

SysCall

syscall的过程

1.系统调用号放入a7寄存器

2.ecall跳转到syscall, 执行此函数

3. syscall函数取出a7寄存器中的系统调用号, 然后执行sys_trace(), 这里将mask已经读取并保存到proc结构体中了, 并将结果存入到a0

4. 打印syscall的结果, 需自己添加

调用并保存结果

```
p->trapframe->a0 = syscalls[num]();
```

实验步骤

1.user/user.h中添加trace()函数

2.user/usys.pl中添加trace的entry, 编译时会自动在.S文件中添加trace的汇编代码

3.kernel/syscall.h中定义trace的系统调用编码

4.kernel/sysproc.c中添加sys_trace()函数

5.kernel/proc.h中的proc添加mask掩码 (掩码是1 << 系统调用编码的位运算结果, 用十进制来表示);

6.从用户空间接收用户参数, 参考kernel/syscall.c以及kernel/sysproc.c

7.修改fork(), 复制mask

8.修改syscall(), 声明一个字符串数组来记录名字方便打印, 并且syscall来打印调用的数组;