PROF: 梗概计时器，它会根据进程时间和系统时间而变化，不能实现准确定时，超时发出SIGPROF信号； 在进程中应该捕捉所设定时器会发出的信号，因为进程收到定时器超时发出的信号后，默认动作是终止。 value是设置定时器时间，相关结构如下： struct itimerval { struct timeval it_interval. struct timeval it_value. }. struct timeval { long tv_sec. long tv_usec. }. it_interval指定间隔时间，it_value指定初始定时时间。如果只指定it_value，就是实现一次定时；如果同时指定it_interval，则超时后，系统会重新初始化it_value为it_interval，实现重复定时；两者都清零，则会清除定时器。 tv_sec提供秒级精度，tv_usec提供微秒级精度，以值大的为先，注意1s = 1000000ms。 ovalue用来保存先前的值，常设为NULL。 如果是以setitimer提供的定时器来休眠，只

需阻塞等待定时器信号就可以了。 第二种方法是使用0select来提供精确定时和休眠： int 0select(int n, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout). n指监视的文件描述符范围，通常设为所要0select的fd＋1，readfds，writefds和exceptfds分别是读，写和异常文件描述符集，timeout为超时时间。 可能用到的关于文件描述符集操作的宏有： FD_CLR(int fd, fd_set *set). 清除fd FD_ISSET(int fd, fd_set *set). 测试fd是否设置 FD_SET(int fd, fd_set *set). 设置fd FD_ZERO(fd_set *set). 清空描述符集 我们此时用不到这些宏，因为我们并不关心文件描述符的状态，我们关心的是0select超时。所以我们需要把readfds，writefds和exceptfds都设为NULL，只指定timeout时间就行了。至于n我们可以不关心，所以你可以把它设为任何非负值。实现代码如下： int msSleep(long ms) { struct timeval tv. tv.tv_sec = 0. tv.tv_usec = ms. return 0select(0, NULL, NULL, NULL, &amp.tv). } 呵呵，怎么样，是不是很简单？ 结语： setitimer和0select都能实现进程的精确休眠，本文分别对他们进行了简单介绍，并给出了一个简单的给予0select的实现。我不推荐使用setitimer，因为一者Linux系统提供的timer有限（每个进程至多能设3个不同类型的timer），再者ssetitimer实现起来没有0select简单。 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