0326_poll

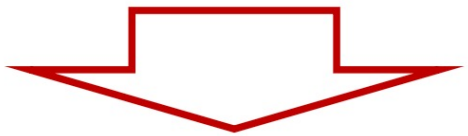
select的优缺点：

优点：（任何的多路转接方案都有这些优点）

1. 效率高（对比之前的，在多路转接这里，select只是弟弟） -> 因为他一直在HandlerEvent
2. 应用场景：有大量链接，但是只有少量的活跃的

缺点：

1. 为了维护第三方数组，所以select服务器会充满遍历操作
2. 每一次到要对select输出参数进行重新设定
3. select能够同时管理的fd个数是有上限的！
4. 因为几乎每一个参数都是输入输出型的，select一定会频繁的用户到内核，内核到用户的参数数据拷贝
5. 编码比较复杂



POLL

poll做了哪些改进呢？

1. 输入输出参数分离了
2. poll监管的文件描述符不再有上限

```
SYNOPSIS
       #include <poll.h>

       int poll(struct pollfd *fds, nfds_t nfds, int timeout);
```

timeout其实就是
毫秒为单位

传1000，就等1000
传0，就非阻塞等
传-1，就阻塞等
很简单，比select的好用

poll只负责等！
1. 用户告诉内核：你要帮我关心，哪些fd，哪些事件
2. 内核告诉用户：哪些fd，哪些事件已经就绪了

```
struct pollfd {
    int   fd;         /* file descriptor */
    short events;     /* requested events */
    short revents;    /* returned events */
};
```

用户告诉内核的时候，设置events就行了
如果就绪了，内核通知用户的时候，设置revents

事　件	描　述	是否可作为输入	是否可作为输出
POLLIN	数据（包括普通数据和优先数据）可读	是	是
POLLRDNORM	普通数据可读	是	是
POLLRDBAND	优先级带数据可读（Linux 不支持）	是	是
POLLPRI	高优先级数据可读，比如 TCP 带外数据	是	是
POLLOUT	数据（包括普通数据和优先数据）可写	是	是
POLLWRNORM	普通数据可写	是	是
POLLWRBAND	优先级带数据可写	是	是
POLLRDHUP	TCP 连接被对方关闭，或者对方关闭了写操作，它由 GNU 引入	是	是
POLLERR	错误	否	是
POLLHUP	挂起。比如管道的写端被关闭后，读端描述符上将收到 POLLHUP 事件	否	是
POLLNVAL	文件描述符没有打开	否	是

这三个就是select对应的三个参数

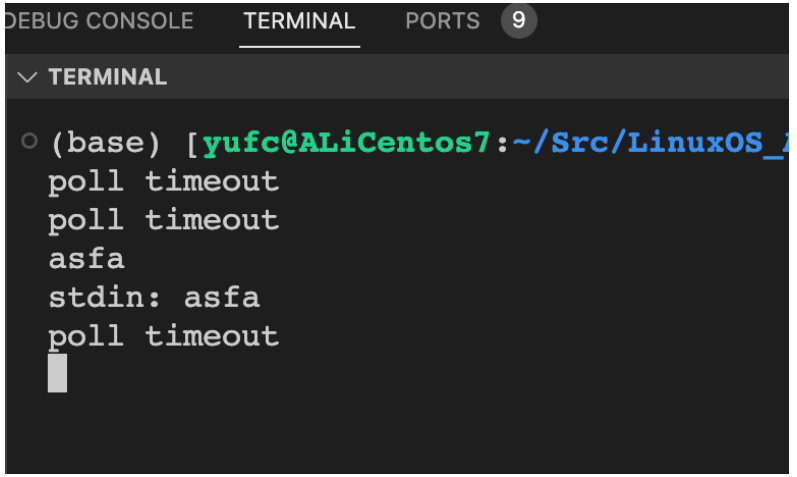
关于poll最大的大小

由程序员决定！

```c
int main()
{
    struct pollfd poll_fd;
    poll_fd.fd = 0;
    poll_fd.events = POLLIN;

    for (;;)
    {
        int ret = poll(&poll_fd, 1, 1000);
        if (ret < 0)
        {
            perror("poll");
            continue;
        }
        if (ret == 0)
        {
            printf("poll timeout\n");
            continue;
        }
        if (poll_fd.revents == POLLIN)
        {
            char buf[1024] = {0};
            read(0, buf, sizeof(buf) - 1);
            printf("stdin: %s", buf);
        }
    }
    return 0;
}
```

等待1000ms，如果1000ms之后，数组里面没有任何一个字段的revents是非0，即
没有任何一个fd就绪，就返回0，然后打印“poll timeout”
如果有一个就绪了，就打印字符串，
如果我们从键盘输入，0fd就会就绪，就会把键盘输入的东西打印出来



poll的优点：

1. 效率高（和多进程多线程比）
2. 有大量的链接，只有少量的是活跃的，比较合适
3. 输入和输出是分离的，不需要进行大量的重置
4. 参数级别，没有可以管理fd上限

poll的缺点：

1. poll还是要遍历，在用户层检测就绪，和与内核检测fd就绪，都是一样的
2. 需要内核到用户进行拷贝 --- 这个也很难避免
3. poll的代码也比较复杂，但是比select容易