探究在Linux中添加新的系统调用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__E6_8E_A2_E7_A9_B6_E5_9C_A8L_c103_645224.htm 系统调用是应用程序和操作系统内核之间的功能接口。其主要目的是使得用户可以使用操作系统提供的有关设备管理、输入/输入系统、文件系统和进程控制、通信以及存储管理等方面的功能，而不必了解系统程序的内部结构和有关硬件细节，从而起到减轻用户负担和保护系统以及提高资源利用率的作用。 Linux操作系统作为自由软件的代表，它优良的性能使得它的应用日益广泛，不仅得到专业人士的肯定，而且商业化的应用也是如火如荼。在 Linux中，大部分的系统调用包含在Linux的libc库中，通过标准的C函数调用方法可以调用这些系统调用。那么，对Linux的发烧友来说，如何在Linux中增加新的系统调用呢？ 1 Linux系统调用机制 在Linux系统中，系统调用是作为一种异常类型实现的。它将执行相应的机器代码指令来产生异常信号。产生中断或异常的重要效果是系统自动将用户态切换为核心态来对它进行处理。这就是说，执行系统调用异常指令时，自动地将系统切换为核心态，并安排异常处理程序的执行。 Linux用来实现系统调用异常的实际指令是： Int ?$0x80 这一指令使用中断/异常向量号128(即16进制的80)将控制权转移给内核。为达到在使用系统调用时不必用机器指令编程，在标准的C语言库中为每一系统调用提供了一段短的子程序，完成机器代码的编程工作。事实上，机器代码段非常简短。它所要做的工作只是将送给系统调用的参数加载到CPU寄存器中，接着执行int ?$0x80指令。然后运行系统调

用，系统调用的返回值将送入CPU的一个寄存器中，标准的库子程序取得这一返回值，并将它送回用户程序。 为使系统调用的执行成为一项简单的任务，Linux提供了一组预处理宏指令。 它们可以用在程序中。这些宏指令取一定的参数，然后扩展为调用指定的系统调用的函数。 这些宏指令具有类似下面的名称格式： _syscallN(parameters) 其中N是系统调用所需的参数数目，而parameters则用一组参数代替。这些参数使宏指令完成适合于特定的系统调用的扩展。例如，为了建立调用setuid()系统调用的函数，应该使用： _syscall1(int, setuid， uid_t， uid) syscallN( )宏指令的第1个参数int说明产生的函数的返回值的类型是整型，第2个参数setuid说明产生的函数的名称。后面是系统调用所需要的每个参数。这一宏指令后面还有两个参数uid_t和uid分别用来指定参数的类型和名称。 另外，用作系统调用的参数的数据类型有一个限制，它们的容量不能超过四个字节。这是因为执行int ?$0x80指令进行系统调用时，所有的参数值都存在32位的CPU寄存器中。使用CPU寄存器传递参数带来的另一个限制是可以传送给系统调用的参数的数目。这个限制是最多可以传递5个参数。所以Linux一共定义了6个不同的_syscallN()宏指令，从_syscall0()、 _syscall1()直到_syscall5()。 一旦_syscallN()宏指令用特定系统调用的相应参数进行了扩展，得到的结果是一个与系统调用同名的函数，它可以在用户程序中执行这一系统调用。 2 添加新的系统调用 如果用户在Linux中添加新的系统调用，应该遵循几个步骤才能添加成功，下面几个步骤详细说明了添加系统调用的相关内容。 (1) 添加源代码 第一个任务是编写加到内核中的源程序，即将要加到一个内核文件中去的一个

函数，该函数的名称应该是新的系统调用名称前面加上sys_标志。假设新加的系统调用为mycall(int number)，在/usr/src/linux/kernel/sys.c文件中添加源代码，如下所示：asmlinkage int sys_mycall(int number) { return number. } 作为一个最简单的例子，我们新加的系统调用仅仅返回一个整型值。(2) 连接新的系统调用 添加新的系统调用后，下一个任务是使Linux内核的其余部分知道该程序的存在。为了从已有的内核程序中增加到新的函数的连接，需要编辑两个文件。 在我们所用的Linux内核版本(RedHat 6.0，内核为2.2.5-15)中，第一个要修改的文件是： /usr/src/linux/include/asm-i386/unistd.h 该文件中包含了系统调用清单，用来给每个系统调用分配一个唯一的号码。文件中每一行的格式如下： #define __NR_name NNN 其中，name用系统调用名称代替，而NNN则是该系统调用对应的号码。应该将新的系统调用名称加到清单的最后，并给它分配号码序列中下一个可用的系统调用号。我们的系统调用如下： #define __NR_mycall 191 系统调用号为191，之所以系统调用号是191，是因为Linux-2.2内核自身的系统调用号码已经用到190。 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