

# Select模型

## select函数：

作用：监视socket集合，如果某个socket发生响应（链接或者收发数据），通过返回值以及参数告诉我们哪个socket有响应

```
1  int select(int nfd, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);
2  参数maxfd是需要监视的最大的文件描述符值+1
3  rdsetfd, writefds, exceptfds分别对应于需要检测的可读文件描述符的集合，可写文件描述符的集合及异常文件描述符的集合
4  struct timeval结构用于描述一段时间长度，如果在这个时间内，需要监视的描述符没有事件发生则函数返回-1
```

## 操作fd\_set的宏

```
1  FD_ZERO(fd_set *fdset);将指定的文件描述符集清空，在对文件描述符集合进行设置前，必须对其进行初始化
2  FD_SET(fd_set *fdset);用于在文件描述符集合中增加一个新的文件描述符。
3  FD_CLR(fd_set *fdset);用于在文件描述符集合中删除一个文件描述符。
4  FD_ISSET(int fd, fd_set *fdset);用于测试指定的文件描述符是否在该集合中。
```

## select模型：

将一组socket数组投递给系统，然后在系统里去查询socket是否有信号，过程都是在select函数里面去进行的，再到返回有操作的socket集合

select()函数主要解决的是**accept()函数阻塞问题**，而没有解决recv()和send()函数阻塞问题