# IO函数

## I/O读写函数

### recv

使用recv函数接收数据：

#include <sys/socket.h>

ssize\_t recv(int fd,void\* buf,size\_t len,int flags);

其中，buf是用户接收缓冲区的地址，len表示接收数据长度，recv函数从内核的接收缓冲区中复制数据到用户指定的缓冲区中：当内核中的数据比指定的缓冲区小时，一般会复制内核缓冲区的所有数据到用户缓冲区，并返回数据的长度；当内核接收缓冲区的数据比用户指定的长度多时，会将用户指定len长度的数据复制到用户缓冲区，其余数据等下次调用的时候复制。内核在复制完内核缓冲区的数据后，会销毁该数据。

其中flag的取值：

 recv函数返回错误时，错误码的含义：

recv函数的特点：

1. recv()常用于TCP连接，recvfrom()用于UDP连接（在数据报套接字DUP绑定了地址和端口后，也可以使用recv接收数据）；
2. 可以使用flag参数使其适用于多种场景；
3. 需要根据返回值来判断数据是否接收完毕。

### send

使用send函数发送数据：

#include <sys/socket.h>

ssize\_t send(int fd,const void\* buf,size\_t len,int flags);

send()函数的特点：

需要根据返回值来判断数据是否发送完毕

send()用于处理连接状态的描述符

flag标志位使得其行为多样

send函数errno含义：



### readv

#include <sys/uio.h>

ssize\_t readv(int fd,struct iovec\* vector,int count);

ssize\_t writev(int fd,const struct iovec\* vector,int count);

### writev

### recvmsg

### sendmsg

### I/O函数对比



## I/O操作函数

### pipe

### socketpair

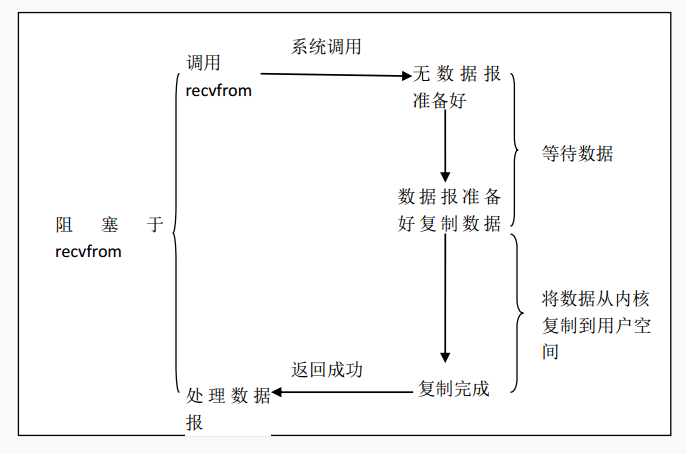
### mmap

### fcntl

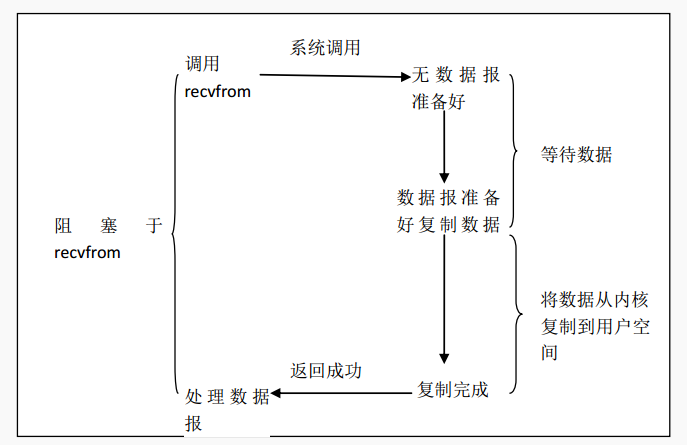
### splice

# IO模型

## 阻塞式I/O



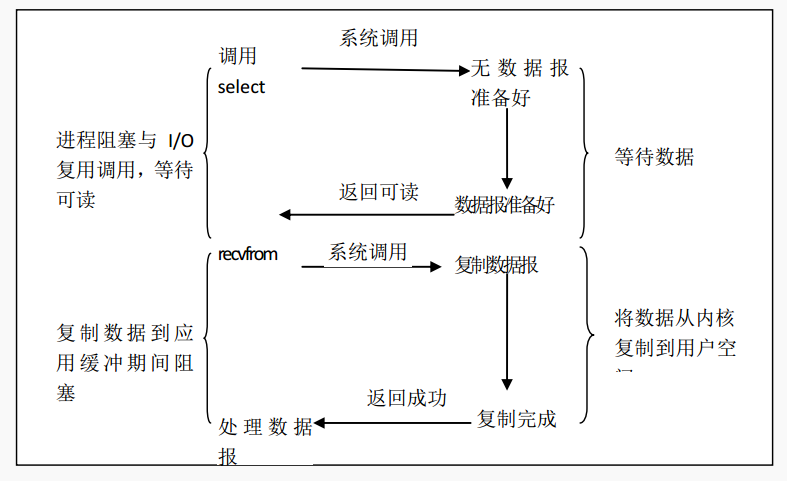
## 非阻塞式I/O



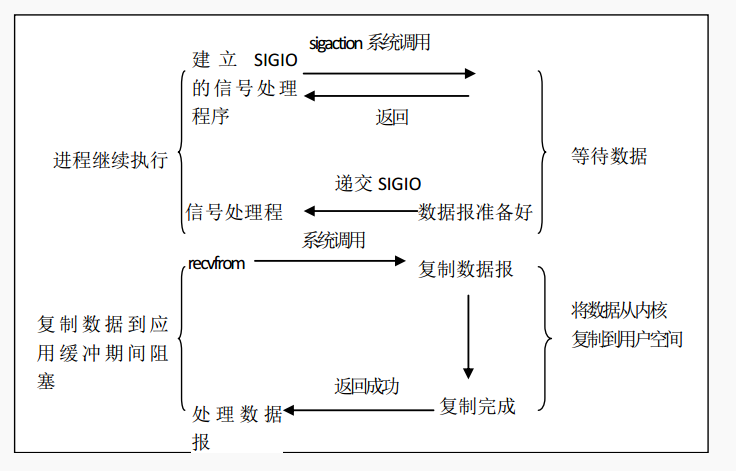
### 非阻塞accept

### 非阻塞connect

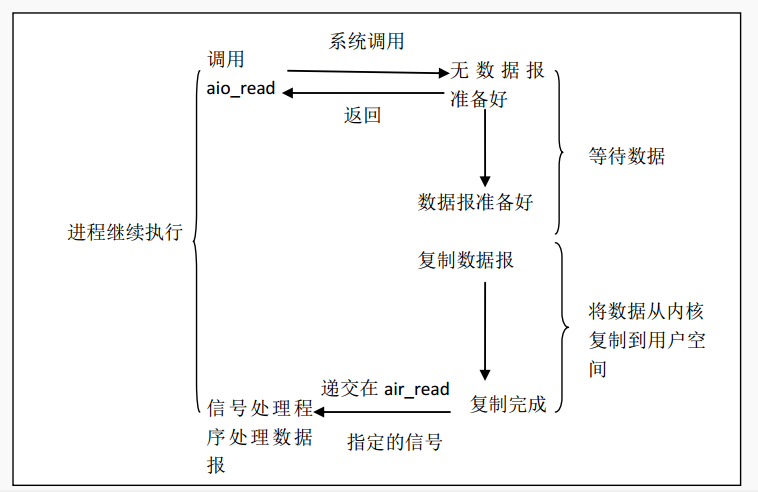
## I/O复用



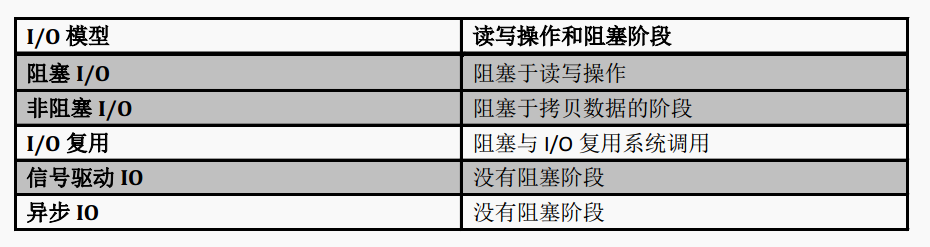
## 信号驱动式I/O



## 异步I/O



## I/O模型比较



### 阻塞/非阻塞

### 同步/异步