

为 linux 内核添加新的系统调用

开发平台：x86 ubuntu

目标平台：S3C6410 linux3.4.4

1、打开内核源码目录下 arch/arm/kernel/calls.S 文件，在 389 行添加：

```
/* 378 */ CALL(sys_my_syscall)
```

378 是新添加的系统调用的系统调用号偏移，sys_my_syscall 是该系调用的名字

二、打开内核源码目录下 arch/arm/include/asm/unistd.h 文件，在 407 行添加：

```
#define __NR_my_syscall    (__NR_SYSCALL_BASE+378)
```

my_syscall 是新系统调用的名字，378 是系统调用号

三、打开内核源码目录下 arch/arm/kernel/sys_arm.c 文件，在最后添加：

```
asmlinkage long sys_my_syscall(int a, int b)
{
    return a + b;
}
```

这个就是系统调用的实现，在这里简单的写一个加法为例

四、重新编译内核，启动内核并挂在网络文件系统，挂在成功后：

```
[lzy@uplooking]#
```

五、在用户态测试新添加的系统调用 my_syscall

```
#include <stdio.h>
```

```
int main(int argc, char *argv[])
{
    int add;

    add = syscall(378, 1, 2);

    printf("add = %d\n", add);

    return 0;
}
```

其中 378 是新添加的系统调用的系统调用号

编译并把可执行文件拷贝到/nfsroot，/nfsroot 是开发板的根

```
lizhiyong@ThinkPad:/home/test$ arm-linux-gcc my_syscall.c -o my_syscall
```

```
lizhiyong@ThinkPad:/home/test$ cp my_syscall /nfsroot
```

在开发板上执行 my_syscall，运行结果如下：

```
[lzy@uplooking]# ./my_syscall
add=3
```