Linux进程内核栈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_E8_BF_9B_E7_A8_c103_645115.htm 在内核2.4中堆栈是这么定义的： union task_union { struct task_struct task. unsigned long stack[INIT_TASK_SIZE/sizeof(long)]. }. 而INIT_TASK_SIZE只能是8K。 内核为每个进程分配一个task_struct结构时，实际上分配两个连续的物理页面(8192字节)，如图所示。底部用作task_struct结构(大小约为1K字节)，结构的上面用作内核堆栈(大小约为7K字节)。访问进程自身的task_struct结构，使用宏操作 current, 在2.4中定义如下： #define current get_current() static inline struct task_struct * get_current(void) { struct task_struct *current. __asm__("andl %%esp,%0. ":"=r" (current) : "" (~8191UL)). return current. } ~8191UL表示最低13位为0, 其余位全为1。 %esp指向内核堆栈中，当屏蔽掉%esp的最低13后，就得到这个 ” 两个连续的物理页面 ” 的开头，而这个开头正好是task_struct的开始，从而得到了指向task_struct的指针。 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