**Lab 2 系统调用**

1. 从Linux 0.11现在的机制看，它的系统调用最多能传递几个参数？你能想出办法来扩大这个限制吗？

include/unistd.h 中的 \_syscall 可知，应用函数经过库函数向内核发送了一个0x80的中断，然后开始执行系统调用，在Linux 0.11中是使用ebx ecx edx来传递参数的，所以在linux 0.11中最多能够传递3个参数。

为了解决这个问题，我们可以传值的时候传给函数的是一个指针，所有的数据是由这个指针所指向的，这样能传递的参数就不会受到限制了，通过寄存器的间接寻址就可以向系统调用传递多个参数了。

1. 用文字简要描述向Linux 0.11添加一个系统调用foo()的步骤。
   1. 修改include/linux/sys.h ，在里面加上新系统调用的宏定义。即仿照系统调用格式添加extern rettype sys\_foo(); 然后在sys\_call\_tab数组中加上sys\_foo.
   2. 修改 include/unistd.h，在里面面添加系统调用号 #define \_\_NR\_foo 72(后面的72是假设原来的系统就是纯净的Linux 0.11，实际需按照情况填写)
   3. 修改 kernel/system\_call.s, 改变系统的总调用个数 nr\_system\_calls = 73，在原值上加一
   4. kernel/foo.c 写系统调用函数
   5. 修改makefile 让他在编译的时候对新添加的系统调用函数也进行编译