**Lab2 系统调用**

1. 回答问题
2. 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？

答：它的系统调用最多能传递3个参数。因为寄存器的种类疏朗限制了参数的传递。

由于寄存器eax存放了系统调用功能号，所以只有ebx、ecx、edx三个寄存器来传递系统调用过程中要用到的参数 。

解决方法：eax，ebx，ecx都是32位的通用寄存器，都存在低16位和高16位，可以将上述寄存器的高位和低位分开使用，那么参数的传递就可以扩展到6个。

1. 用文字简要描述向Linux 0.11添加一个系统调用foo()的步骤。

这个步骤和实验2的步骤基本一致。

首先是要在在内核中（linux-0.11/kernel）编写foo()对应的系统调用处理函数sys\_foo();

然后在include/unistd.h中添加系统调用foo()的功能号（#define \_\_NR\_foo \*\*）；这之后还要挂载在HDC中更改unistd.h文件。

在kernel/system\_call.s中修改系统调用总数。

在include/linux/sys.h中声明新的系统调用处理函数sys\_foo()以及添加系统调用处理函数指针数组表中sys\_foo()的索引值。

在Makefile中添加系统调用处理函数sys\_foo()所在文件的编译、链接规则（主要是依赖关系）。

在用户应用程序中提供系统调用foo()。

之后系统调用便可添加完毕，在linux0.11目录下，./run之后即可调用内核中的函数。挂载后可进行修改