头文件：

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <strings.h>
#include <errno.h>
```

知识点1【错误退出perr_exit】（了解）

```
void perr_exit(const char *s)
{
 perror(s);
 exit(-1);
}
```

知识点2【对accept的包裹】（重要）

```
int Accept(int fd, struct sockaddr *sa, socklen_t *salenptr)
{
 int n;

again:
 if ((n = accept(fd, sa, salenptr)) < 0) {
 if ((errno == ECONNABORTED) || (errno == EINTR))
 goto again;
 else
  perr_exit("accept error");
  }
  return n;
 }
```

通过手册查看：man 2 accept

accept链接失败，返回-1，同时将错误值 设置到全局变量errno中

```
ECONNABORTED
       A connection has been aborted.

EFAULT The  addr argument is not in a writable part of the user address
       space.

EINTR  The system call was interrupted by  a  signal  that  was  caught
       before a valid connection arrived; see signal(7).
```

ECONNBABORTED:链接终止

原因：建立三次握手后 客户端给服务器发送的RST（复位）导致的异常

EINTR:链接被中断

原因：accept是慢系统调用，如果进程在一个慢系统调用(slow system call)中阻塞时，当捕获到某个信号且相应信号处理函数返回时，这个系统调用被中断，调用返回错误，设置errno为EINTR

知识点3【对bind的包裹】（了解）

```
int Bind(int fd, const struct sockaddr *sa, socklen_t salen)
{
 int n;

 if ((n = bind(fd, sa, salen)) < 0)
 perr_exit("bind error");

 return n;
}
```

知识点4【对connect的包裹】（了解）

```
int Connect(int fd, const struct sockaddr *sa, socklen_t salen)
{
 int n;

 if ((n = connect(fd, sa, salen)) < 0)
 perr_exit("connect error");

 return n;
}
```

知识点5【对listen的包裹】（了解）

```
int Listen(int fd, int backlog)
{
 int n;

 if ((n = listen(fd, backlog)) < 0)
 perr_exit("listen error");

 return n;
}
```

知识点6【对socket的包裹】（了解）

```
int Socket(int family, int type, int protocol)
```

```
{
 int n;

 if ((n = socket(family, type, protocol)) < 0)
 perr_exit("socket error");

 return n;
}
```

知识点7【对read的包裹】（了解）

```
ssize_t Read(int fd, void *ptr, size_t nbytes)
{
 ssize_t n;

again:
 if ( (n = read(fd, ptr, nbytes)) == -1) {
 if (errno == EINTR)//如果是被信号中断,不应该退出
 goto again;
 else
  return -1;
  }
  return n;
 }
```

知识点8【对write的包裹】（了解）

```
ssize_t Write(int fd, const void *ptr, size_t nbytes)
{
 ssize_t n;

again:
 if ( (n = write(fd, ptr, nbytes)) == -1) {
 if (errno == EINTR)
 goto again;
 else
  return -1;
  }
  return n;
 }
```

知识点9【对close的包裹】（了解）

```c
int Close(int fd)
{
 int n;
 if ((n = close(fd)) == -1)
 perr_exit("close error");

 return n;
}
```

知识点10【对bind的包裹】（了解）

```c
int tcp4bind(short port,const char *IP)
{
 struct sockaddr_in serv_addr;
 int lfd = Socket(AF_INET,SOCK_STREAM,0);
 bzero(&serv_addr,sizeof(serv_addr));
 if(IP == NULL){
 //如果这样使用 0.0.0.0,任意ip将可以连接
 serv_addr.sin_addr.s_addr = INADDR_ANY;
 }else{
  if(inet_pton(AF_INET,IP,&serv_addr.sin_addr.s_addr) <= 0){
  perror(IP);//转换失败
  exit(1);
  }
  }
  serv_addr.sin_family = AF_INET;
  serv_addr.sin_port = htons(port);
  int opt = 1;
  setsockopt(lfd, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt));

  Bind(lfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr));
  return lfd;
}
```

附件

wrap.c
3.81KB

wrap.h
860B