poll提供的功能与select类似，不过在处理流设备时，它能够提供额外的信息。

poll(struct pollfd fd[], nfds\_t nfds, int timeout)

第一个参数:一个结构数组

struct pollfd{

int fd; //文件描述符

short events; //请求的事件

short revents; //返回的事件

};

fd：是一个struct pollfd结构类型的数组，用于存放需要检测其状态的Socket描述符；每当调用这个函数之后，系统不会清空这个数组，操作起来比较方便；特别是对于socket连接比较多的情况下，在一定程度上可以提高处理的效率；这一点与select()函数不同，调用select()函数之后，select()函数会清空它所检测的socket描述符集合，导致每次调用select()之前都必须把socket描述符重新加入到待检测的集合中；因此，select()函数适合于只检测一个socket描述符的情况，而poll()函数适合于大量socket描述符的情况；

events包括要监视的事件。

poll用已经发生的事件填充revents。

poll函数的事件标志符值：



epoll()函数的用法

epoll\_wait(int epfd, struct epoll\_event \*events,int maxevents, int timeout);

功能：开始epoll监听，将就绪的事件放到events事件数组中

epfd：epoll\_create返回的epoll描述符

events：用于存放就绪的事件

maxevents：同时能够处理的就绪事件最大个数，跟events数组大小相同

timeout: epoll超时时间（毫秒）

需要使用:

epoll\_create函数。该函数生成一个epoll专用的文件描述符，其中的参数是指定生成描述符的最大范围。

epoll\_ctl函数。该函数用于控制某个文件描述符上的事件，可以注册事件，修改事件，删除事件。

epoll\_wait函数:该函数用于轮询I/O事件的发生。