解析Linux内核获取当前进程指针的方法（1） PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/144/2021_2022__E8_A7_A3_E6_9E_90Linu_c103_144935.htm 一、内存数据表示： 我们在教材或阅读中，经常需要直观的用图示来展示数据在内存中的分布，那么数据是如何在内存中组织的呢？不同的机器有不同的表示法，我们以最常见的Intel X86系列计算机为例来说明这个问题。 如上图示内存示意图：内存低址在上。内存高址在下，内存单位为16bit。对于基于intel i386架构的计算机，系统采用小端字节序来存放数据，所谓小端字节序是指低序字节低地址，高序字节高地址(内存地址增大方向)，大端字节序反之，给定系统所用的字节序称为主机字节序；CPU也以小端字节序形式读取数据，如上图所示，如果变量num是16位的short短整类型，则CPU从内存中读出的num=0x1234；如果num是32位的int类型，则CPU从内存中读出的是num=0x56781234,其中num地址是0x12345678,即&amp.num=0x12345678。 二、linux内核获取进程任务结构的指针 明白了系统内存数据表示，我们现在来看看linux内核是如何获取当前进程的任务结构指针的，以下代码均参照linux内核2.4.0的源码。 在include\asm-i386\ current.h中 #ifndef _I386_CURRENT_H#define _I386_CURRENT_Hstruct task_struct.static inline struct task_struct * get_current(void){struct task_struct *current.__asm__("andl %%esp,%0. ":"=r" (current) : "0" (~8191UL)).return current.}#define current get_current()#endif /* !(_I386_CURRENT_H) */ 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