内容目录

[一 输入系统\_必备Linux编程知识 1](#__RefHeading___Toc248_2054555503)

# 一 输入系统\_必备Linux编程知识

(1) 键盘即插即用=》怎么检测键盘的接入与拔出

1) hotplug: 内核发现键盘插入/拔出=》启动hotplug进程=》发送消息给输入系统；ps: android没有用这个方式

2) inotify: 输入系统使用inotify检测目录/dev/input

(2) 对于多键盘，使用的是epoll

## 1.1 inotify和epoll

### 1.1.1 inotify

监测目录 / 文件的变化，参考代码: frameworks\native\services\inputflinger\EventHub.cpp

APP\_0006\_inotify\_epoll/inotify.c

gcc -o inotify inotify.c

mkdir tmp

./inotify tmp &

echo > tmp/1

echo > tmp/2

rm tmp/1 tmp/2

### 1.1.2 epoll

检测多个文件有无数据供读出，有无空间供写入

APP\_0006\_inotify\_epoll/epoll.c

gcc -o epoll epoll.c

mkdir tmp

mkfifo tmp/1 tmp/2 tmp/3

./epoll tmp/1 tmp/2 tmp/3 &

echo aaa > tmp/1

echo bbb > tmp/2

使用fifo是，我们的epoll程序是reader

echo aa > tmp/1 是writer

a. int tmpFd = open(argv[i], O\_RDONLY|O\_NONBLOCK);//会导致无限循环

如果reader以 O\_RDONLY|O\_NONBLOCK打开FIFO文件，

当writer写入数据时，epoll\_wait会立刻返回；

当writer关闭FIFO之后，reader再次调用epoll\_wait，它也会立刻返回(原因是EPPLLHUP，描述符被挂断)

b. int tmpFd = open(argv[i], O\_RDWR);

如果reader以 O\_RDWR打开FIFO文件

当writer写入数据时，epoll\_wait会立刻返回；

当writer关闭FIFO之后，reader再次调用epoll\_wait，它并不会立刻返回，而是继续等待有数据

### 1.1.3 inotify + epoll

编写 inotify\_epoll.c，用它来监测tmp/目录: 有文件被创建/删除，有文件可读出数据

a. 当在tmp/下创建文件时，会立刻监测到，并且使用epoll监测该文件

b. 当文件有数据时，读出数据

c. 当tmp/下文件被删除时，会立刻监测到，并且把它从epoll中移除不再监测

inotify\_epoll.c

gcc -o inotify\_epoll inotify\_epoll.c

mkdir tmp

./inotify\_epoll tmp/ &

mkfifo tmp/1 tmp/2 tmp/3

echo aaa > tmp/1

echo bbb > tmp/2

rm tmp/3

## 1.2 socketpair

APP\_0007\_socketpair\_binder/socketpair.c，亲缘关系是指fork

