

select、poll，epoll 笔记

sharwen

目录

1	select	2
1.1	select 接口	2

1 select

select 是 Linux 内核自身支持的一个网络编程，支持同时多 fd 监听的网络编程接口。

1.1 select 接口

```

1  int select(int nfd, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout
2  );
3  void FD_CLR(int fd, fd_set *set);
4  int  FD_ISSET(int fd, fd_set *set);
5  void FD_SET(int fd, fd_set *set);
6  void FD_ZERO(fd_set *set);
7
8  /* fd_set for select and pselect.  */
9  #define __FD_SETSIZE 1024
10
11 /* The fd_set member is required to be an array of longs.  */
12 typedef long int __fd_mask;
13
14 /* It's easier to assume 8-bit bytes than to get CHAR_BIT.  */
15 #define __NFDBITS (8 * (int) sizeof(__fd_mask)) // 每 long int 的位数 32
16 #define __FD_ELT(d) ((d) / __NFDBITS)
17 #define __FD_MASK(d) ((__fd_mask)1 << ((d) % __NFDBITS)) // 取模进行左移
18
19 typedef struct {
20     __fd_mask __fds_bits[__FD_SETSIZE / __NFDBITS];
21 } fd_set;
22
23 #define __FD_SET(d, set) ((void)(__FDS_BITS(set)[__FD_ELT(d)] |= __FD_MASK(d)))
24 #define __FD_CLR(d, set) ((void)(__FDS_BITS(set)[__FD_ELT(d)] &= ~__FD_MASK(d)))
25 #define __FD_ISSET(d, set) ((__FDS_BITS(set)[__FD_ELT(d)] & __FD_MASK(d)) != 0)
26

```

从代码易知道，select 支持的最大 fd 数量为1024个，fd_set 使用位运算来标识 $0 \leq fd < 1024$ ，如果超过这个范围，则会引发未知错误如程序崩溃等。这个 __FD_SETSIZE 也可以自己手动修改，以便支持更多的 fd 数量。虽然一般单个程序 fd 的标号不会超过 1024，但也存在一个程序处理超过这么多的 fd，因此这个隐患还是很明显的。

1.2 为什么 select 慢

在第一次所有监听都没有事件时，调用 select 都需要把进程挂到所有监听的文件描述符一次，并切调用 select 会将所有的 fd 从用户空间拷贝到内核空间。

有事件到来时，不知道是哪些文件描述符有数据可以读写，需要把所有的文件描述符都轮询一遍才能知道。

通知事件到来给用户进程，需要把整个 bitmap 拷到用户空间，让用户空间去查询。

参考文献

- [1] STL 源码剖析，侯捷等