

```
1. void sysfs_notify(struct kobject *kobj, const char *dir, const char *attr);
2. void sysfs_notify_dirent(struct sysfs_dirent *sd);
```

用来唤醒在读写属性文件(sysfs节点)时因调用select()或poll()而阻塞的用户进程

```
1. int main()
2. {
3.     int hw_fd;
4.     int err;
5.     struct pollfd pfd;
6.     int value = 0;
7.
8.     hw_fd = open("/sys/devices/platform/hw_ps/install", O_RDONLY); //sysfs节点
9.     if (hw_fd < 0)
10.         return -EFAULT;
11.
12.     pfd.fd = hw_fd;
13.
14.     while (1) {
15.         pfd.revents = 0;
16.         err = poll(&pfd, 1, -1); //用户进程调用poll操作时会睡眠，等待内核空
间代码调用sysfs_notify()唤醒
17.         if (err) {
18.             err = read(hw_fd, &value, 1);
19.             if(err < 0)
20.                 return -ERR;
21.             if(value != 0)
22.                 break;
23.         }
24.     }
25.     close(hw_fd);
26. }
```

内核空间代码在某处调用sysfs_notify()用来唤醒用户进程：

```
sysfs_notify(&(pedv->dev.kobj), NULL, "install");
```

sysfs_notify()第一个参数是内核调用sysfs_create_group/sysfs_create_file创建sysfs节点时的struct kobject对象；

第三个参数是节点的名字；

第二个参数暂时未知，常用NULL。

<http://lwn.net/Articles/174660/>