内核态和用户态的区别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_ti2020/642/2021_2022__E5_86_85_E6_A0_B8_E6_80_81_E5_c103_642571.htm 内核态与用户态是操作系统的两种运行级别，intel cpu提供Ring0-Ring3三种级别的运行模式。Ring0级别最高，Ring3最低。 当一个任务(进程)执行系统调用而陷入内核代码中执行时，我们就称进程处于内核运行态(或简称为内核态)。此时处理器处于特权级最高的(0级) 内核代码中执行。当进程处于内核态时，执行的内核代码会使用当前进程的内核栈。每个进程都有自己的内核栈。当进程在执行用户自己的代码时，则称其处于用户运行态(用户态)。即此时处理器在特权级最低的(3级)用户代码中运行。 在内核态下CPU可执行任何指令，在用户态下CPU只能执行非特权指令。当CPU处于内核态，可以随意进入用户态；而当CPU处于用户态时，用户从用户态切换到内核态只有在系统调用和中断两种情况下发生，一般程序一开始都是运行于用户态，当程序需要使用系统资源时，就必须通过调用软中断进入内核态。 Linux使用了Ring3级别运行用户态，Ring0作为内核态，没有使用Ring1和Ring2。Ring3状态不能访问Ring0的地址空间，包括代码和数据。Linux进程的4GB地址空间，3G-4G部分大家是共享的，是内核态的地址空间，这里存放在整个内核的代码和所有的内核模块，以及内核所维护的数据。用户运行一个程序，该程序所创建的进程开始是运行在用户态的，如果要执行文件操作，网络数据发送等操作，必须通过 write，send等系统调用，这些系统调用会调用内核中的代码来完成操作，这时，必须切换到Ring0，然后进

入3GB-4GB中的内核地址空间去执行这些代码完成操作，完成后，切换回Ring3，回到用户态。这样，用户态的程序就不能随意操作内核地址空间，具有一定的安全保护作用。