

Exit.c

37d36

```
< #include <linux/types.h>
707,708d705
<     struct list_head *list ;
<     struct task_struct *process, *temp ;
711,723d707
<
<     if(current->myFlag == 1 && task_nice(current) > 10){
<         printk(KERN_ALERT "\n # static prio = %d \n task_nice = %d ",current-
>static_prio,task_nice(current));
<         printk(KERN_ALERT "\n # PARENT %d is going to exit which has myFlag=1
\n",current->pid); // or use getpid()
<         process = current ;
<         list_for_each(list,&process->children){
<             temp = list_entry(list,struct task_struct,sibling);
<             printk(KERN_ALERT "\n # CHILD %d is going to be killed \n",temp-
>pid); // or use getpid()
<             //do_send_sig_info(SIGKILL,SEND_SIG_FORCED,temp,true);
<             kill_pid(find_vpid(temp->pid),SIGKILL,1);
<         }
<     }
<
<
```

Fork.c

1664,1667d1663

```
<
<     p->myFlag = 0;
<     printk(KERN_ALERT "### myFlag = %d",p->myFlag);
<
```

init_task.h

164d163

```
<     .myFlag = 0, \
```

Makefile

540c540

```
< core-y          := usr/ mycall/ set_myFlag/  
---  
> core-y          := usr/
```

sched.h

```
1468d1467  
< int myFlag;
```

syscall_32.tbl

```
364,365d363  
< 355 i386 mycall sys_mycall  
< 356 i386 set_myFlag sys_set_myFlag
```

syscalls.h

```
852,853d851  
< asmlinkage int sys_mycall(int flag);  
< asmlinkage long set_myFlag(pid_t pid,int flag);
```