# 一、异步通知简介

##  1、硬件中断

##  2、信号

 软件层次上的“中断”，也叫做软中断信号，软件层次上对中断机制的一种模拟

kill -9 xxx //关闭某个应用

##  3、信号处理函数

 应用使用signal函数，设置信号处理函数，原型为：

sighandler\_t signal(int signum, sighandler\_t handler);

 信号处理函数为：

typedef void (\*sighandler\_t)(int);

CTRL+C关闭应用，发送的SIGINT信号。

谁向应用发送信号。

##  4、驱动中对异步通知的处理

 需要实现file\_operations中的fasync函数，函数原型如下：

int (\*fsync) (struct file \*, loff\_t, loff\_t, int datasync);

 1、要使用fasync\_struct定义一个指针结构体变量。

 2、实现file\_operations里面的fasync函数，函数原型：

int (\*fasync) (int, struct file \*, int);

 fasync还需要借助fasync\_helper函数。

 3、驱动里面调用fasync向应用发送信号，函数原型

void kill\_fasync(struct fasync\_struct \*\*fp, int sig, int band)

 4、关闭驱动的时候要删除信号：

# 二、编写试验驱动

# 三、测试